Agent-Based Structural Condition Monitoring for Nuclear Reactor Cores

Gordon James Jahn

A thesis submitted for the degree of Doctor of Philosophy to

Institute for Energy and Environment
University of Strathclyde

April 2011

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: ..... Date: .....

Abstract

A significant proportion of the UK energy needs are currently serviced by a fleet of ageing nuclear reactors. Ensuring that these reactors are operated safely is the highest priority and the structural health of their cores, that provide channels for control rods and coolant gas, is a key aspect.

This thesis focuses on the application of structural health monitoring to the graphite reactor cores used in the UK and presents a specification for the use of structural health monitoring (SHM) techniques already established in bridge and aircraft monitoring, with data obtained through existing reactor monitoring processes. This approach utilises statistical and clustering techniques on monitoring data that can be acquired during online operation of the plant. The use of existing monitoring processes to complement the established inspection regime for nuclear reactors is a novel contribution from this work.

As part of proving the SHM approach, this thesis reports on work undertaken to identify suitable data and numerical limits for the cluster analysis. This analysis considers the data with respect to the stated aim of detecting core distortion and demonstrates that the chosen data and values are acceptable and conservative in the context of reactor condition monitoring.

An assessment of the SHM solution is presented describing the implementation of the SHM approach using a multi-agent system (MAS), IMAPS. This implementation required consideration of using MAS technology for condition monitoring, and the novel contribution of a technique for storing and retrieving historical data in a manner concomitant with both MAS and relational database theory is presented.

The thesis concludes that condition monitoring is feasible on the graphite cores, and that multi-variate analysis through SHM implemented within a MAS offers a storage and analysis platform that can both handle the data volumes and accommodate further extensions as required.

Acknowledgements

I would like to express my gratitude to Professors Jim McDonald and Stephen McArthur at the University of Strathclyde for the opportunity to work on this project.

I would like to thank all of my colleagues in the Advanced Electrical Systems Research Group for creating a fun and inspiring environment within which to undertake this research. Particular thanks are extended to Dr. Victoria Catterson for access to her work on transformer monitoring which allowed the case study covering this area to be prepared.

Special thanks are extended to Jacqueline Jahn and Emma Stewart for reading the numerous drafts.

Finally, I must sincerely thank EDF Energy and everyone in the Graphite Core Project Team who financially supported this work and helped with domain knowledge to deliver the project.

This thesis represents the views of the author and not necessarily those of EDF Energy, EDF Group companies, or their employees.

Contents

List of Figures

List of Tables

List of Publications

A number of publications have resulted from the research undertaken in the completion of this work. This includes publications arising from the work on monitoring nuclear plant and related work on power transformers.

Publications Arising From The Research

The following publications are directly related to the work reported within this thesis:

C.J. Wallace, G. J. Jahn and S. D. J. McArthur. Multi-Agent System For Nuclear Condition Monitoring. In *Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Taipei, May 2011.

G. J. Jahn, S. D. J. McArthur and D. Towle. Data Storage and Analysis Techniques for Monitoring Advanced Gas-cooled Reactor Structural Integrity. In *Proceedings of the 7th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPIC&HMIT 2010)*, Las Vegas, November 2010.

G. J. Jahn and S. D. J. McArthur. IMAPS – A System For Managing Graphite Core Information. In *Securing the Safe Performance of Graphite Reactor Cores Conference Proceedings*, Nottingham, November 2008.

G. J. Jahn, S. D. J. McArthur and J. McDonald. Utilising Control and Instrumentation Data for Reactor Condition Monitoring. In *Institute of Nuclear Engineers Control and Instrumentation Conference Proceedings*, Manchester, 2007.

G. J. Jahn, S. D. J. McArthur, J. Reed and D. Towle. Staged Implementation of an Agent Based Advanced Gas-Cooled Reactor Condition Monitoring System . In *IEEE Power Engineering Society General Meeting, 2007*, Tampa, July 2007.

G. J. Jahn, S. D. J. McArthur and J. Reed. An Integrated Architecture for Graphite Core Data Analysis. In *Ageing Management of Graphite Reactor Cores Conference Proceedings*, Cardiff, November 2005.

G. M. West, G. J. Jahn, S. D. J. McArthur and J. Reed. Graphite Core Condition Monitoring Through Intelligent Analysis of Fuel Grab Load Trace Data. In *Ageing Management of Graphite Reactor Cores Conference Proceedings*, Cardiff, November 2005.

Publications Related To The Research

The following publications on equipment monitoring are related to the work reported within this thesis:

S. D. J. McArthur, S. M. Strachan and G. J. Jahn. The design of a multi-agent transformer condition monitoring system. In *IEEE Transactions on Power Systems*, Vol. 19, Number 4, Pages 1845-1852, 2004.

S. M. Strachan, G. J. Jahn and S. D. J. McArthur. Intelligent Diagnosis of Defects Responsible for Partial Discharge Activity Detected in Power Transformers. In *Intelligent System Applications in Power Systems Conference (ISAP2003)*, 2003.

Glossary of abbreviations

ACL	Agent Communication Language
AGR	Advanced Gas-cooled Reactor
AI	Artificial Intelligence
ALADDIN	Autonomous Learning Agents for Decentralised Data and Information Networks
ALARP	As Low As Reasonably Practicable
AMS	Agent Management System
AP	Agent Platform
APACS	Advanced Plant Analysis and Control System
API	Application Programming Interface
BDI	Belief, Desire and Intention
BWR	Boiling Water Reactor
CBMU	Channel Bore Monitoring Unit
CCGT	Combined Cycle Gas Turbine
CCTV	Closed Circuit Television
CIM	Common Information Model
CO	Carbon Monoxide
CO_2	Carbon Dioxide
COMMAS	COndition Monitoring Multi Agent System
CPCS	Core Protection Calculator System
DAI	Distributed Artificial Intelligence
DBA	Database Administrator
DBMS	Database Management System
DF	Directory Facilitator
FGLT	Fuel Grab Load Trace
FIPA	Foundation for Intelligent Physical Agents
GIS	Gas Insulated Substation

HP	High Pressure
HQL	Hibernate Query Language
HSE	Health and Safety Executive
HTTP	HyperText Transfer Protocol
IIOP	Internet Inter-Orb Protocol
IMAPS	Intelligent Monitoring Assessment Panel System
IP	Intermediate Pressure
JADE	Java Agent Development Environment
JPA	Java Persistence API
KIF	Knowledge Interchange Format
LCO	Limiting Conditions for Operation
LP	Low Pressure
MAP	Monitoring Assessment Panel
MAS	Multi-Agent System
MTS	Message Transport System
NII	Nuclear Installation Inspectorate
OECD	Organisation for Economic Co-operation and Development
OO	Object-oriented
ORDBMS	Object Relational Database Management System
OWL	Web Ontology Language
PEDA	Protection Engineering Diagnosis Agents
PWR	Pressurised Water Reactor
QoS	Quality of Service
RDBMS	Relation Database Management System
SHM	Structural Health Monitoring
SL	Semantic Language
SQL	Structured Query Language
T&D	Transmission and Distribution
TSSM	Technical Safety Support Manager
UK	United Kingdom

Chapter 1

Introduction

1.1 Introduction to the research

The electrical power industry has been undertaking increasing amounts of asset management over the last 20 years with companies striving to find the safest and most economic strategy for operating, inspecting, maintaining and monitoring their plant items. During this time, plant items such as transformers and circuit breakers have increasingly seen the use of diagnostic and monitoring techniques for lifetime management which aim to not only maximise the utility of the monitored device, but to also help identify the most appropriate time to consider replacement.

Monitored devices are typically high-value assets in terms of both equipment cost and contribution to safe and economic operation of the network, and this has evolved, for some operators, into a change in maintenance approach whereby fixed-term maintenance costs are reduced by implementing a condition-based maintenance strategy. More recently, the falling cost of data storage and communication has made it viable to equip individual plant items with online systems constantly surveying and logging information that can be used to infer the operational condition of the device.

Against this general backdrop of increasing monitoring in the power industry, the nuclear generation domain has been relatively stagnant. A conservative approach has been employed in this domain and efforts have been focused on improving predictive models and attempting to validate these with observations encountered during operation. Whilst improving the models through increased understanding aids lifetime management,

it is only over the past 5 years that the unique political, commercial and safety considerations are allowing the nuclear industry to move towards novel monitoring techniques and systems [JMR05].

This thesis presents the background to asset management within the wider power industry and the unique challenges associated with nuclear power generation before describing the current state-of-the-art and commercial systems available to undertake condition monitoring today. The challenges of analysing data gathered from various nuclear generation systems are described and a framework that is able to operate with the diversity encountered in such systems is outlined. Suitable analyses for the application are then presented along with the results of analysing data from an in-service reactor. The research area of multi-agent systems is described and an obstacle to their use in this application, reliable data storage, is detailed and a method for solving this problem is presented. Finally, a case study demonstrating the data analysis techniques and integration platform is presented in which the outcomes are assessed.

It is shown that the agent-based model is suitable for the storage and analysis of graphite core data and that this will allow for the integration of further data types and data sources in the future.

1.2 Contributions

This research reports on work undertaken in providing techniques for condition monitoring of graphite reactor cores through the application of techniques used within other key domains which involve large structures with life safety implications. In doing this, examples from bridge [OBM06], airframe [MKB+02] and nuclear containment [Bro07] monitoring are considered along with the approach being used in each case. Having demonstrated the applicability of these techniques, the available data are considered in this context for analysis.

The data available from the reactor monitoring processes is then considered for this analysis; a method of assessing the condition is the devised and demonstrated.

Having demonstrated that the analysis is possible, a deployment platform is chosen. Multi-agent systems have been used for several condition

monitoring applications and are chosen for this application. Drawbacks of the multi-agent approach for this application are discussed and a method of overcoming this by integrating agent-based systems with database systems is proposed and demonstrated.

The key novelty contributed in this work is summarised as:

- An approach for the use of monitoring data collected during routine nuclear plant operation to identify structural defects within the reactor,
- Specification of how to use established structural health monitoring techniques with the discrete data available from reactor analyses using a cluster analysis approach,
- Identification of appropriate limits for the identification of structural defects within graphite reactor cores,
- A proposal to implement the core monitoring system as a multi-agent system allowing for the later integration of further analysis techniques,
- An implementation of a flexible and generic archive agent that can store arbitrary data within a multi-agent system,
- A case study demonstrating the use of the flexible archive agent within a transformer monitoring system,
- A case study of the use of the flexible archive agent within a multi-agent system for the detection of structural defects,
- A case study of the clusters identified by the process to demonstrate the effectiveness of the approach to detecting defects,
- A series of recommendations for how the cluster analysis process itself might be automated to further filter the information presented to end-users.

In addition to the research novelty highlighted above, there were a number of useful industrial outcomes. The research has led to a production-grade system being implemented for the storage of the reactor data with

experimentation and analysis continuing to appraise the techniques for automating the data analysis. This system supports the monitoring process at four nuclear power stations within the UK and, as part of this, contributes to the safe operation of nuclear generation plant.

1.3 Thesis overview

This thesis begins, in Chapter 2, by explaining the process of electrical power generation within a conventional steam-powered plant and goes on to look at how nuclear reactor systems can be added in order to raise steam. Chapter 3 then describes condition monitoring and specifically the technique of Structural Health Monitoring (SHM). This technique has been applied to the lifetime management of many building, bridges and aircraft but has not been applied to the similar area of structural monitoring of graphite reactors.

Having established that SHM is a suitable analysis technique for monitoring nuclear reactor cores, Chapter 4 shows how the SHM techniques can be applied to the data available in the graphite monitoring application.

Next, Chapter 5 investigates multi-agent systems used in more traditional condition monitoring systems and their suitability for the non-maintainable application of the graphite cores. This chapter concludes that data storage presents a challenge to using agent-based systems and that this will require resolution.

Chapter 6 describes the current state of the art in terms of data storage within agent-based systems and explains why, within certain condition monitoring areas, a more conventional approach to data storage is desired and proceeds to suggest three different approaches to improved data storage for agent-based systems, all of which offer different advantages and disadvantages to both organisations and agent developers.

Following this, Chapter 7 describes the application of condition monitoring to civil nuclear power reactors utilising data which is already collected as a matter of course, and shows how the technologies researched and developed throughout Chapters 4 and 6 can be used in practice. Additional benefits which can be realised for the operator through the use of such a system are also highlighted at this point during an appraisal of the

system.

The general topics covered in the chapters throughout this thesis are shown in the following diagram, which identifies where the different concepts are brought together in order to build the whole monitoring system, IMAPS.

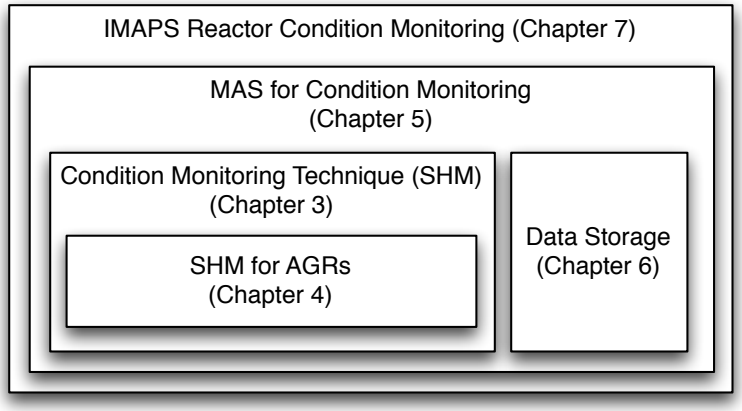

Figure 1.1: Thesis arrangement with respect to IMAPS

The thesis then closes with final conclusions in Chapter 8 which additionally highlights further research which could be pursued, following on from the work presented here.

Chapter 2

Nuclear power generation and safety processes

2.1 Introduction

Electrical power can be generated in many ways; renewable energy sources such as wind and solar play an important part in the overall generation mix with a government target of fifteen percent of UK electricity being generated from renewable sources by 2020 [Ren08]. Despite this, as of 2007, 78% of electricity was generated from fossil sources – coal, oil and gas – with nuclear making up 15% and other sources, including renewables, accounting for the remainder [Dep08].

Despite a declining share of the electrical generation mix, dropping from 18% in 1996 to 15% in 2007 [Dep08], nuclear power remains an important energy source contributing over 57 TWh of electrical energy to UK consumers with no carbon output during electricity production. This compares with values for coal-fired stations of 229 kg C/MWh and gas-fired combined cycle stations of 103-133 kg C/MWh.

Carbon output from electricity generation has become an issue as world governments attempt to meet climate change obligations and the low carbon properties of nuclear generation has made the prospect of a new generation of nuclear plants in the UK increasingly likely. Long lead times on building and commissioning new nuclear stations [Tra06], of between 5 and 11 years after all regulatory approval has been obtained, requires that the existing fleet be able to continue safely generating power until the

replacement units are available. If they cannot continue to generate then it is very likely that further fossil plants will be built to ensure that there is no electricity shortfall.

Fossil fuel and nuclear generating stations both operate on a steam-cycle where the fuel is used to heat water to pressurised steam, which is then used to turn a shaft and thus generate electricity. Most power stations use a multiple-stage steam process with a high-pressure turbine, and further intermediate and lower pressure turbines intended to maximise the amount of energy extracted from the steam; the number of processes vary depending upon the available steam conditions.

In most cases, all turbines for a particular steam source are connected to one shaft which is attached to an electrical generator. The steam causes the shaft to rotate which moves the rotor within the generator. This produces electricity which is then transmitted to consumers via a power grid.

A diagram showing typical feedwater and steam paths in a steam-cycle thermal generation plant is shown in Figure 2.1. This diagram shows the typical three-stage generator with energy in the form of heat being applied to a boiler to convert feedwater to steam, steam passing initially through a high pressure (HP) turbine before entering a re-heat stage and passing through intermediate and low pressure (IP and LP) stages before passing through a condenser for recirculation. Cooling for the condenser is not shown, but will typically be another water-based cooling system utilising either cooling towers or once-through cooling from a nearby river or sea.

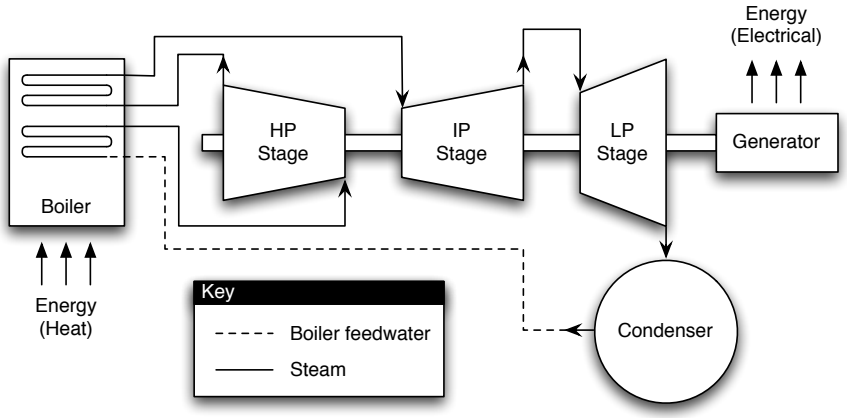

Figure 2.1: General Steam-Cycle Power Station Layout

The layout varies between designs and these variations can be for diverse reasons. For example, fewer turbines being used due to lower steam temperature. Variations such as these change the overall thermal efficiency of the plant which is normally expressed as a percentage of thermal power used to heat the steam which is converted to electrical energy. The European average thermal efficiency for coal-fired plants is about 35% with new designs reporting efficiencies around 50% [BKB06]. Combined cycle gas turbine (CCGT) plants which operate on natural gas can have a thermal efficiency of up to 60% [SRG03].

The fundamental difference between nuclear and fossil-fuel stations is how that fuel is used to raise steam. Fossil stations burn coal, oil or gas to create heat whilst the nuclear power station utilises exothermic nuclear fission to achieve the same goal.

2.2 Nuclear power generation

The civil nuclear plants in commercial use today all utilise some form of controlled nuclear fission reaction to generate heat. The process of fission requires that a neutron collides and be captured by the nucleus of a fissile atom which then splits into two or more smaller nucleii. The likelihood of a successful neutron capture and fission is increased when neutrons are slower-moving "thermal" neutrons [Ben71]. Typical fast neutron speeds are around 20,000 km/s whilst moderated neutrons are moving at around 2.7 km/s. During a successful fission, further fast-moving neutrons are released along with the energy which is subsequently used to raise steam. In order to support the sustained nuclear reaction required for a power-generation plant these fast neutrons must be slowed by a moderator material which provides more slow neutrons to sustain the reaction; the most common moderators in commercial reactors are water and graphite.

Through moderation and further fission, a single neutron can result in several further free neutrons within the reactor very quickly. This process can therefore result in explosive, and hence dangerous, release of energy. As such, the number of neutrons within the reactor must be controlled by placing a material within the reactor that can absorb neutrons. The overall fission reaction is controlled by altering the amount of neutron absorption

to keep the process running at the required rate.

The nuclear fission process continues until either all fissile material, the reactor's fuel, has been used up or until there are no neutrons to propagate the reaction. Natural radioactive decay can result in a small number of neutrons being spontaneously ejected from some nucleii and statistically some of these may eject at a rate suitable to restart the fission reaction. As such, wherever there is a suitable density of fissile material, or critical mass, neutronic control must be exercised.

All nuclear reactors have some form of primary coolant which allows the heat generated as part of the fission reaction to be carried away from the core. The coolant is generally used to heat water in a secondary coolant circuit to steam and thus turn the turbines in the same way as a fossil-fuelled station. For some designs, there is only a single cooling circuit with the same coolant that passes through the reactor being used to drive the turbines.

There are a number of different nuclear reactor designs which use various fuels, fuelling mechanisms, moderators and control technologies. Before considering common reactor designs, it should be noted that the maximum operating lifetime of the station is usually limited by the maximum usable life of the reactor itself; generally all other parts on the station can be replaced or repaired, if at high cost, but the reactor itself cannot [NJT87]. As such, this is a high-value asset upon which the entire revenue earning capability of a station can rest.

At the current time, most operating reactors are generation II reactors. A small number of generation III reactors offering longer lifetimes, increased safety and improved economics of operation have been commissioned with others under construction. Generation IV designs are still under development. Due to the small population of reactors of newer designs, three main reactor designs, all generation II, are detailed. These designs are listed below and cover over three-quarters of all operational reactors [Int06].

- The Advanced Gas-Cooled Reactor, the principal reactor design in use within the UK, but used only in the UK,
- The Pressurised Water Reactor, a design used for around half of all power reactors and of which there is one of in the UK, and

- The Boiling Water Reactor, a design which is used for around one quarter of all reactors worldwide.

2.2.1 Advanced gas-cooled reactors

The Advanced Gas-cooled Reactor (AGR) design dates from the 1960s when a new reactor design was developed based on the experience of the Magnox stations of the 1950s. The design therefore uses gas-cooled reactors with a graphite moderator.

AGR graphite is manufactured in cylindrical bricks which are then drilled giving spaces to hold fuel, to give control rods access to the core and to allow coolant to flow around the reactor. The bricks are arranged in a lattice and, by aligning each layer of bricks with the one below, they form a number of vertical channels.

The AGR fuel stringers that occupy the large fuel channels each comprise eight elements. Each fuel element contains thirty six fuel pins arranged in such a way as to give space for coolant flow and sufficient surface area for good heat transfer. There are articulations between each fuel element making them tolerant of any distortion in the shape of the channel. There are around 300 fuel channels in an AGR reactor [Bow82].

Between the large fuel channels there are a number of smaller interstitial channels. Around 80 of these are used for control rods and the remainder are either left empty or have alternative uses, such as the installation of neutron sources or flux scanning equipment. The bricks which form the channels each have keyways on the outer edge and graphite keys are loosely placed into these gaps to limit the movement between adjacent bricks, making the core a single interlocking structure. The keyways are slightly oversized to account for changes in graphite dimensions throughout the operating life of the station [Dav96]. A diagram of these bricks and the keying system is shown in Figure 2.2.

A picture taken during the construction of an AGR reactor shows how these bricks are arranged and is shown in Figure 2.3 [WH72]. The larger fuel bricks can be seen with smaller interstitial bricks between them.

The general layout of the AGR station is shown in Figure 2.4. The AGR design uses two coolant loops; the primary coolant loop uses carbon dioxide (CO_2) which enters the core at the bottom of the reactor and passes

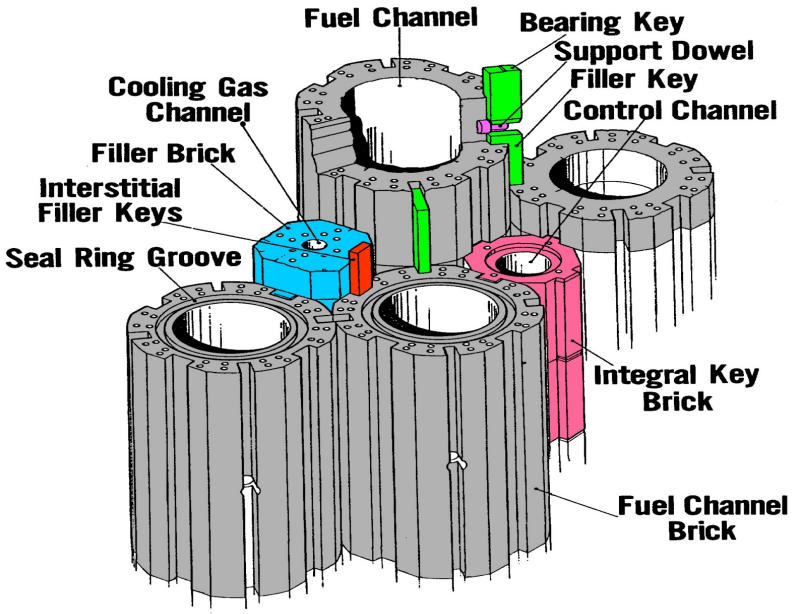

Figure 2.2: AGR graphite brick arrangement [Ste05]

vertically through the core before exiting and being used by the boilers for steam production [How06b]. The second loop contains water in liquid and steam forms; boilers create high pressure steam to drive the turbines and the condenser, cooled by seawater, turns the steam back to water to be reused.

The AGR was designed to have comparable steam conditions leaving the boilers as with conventional coal-fired plants. This was intended to allow standard parts to be used within the steam cycle of AGR-based nuclear power stations. It also means that there are three turbine stages with a reheater between the high pressure and intermediate pressure turbines. This results in thermal efficiency for these plants of 42% [How06b], which exceeds many coal-fired plants.

2.2.1.1 AGR power control

AGR power output is managed using control rods. The rods are driven and held by a control rod actuator above the core which ensures that, in the event of a power failure, the rod will "fail-safe" and fall into the core, arresting the fission reaction.

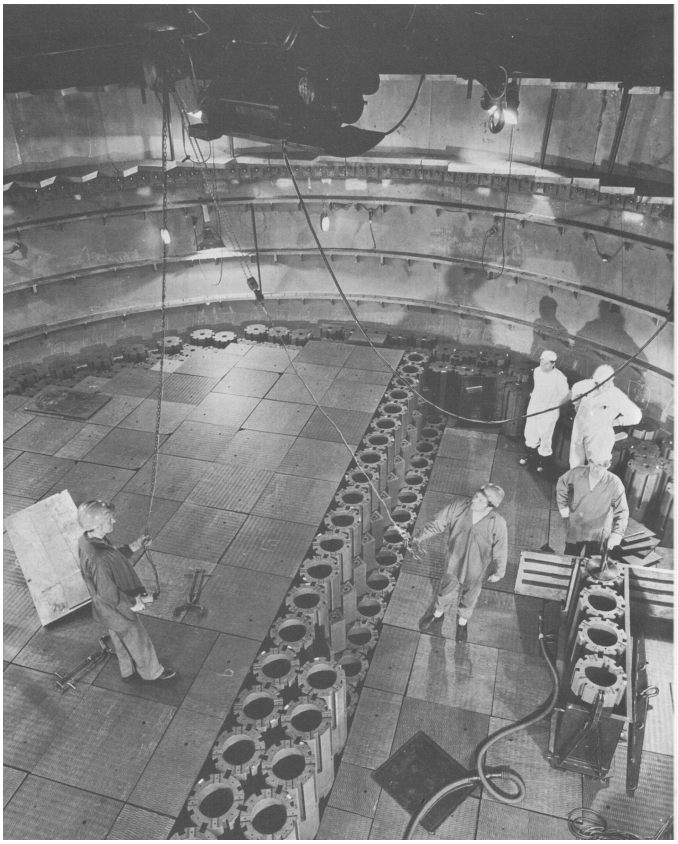

Figure 2.3: Graphite moderator under construction [WH72]

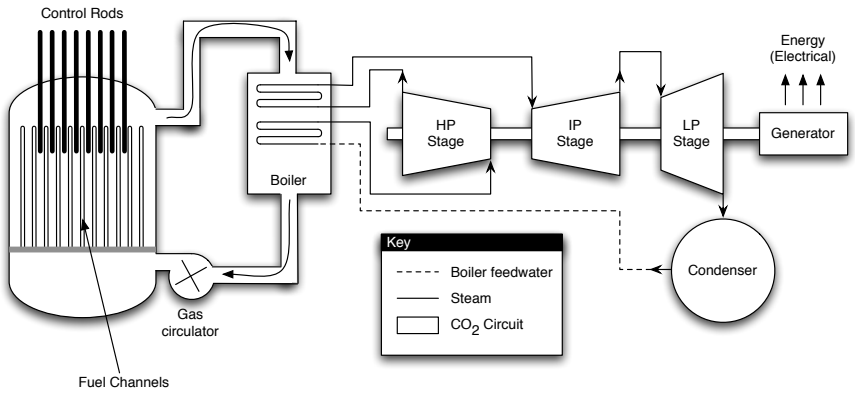

Figure 2.4: Advanced Gas-Cooled Reactor Power Station Layout

There are two types of control rod in AGRs; the terminology for these varies, but in general there are "bulk" rods which are used for shutting down the reactor and stopping all nuclear fission and "regulating" rods which are adjusted during the day-to-day running of the reactor. The bulk rods are made of steel with a boronated insert whilst steel-only rods are used for the regulating rods. The different materials allow for different levels of power control [How06b].

2.2.2 Pressurised water reactor

Whilst the UK was progressing with the AGR programme, the Pressurised Water Reactor (PWR) design was heavily adopted elsewhere, particularly in the United States and France. At the time that the UK was building its first PWR – Sizewell B in 1990 – 65% of all nuclear reactors worldwide were of this design [Bin90]. Sizewell B station opened in 1995 and remains the only PWR in the UK.

The overview of the steam cycle used within PWR systems is shown in Figure 2.5 [How06a]. It can be seen immediately that this design does not have the intermediate pressure turbine that would normally be present on conventional plant and which is seen in the AGR design. This has been removed due to the lower steam temperatures in the PWR design and results in lower thermal efficiency for this type of plant, typically around 32% [YXQ01][Bri06].

The PWR design uses water as both coolant and moderator, relying on the water being kept at high pressures in order that it remain in a liquid state at operating temperatures. Different techniques, when compared with the AGR, can be used in order to regulate the power output of a PWR reactor system. Since the insertion of control rods results in less fuel-use in the part of the reactor where control rods are positioned, an alternative of introducing boric acid into the coolant water can be used. This allows regular fuel burn-up across the reactor but the water then requires treatment to remove the boron [Wor09].

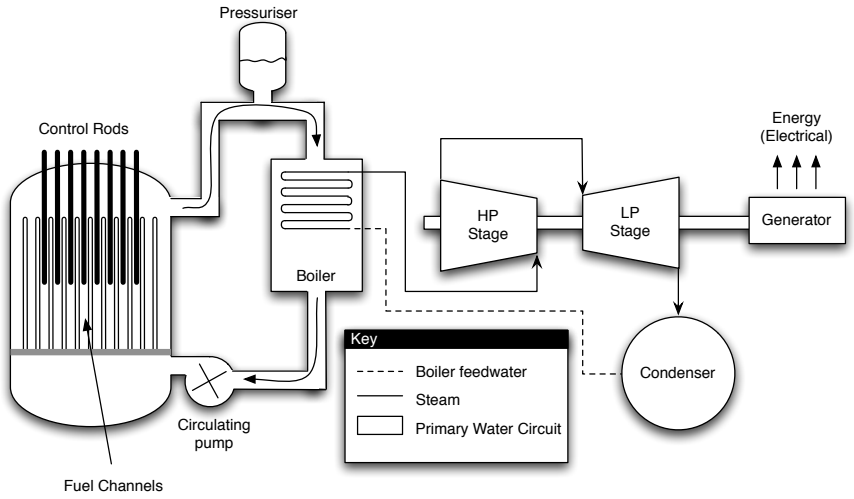

Figure 2.5: Pressurised Water Reactor Power Station Layout

2.2.3 Boiling water reactors

The Boiling Water Reactor (BWR) design, like the PWR design, uses demineralised water within the primary coolant circuit. The BWR vapourises water directly in the reactor; it performs both the role of the energy source and the boiler seen in both the AGR and PWR designs. The BWR design does not require a separate steam-raising unit. The general layout of the BWR is shown in Figure 2.6.

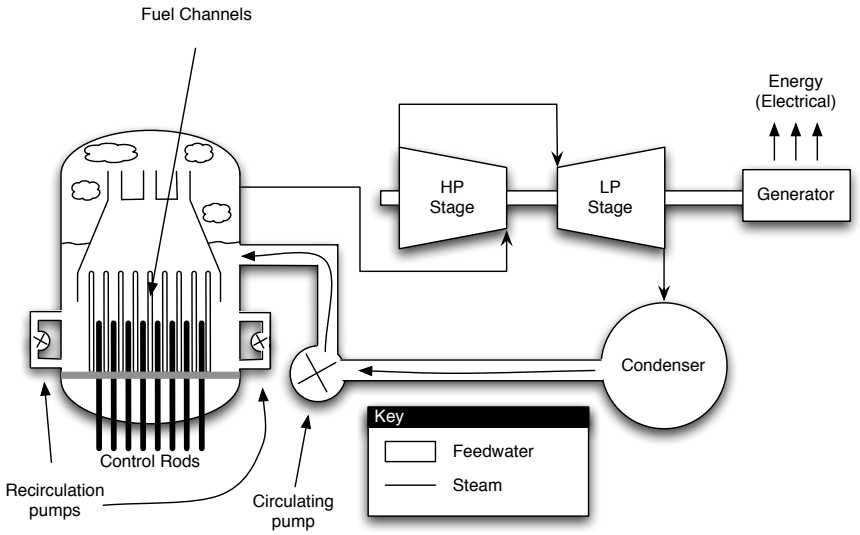

Figure 2.6: Boiling Water Reactor Power Station Layout

In this reactor design, the control rods are mainly used for turning the reactor on or off; usually they are not used to manage the power output. This is possible due to the process of the water boiling in the reactor; as the water boils and creates steam bubbles, or voids, within the reactor, the volume of liquid moderator in the core falls as it is displaced by gaseous steam. With less liquid water moderator to provide thermal neutrons, the reactivity falls and so too does power output. The BWR reactor design therefore features recirculating pumps (seen towards the bottom of the reactor in Figure 2.6). By increasing the recirculation, the voids can be expelled quicker resulting in increasing moderation and therefore power output [Ped98].

The BWR is the second-most prevalent design in use and has a similar thermal efficiency – 32% – to the PWR design [Bri06].

2.3 Nuclear systems operation

Within the UK, sites on which nuclear systems operate must be licensed by the Health and Safety Executive (HSE). Under their remit, the HSE takes on the role of regulator and is obligated to ensure that all nuclear operations carried out by licensees are done so in a safe and responsible manner. In order to meet their remit, the HSE requires that licensees provide safety cases [Hea07] for the systems and procedures that they use during the operation of their facility. The formal definition of this document is provided below.

> "A safety case is the totality of documented information and arguments developed by the licensee which substantiates the safety of the facility, activity, operation or modification in question. It provides a written demonstration that relevant standards have been met and that risks have been reduced to a level that is as low as reasonably practicable (ALARP). The safety case is not a one-off series of documents prepared to obtain a nuclear site licence but an holistic, living framework which underpins all safety- related decisions made by the licensee. The safety case must be updated regularly and the implications of proposed facility and other safety-related changes need to be

> examined against it and, when necessary, additional demonstrations of safety provided. Accordingly, the requirements to produce and maintain safety cases are embodied in the conditions attached to all nuclear site licences."

From this, it is clear that safety cases are important; all operations within a nuclear power station, including equipment and processes, are affected and this naturally extends to the nuclear reactors where it must be demonstrated that these are safe to operate. The most important factor for operational reactor cores is that they are able to be safely shut down, preferably using the primary shutdown mechanism. In AGR cores, the primary shutdown mechanism comprises the control rods. Significant distortion of the core may, however, lead to the channels formed by the graphite bricks being misaligned with the control rod mechanisms and thereby preventing them from entering the core and shutting it down.

As part of proving reactor safety, reactors have historically been subject to statutory outages at regular intervals. One AGR station in Scotland, Hunterston B, originally had statutory outages every two years [Yeo80], but after justifying a change in the interval these later became every three years [Bro07]. During these outages, maintenance across the station is carried out and inspections on the graphite moderator take place. These inspections cover material properties of the graphite and individual channel surveys and aim to demonstrate that no unsafe distortion has occurred. Originally these inspections involved lowering a television camera through the channel and simply viewing the core, but the inspection process later evolved with the addition of the Channel Bore Monitoring Unit (CBMU) to accurately measure many parameters associated with the reactor. A diagram showing this unit, which is lowered into the core, is shown in Figure 2.7 [CJRW96]. This device is capable of measuring channel diameter, ovality and tilt and gives very accurate measurements of the channel condition [CBR07]. Further developments of this device have also been developed to reduce the time taken to perform inspections.

Having performed the inspections, the station operators use the information in association with experimentation, testing and modelling of a range of factors to assess the condition against predictions of how the cores will degrade. The periodic inspection should always ensure that the core

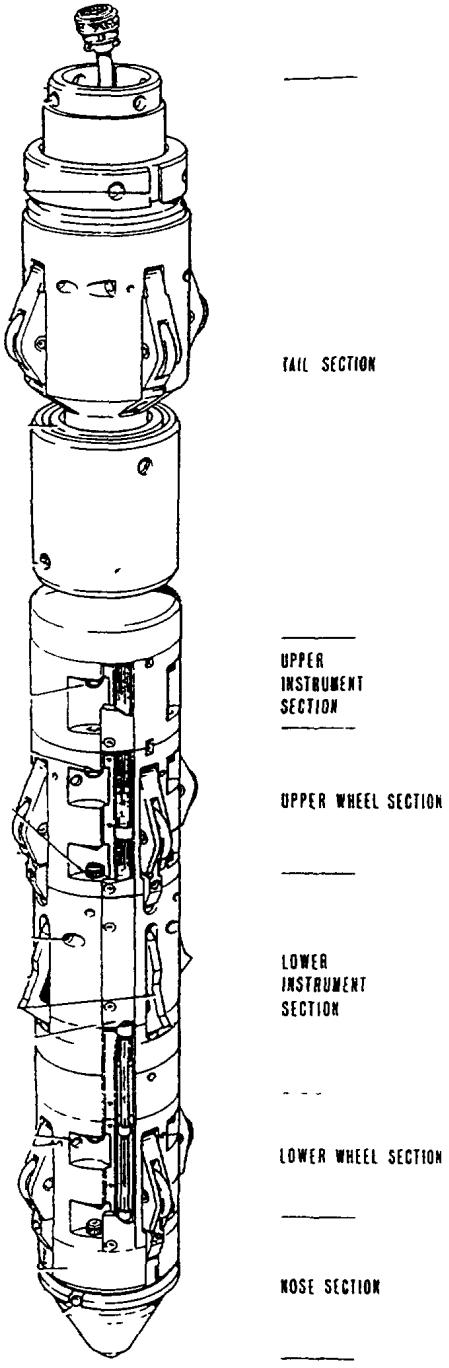

Figure 2.7: AGR Channel Bore Measuring Unit [CJRW96]

is ageing in line with the predictions gleaned from the extensive modelling of how the core should age when subjected to neutron dose.

In addition to the periodic inspections, there are ongoing data analysis processes. As an example, two separate modelling analyses use both the expected power output from a fuel channel, as calculated by the neutronic properties of the fuel, and the thermal power calculated from the gas temperatures measured using the thermocouples embedded within the fuel assembly. These can be compared to ensure that power distribution and coolant across the core is correct. There are additional analyses undertaken on control rods and, more recently, fuel grab load traces.

The regulatory oversight that is required for the approval of safety cases is the primary means of ensuring that nuclear operations are safe. Similar bodies exist across the world, such as the Nuclear Regulatory Commission in the US, and the worldwide International Atomic Energy Authority provides for checks and balances to help ensure that all nuclear power production is safe.

2.4 Sensors for nuclear systems

Whilst there are regular analyses undertaken on AGR data, there are no sensors, such as strain gauges, within the reactors to provide structural information. Electronic sensors fail quickly within the radioactive environment and optical techniques using doped fibres, whilst promising, do not yet provide a proven measurement platform [FBB+02].

The harsh environment within all reactors means that measurements are usually limited to temperature, pressure and flux, and these are the available measurements for AGR cores. Due to the radioactivity of the cores and the constraints on access that this imposes, retrofitting new sensors, even if they were developed, is unlikely to be feasible due to the risks involved.

This means that at the present time the only reliable measurements available are those already retrieved and analysed. These are now examined in more detail.

2.5 Monitoring reactor core data

Given the difficulty in providing additional sensors to detect the effects of core distortion, the AGR operators have opted to try to make better use of data that is already collected during normal operations.

Since around 2005, the operators have been considering the results of several previously separate data analysis methods together, with a view to increasing their knowledge of the state of the core. This has involved collating and assessing the results of thermal-neutron power comparisons, control rod performance statistics and Fuel Grab Load Trace (FGLT) analyses as well as any other appropriate indicators. The formal process for undertaking this analysis is the Monitoring Assessment Panel.

2.5.1 Monitoring Assessment Panels

As the AGRs have aged, the operators have instituted a process known as Monitoring Assessment Panels (MAPs) to help meet the safety case requirement for monitoring [JMR05]. MAPs aim to holistically assesses the data obtained from the operation of the nuclear reactors. As discussed in the previous section, it is not readily possible to install additional sensing equipment within an operational core. The MAP process therefore brings together the data gathered via a number of routes and identifies whether there are multiple indications of core degradation and deformation around the same time or location, which could be indicative of core distortion.

MAPs comprise a group of representatives from across a station who meet regularly to review observations reported from various analytic and monitoring processes. This information is then correlated to assess whether there is any evidence of core distortion and thereby any likelihood of there being difficulty with fuel or control rod movements which would impact upon safety.

MAP meetings consider a number of class one parameters – information from these areas is likely to be a good indicator of core distortion. These parameters are:

- *Thermal-Neutron Channel Power Ratios*: comparisons between theoretical input power from the nuclear process and the output power

measured by thermocouples. Distortion of the core can be detected by the differences between measured and calculated values.

- *Fuel Grab Load Trace (FGLT) data*: analysis of the changing apparent weight of a fuel stringer during refueling that can show channel distortion whilst the reactor is at power.

- *Control Rod Movements, Control Rod Braking Times and Control Rod Alarms*: statistics of the movement, entry and braking times when rods are falling into the core. Variations in these times might indicate interactions with the channel wall due to distortion. This data is collected whilst the reactor is at power and during the reactor shutdown process.

- *Fuel Handling Observations*: information provided by fuel route operators who can highlight abnormal refueling events and indicate where used fuel stringers are, for example, an abnormal shape.

In order to make any monitoring system worthwhile, there must be an element of changing behaviour based upon the monitoring process. If values are monitored and stored and never analysed or acted upon, there is little value in performing monitoring at all. The characteristics of the data are therefore important and of the parameters listed, only Fuel Grab Load Trace (FGLT) is considered to be capable of giving advance warning of changes towards an unacceptable safety-related core state.

The FGLT analysis process involves analysing data originally collected for reactor protection during online refuelling. A diagram representing the measurement system is show in Figure 2.8.

The figure shows a cylindrical fuel channel in the reactor into which a fuel stringer is placed. The removal and insertion of stringers is carried out using a hoist fitted with a load cell that measures the apparent weight of the fuel stringer assembly. This apparent weight changes depending upon numerous factors, such as the dead weight of the stringer, gas pressures within the channel and friction between the stabilising brushes and the channel wall. It is possible to factor out some of the pressure and dead-weight effects leaving traces which allow the friction effects to be compared over at least part of the trace. If the condition of the brushes is assumed to be constant, these changes can then be inferred to be due to

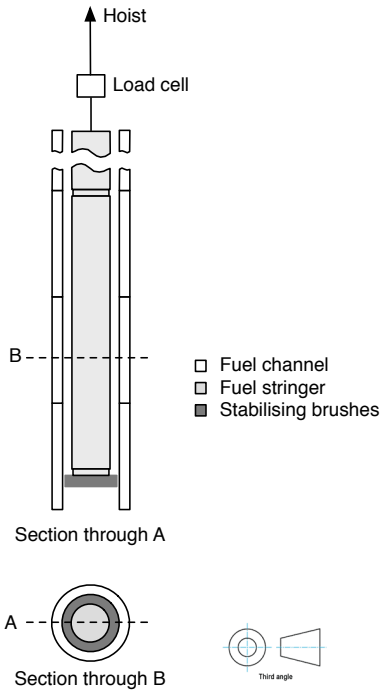

Figure 2.8: Fuel grab load trace measurement arrangement

changes in the channel caused by ageing mechanisms, such as cracks or shrinkage [SWG+09].

The load trace therefore contains valuable information on the current state of the core that can be obtained whilst the reactor is operating. Due to the extensive keying system used in the core, failures are required in a number of bricks in order that the core reach an unacceptable state. FGLT therefore allows early detection of conditions which may lead to unacceptable core distortion.

The standard terminology for monitoring indications within the Health and Safety Executive is that they may be *leading* or *lagging* indications depending on whether they lead or lag the unacceptable state. FGLT is considered to be a lead indicator whilst the others are lagging.

The remaining lagging parameters all remain important as they add to the overall picture of reactor health, particularly as channel power comparisons are carried out every few days and the control rods are in constant motion during normal operation. The capture and correlation of these events may occur after distortion has taken place, being lag indicators, but damage could be limited and safety improved by timely shutdown rather

Monitoring Rating	*Level Description*
Blue	Result indicating that there is likely to be no core distortion
Green	Deviations from normal but that have been adequately explained and do not indicate a core safety issue.
Amber	Observations that could indicate core distortion/displacement – where explained and no core safety issue it is downgraded to green. Where unexplained remains at Amber. Where explained and a core safety issue is confirmed, upgraded to Red.
Red	Observations that unambiguously indicate core distortion/displacement

Table 2.1: MAP Categorisations for Monitoring Observations

than, potentially, continuing to operate for many more months.

In addition to the class one parameters, a number of class two parameters are also monitored – these include environmental reports and any other station information that may be relevant.

Each of the monitoring observations is assigned a severity level which indicates how important that observation is within the context of core distortion. The severity level uses a four-point system as depicted in Table 2.1 [JM08].

The MAP monitors the frequency of these observations, looking for situations where there may be some abnormality, such as a localised cluster of observations, at which time the MAP can order a further investigation into the issue.

It was concluded from experience at the initial MAP meetings that manual recording of hundreds of graded observations per reactor per quarter and subsequent recollection and analysis of this data was difficult and time-consuming to complete manually. Attempts were made to record the data on a whiteboard, but the potential for many observations on each channel and the difficulty of cross-referencing resulted in the conclusion that the efficacy of the process, and therefore the positive impact upon safe operation, would be greater were it supported by a computerised system.

This MAP process is currently underway at two stations, where origi-

nally trialled, with roll-out ongoing for others in the UK.

2.6 Conclusion

This chapter has shown several ways in which nuclear fission can be used to generate electricity, but has also shown that there does exist a single plant item – the reactor – which could, if damaged, result in the whole station reaching the end of its life. The difficulty of fitting additional monitoring devices to the reactors has also been highlighted, along with the key data and information that is already routinely gathered from the core.

Despite having this information at hand, it has historically been used within a relatively narrow scope of control, protection and justification for continued operation, as dictated by the regulatory approach within the UK. A novel contribution offered from this work is the use of existing data as a basis for a robust condition monitoring solution that can simultaneously contribute to safety and improve the economic lifetime of the station. This thesis goes on to present the key research required in meeting this goal through Chapters 3 to 6 followed by a detailed example, in Chapter 7, of how the various aspects of the research are brought together in a working system.

Chapter 3

Structural monitoring

3.1 Introduction

Chapter 2 introduced physical monitoring of nuclear reactors through periodic inspection and outlined the current processes undertaken to ensure safety. The present process, though continuous, operates without visibility of the structural condition of the reactor between inspections. The lack of structural monitoring means that the occurrence of a structural failure leading to unacceptable levels of core distortion could be masked until the the next inspection period or unscheduled shutdown; it may only be diagnosed when control rods fail to fall into the reactor. A contribution of this work is the novel use of condition monitoring techniques applied to existing data as a means of providing a more frequent means of assessing the integrity of the graphite structures.

This chapter describes the key aspects of nuclear reactor operations, the causes of damage within the Advanced Gas-cooled Reactor (AGR) and relevant techniques from the condition monitoring domain that may be applied to this data. One of these techniques, Structural Health Monitoring (SHM), is commonly used for systems which are considered "single use", for example, in systems such as buildings and airframes. This chapter introduces the novel use of the SHM methodology to the monitoring of graphite reactor cores and concludes that the novel application of this technique can provide valuable additional knowledge about the state of the reactor cores in the period between inspections.

3.2 Condition monitoring

Condition monitoring is a general term for the process of appraising the current condition of a component or system and using this information to inform or control some other process. Condition monitoring can be manual, through simple inspection or non-destructive testing on a component, or automatic by measurement of a parameter or proxy parameter which can indicate the condition of a component or subsystem.

A diagram representing the stages involved in implementing a condition monitoring process is shown in Figure 3.1[Bar96]. This shows that the condition monitoring process start with the identification of a critical system. Relevant monitoring techniques must be selected and a method of alerting engineers and operators is required. This diagram also has a simple example of a car braking system appended which is used later to demonstrate the process.

In the example of a car braking system, the simplest check entails removing the brake calliper and inspecting brake pads and discs. This process can then reveal if the braking system is fit for purpose, or if repair and replacement parts are required. Identifying the braking system as critical, because of the potential for loss of life in the event of failure, is the first step shown in Figure 3.1.

In assessing a braking system, the mechanic assessing the system needs to know the requirements for a working brake system; a smooth disc and pads with sufficient frictional material and a means of applying the pads to the disc is essential. Within the overall monitoring system, this knowledge is invaluable and represents a prerequisite for completing the "condition assessment" stage. A mechanic with sufficient experience may be able to offer additional information and indicate that the brakes have, for example, another 5,000 miles of use before needing replacement, or that by replacing the worn pads now, the time required until disc replacement can be delayed by 15,000 miles; a trade-off which can only be considered with the new information provided by the monitoring.

Safety is also an important consideration in this system. Failing brakes will potentially result in a serious accident and potential loss of life so there is a need for a safety margin. In this example, the safety margin would involve servicing the brakes when there is still some braking force

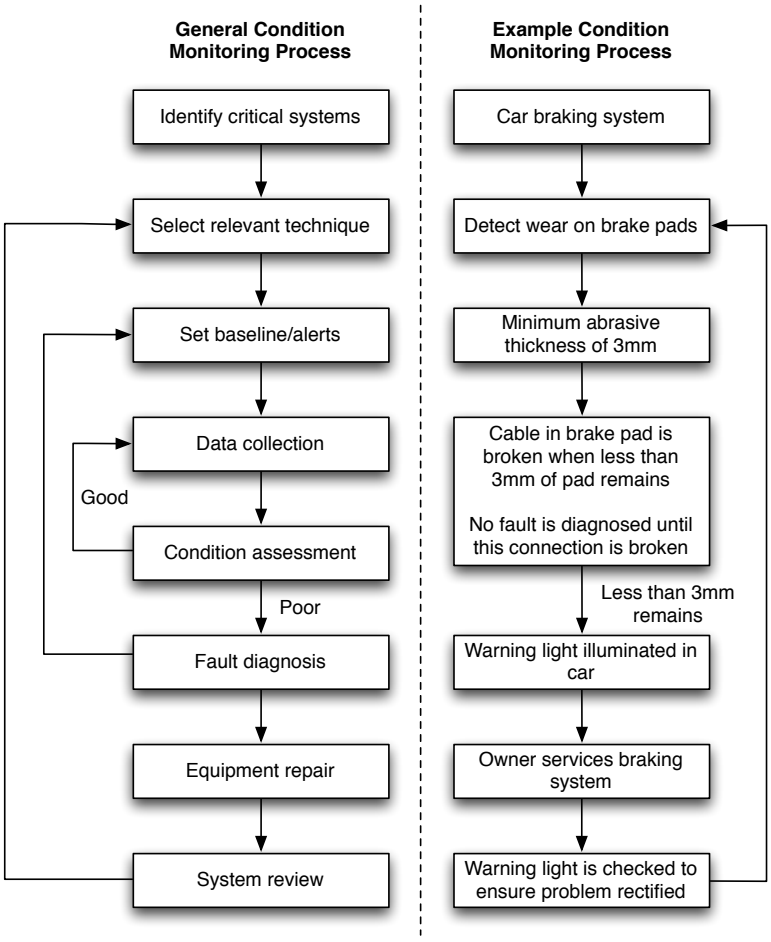

Figure 3.1: Implementing a condition monitoring process [Bar96]

available in the system. Owners should not risk waiting until the very last braking action before replacing components. Fortunately, brakes are often configured with redundancies in place and failure does not tend to occur without prior warning. Cues, such as increased braking distances or having to depress the pedal harder, are all signals that are usually read by an informed driver; a simple form of condition monitoring that could result in booking the car in for a service.

The application of knowledge and consideration of risk and costs associated with trade-offs, like those in the example above, have provoked research into the optimum condition monitoring strategy. Condition monitoring has developed from rudimentary inspections into automated systems incorporating sensors that can gather and analyse data instantly. The automated systems provide operators with valuable additional time to de-

cide and plan for maintenance or part replacement. In the car braking system example above, the addition of sensors indicating excessive wear and problems has removed the requirement for the driver to be aware of the physical performance. Automated systems are beneficial to owners, who then do not replace brakes simply because the car has travelled 20,000 miles; instead the replacement is based on need both reducing the risk and monetary cost associated with replacing parts on a time-based maintenance regime.

The application of good inspection, monitoring and resource planning can form the basis of a robust condition monitoring strategy. The condition monitoring approach employed can drive the maintenance strategy ensuring that it is focused on those assets to which it will deliver the most value; in this case, the technique is often called simply "condition based maintenance" [JLB06].

Within the electricity industry, a key driver for the adoption of new maintenance strategies and techniques to deliver increased uptime, safety and profitability of equipment has been market deregulation [SSM+07]. When a single monopoly provider was responsible for producing and transmitting electrical power, the operators successfully justified investment across the asset base providing considerable redundant capacity. This meant it was easier to schedule downtime on plant and, without competitors, there was less concern about the economic consequences when outages were required. Competition between generators and transmission networks operating under economic regulation means that the cost of redundancy must now be considered. Any savings made in maintenance can then contribute to maintaining company competitiveness and this has pushed condition monitoring forward in many electrical-related areas [TSB+02].

In considering the application of condition monitoring to nuclear reactor cores, three relevant areas are considered:

- electrical transmission and distribution systems which share some of the safety concerns with nuclear power generation,
- electrical power generation which is, in itself, a super-set of the nuclear electrical generation domain, and

- other nuclear-related condition monitoring systems.

These areas are considered in the following sections.

3.2.1 Electricity transmission and distribution

Electrical transmission and distribution (T&D) encompasses the transport of electrical energy from the generator to consumer across a network of lines and devices. By the ubiquitous nature of electricity supply, transmission lines and electrical systems are large, with transmission lines spanning hundreds of miles, and often close to people. This makes monitoring of the entire network using, for example, closed circuit television (CCTV) practically impossible. For this reason, power networks have been designed with protection schemes that can detect faults and isolate parts of the system to protect both equipment and people from such failures.

The installation of devices for the protection of both people near electrical equipment and the equipment itself has long been standard practice. The introduction of electricity markets has seen operators attempt to utilise assets more efficiently and this brought about a new era of monitoring, particularly on large value items such as transformers [SJMM03].

Transformers experience low failure rates of around 0.2-2% per transformer-year [Ben96]. With such relatively low failure rates, the business case for the installation of monitoring must be made on the grounds of cost savings, reduction of downtime and safety.

On the transmission network at major substations, a critical factor is ensuring that there is sufficient network capacity and redundancy to provide a suitable level of service. Coupled to the cost of transmission-capacity equipment and long-lead times for replacements, the case for monitoring can often be made solely on the business risk of losing access to the asset.

The case for monitoring may, however, change significantly when catastrophic failures are considered. In these cases, damage may not be limited to a single transformer unit; an entire substation could be damaged requiring far more extensive replacements and, where such substations are located in heavily populated areas, the risk to life is far greater.

In either case, a monitoring strategy capable of indicating the onset or development of a fault and allowing de-rating or further investigations to take place in a planned manner can prove extremely valuable.

Within the electrical domain, there have also been efforts to increase the degree of switchgear monitoring. Research around 1990 included fibre-optic monitoring [Jon90]; development at this stage focused on sensor technologies. Subsequently, vibration and trip-coil currents were identified as being viable and hardware platforms for collecting such data were made available [Bea96]. More recently, techniques to automatically analyse switchgear operations in the field have been developed. These systems were deployed to quickly identify problems and thereby reduce failures and maintenance costs [SSM+07]. Commercial solutions are also available that can remotely monitor radio frequency signals emanating from equipment across large substations to provide an "early warning" of failures [PMG+09].

The current state-of-the-art in automated condition monitoring extends to a significant number of key electrical network devices. Research projects are currently underway considering the condition of transformers utilising partial electrical discharge signatures [CRMM09] and projects are underway to investigate modelling the degradation of internal components such as tap changers [EGS+08]. Commercial products now include current monitoring on circuit breaker trip-coils [Bea96], partial discharge diagnosis of Gas-Insulated Switchgear (GIS) [AGS+05] and dissolved gas analysis within power transformers [Var02][MCdPF10].

Within each of these key areas, there is a need to increase the condition monitoring capability to improve the quality of detection, the analysis of the detected data or even the predictive capability of the system.

For power transformers, the replacement cost may run to several million pounds. By delaying the purchase of a new transformer and instead holding a "strategic spare" that can be deployed at any of several sites, cash flow can be optimised so that many assets are not bought and left idle for several years. A complication in this case is that transformers can fail catastrophically and there is a safety issue wherever this happens, which must be factored into the monitoring justification. This safety element alongside the capital value of the plant leads to transformers being readily classified as critical equipment that should be targeted for condition monitoring.

When considering circuit breakers, the issue is even more directly focused on safety. Transformer protection may protect against excursions from normal operation, even towards the end-of-life, but this can only

happen if there are working circuit breakers to isolate parts of the network. The same is true of the transmission lines themselves; if there were to be a line problem then circuit breakers should switch out the required section of line, rendering it safe. Where there is a circuit breaker failure though, such as a sticking mechanism, the potential for damage to other items of plant and people is increased. As such, the condition monitoring of these items of plant is also related to protecting investment in other plant items and reducing the risk to life.

In both the transformer and the circuit breaker cases, there are maintenance strategies that can be undertaken to help mitigate the most common faults. This could result in the replacement of insulation oil or tap-changer contacts in a transformer or lubricating a switchgear mechanism. These are activities which may be expected to be undertaken as regular maintenance and the targeted application of maintenance strategy to ensure that they occur at the right moment can increase plant availability and utilisation, and reduce operational risks.

3.2.2 Steam-cycle electricity generation

Steam-cycle electricity generation takes place in a much smaller physical footprint than transmission and distribution. As a result of the smaller size, it is generally easier to have oversight of all parts of the generation system.

In addition to the trip-coil current, partial discharge and gas analyses for transformer and switchgear monitoring, which may still be used on these devices at generating stations, vibration analysis has played an important part in rotating electrical generation plant. Vibration analysis has been cited as "the oldest type of machine monitoring technique" [Hun96]. Rotating plant is expensive and when the turbines, shaft or generator are unavailable to operate, the power station is unable to produce power. Condition monitoring examining shaft vibration and assessing the signals from vibration sensors in order to detect bearing, shaft and other anomalies has been developed.

Vibration monitoring systems are commonly used to monitor gears, bearings, shafts and panels [Hun96]. In power generation the most relevant items are bearings and shafts and for nuclear generation plants the normal

duty cycle is to run, following inspection, for two to three years before the next shutdown. Whilst the power output throughout the operating period may be reduced at times for operational reasons, the generator shafts are expected to continue running throughout the period. This continuous load over several years could potentially lead to failure of the shaft. With generator shafts spinning at thousands of revolutions per minute, failure could in turn have serious safety repercussions as well as threatening the profitability of the station.

Following market deregulation, generation companies often sell the power they will produce far in advance and as such rely on their equipment being available to meet contracts. In the event that the contracted generator is unable to produce power, they will have to make alternative commercial arrangements to supply their customer, and bear any additional costs. A method of mitigating this risk is to employ a continuous monitoring system, alerting operators in the event of any changes in the vibration signature. Where the monitoring system can provide extra time to make alternative arrangements, such as operating at reduced power until a planned outage or finding the most suitable alternative provider, considerable savings can be realised.

Condition monitoring will detect vibration signature changes and give operators advanced warning about their equipment allowing suitable forward planning of, for example, bearing replacements. The replacement of bearings is normal maintenance for rotating plant and the condition monitoring process simply aids the operator in economic plant utilisation.

3.2.3 Nuclear reactor core monitoring

There is considerable literature on the subject of nuclear reactor core monitoring; indeed, the Organisation for Economic Co-operation and Development (OECD) has run several conferences on the subject of "In-core instrumentation and reactor core assessment" [Age92][Age96].

This review focuses on four systems described at these meetings; an ABB system, Westinghouse's BEACON system, a Dutch "adaptive core monitoring system" and the Korean Core Protection Calculator System (CPCS).

The systems described in all of these papers are, indeed, core mon-

itoring systems, but they are all concerned with the nuclear aspects of the core. As such, ABB's systems for alerting operators when Limiting Conditions for Operation (LCOs) are reached are discussed, along with Westinghouse's BEACON system for generating 3-dimensional core-power distributions.

The condition being monitored in these cases is therefore not the physical condition as seen in the electrical examples preceding. Instead, it is the radiological and power conditions that are being monitored. In the case of ensuring safe operation of reactor cores, the radiological conditions affect the structural integrity of the core, but do so as a cumulative effect. This cumulative effect means that monitoring neutron flux in real-time does not give any insight into the structural integrity of the core and whether it will result in difficulties with fuel or control rod movement.

Condition monitoring in all the systems outlined at the OECD conferences is undertaken with a view to ensuring that power is appropriately balanced across the core and that there is no "tilt" effect resulting in increased burn-up, and hence temperature, in any region. This type of condition monitoring is therefore concerned with ensuring economic use of nuclear fuel and protecting the reactor core.

It can therefore be concluded that existing core monitoring techniques are not monitoring applicable conditions that could be extended to the structural aspects of the nuclear reactor cores. Core construction is, however, radically different between the AGR reactor design and the water-moderated PWR and BWR designs and it is therefore likely that the analyses would vary between the different moderators. As an example the PWR and BWR designs do not feature on-load refuelling and, as such, there is no comparable ability to make direct measurements of physical forces during operation as there is for the AGR case.

This work shall now consider only the graphite reactor cores of the AGR design, but the process outlined throughout the remainder of this chapter could equally be applied to other reactor designs as long as suitable monitoring data was available for analysis.

3.2.4 AGR monitoring considerations

The monitoring systems used in electrical generation, transmission and distribution are proving valuable within their own domains, but in general the monitoring techniques employed are not suitable for AGR core monitoring. There are no electrical currents that relate to mechanical properties as in circuit breakers and there are no partial discharge sources as used across substations. The vibration monitoring employed on rotating plant may be feasible, but there are no sensors available due to the harsh conditions inside the reactor. Additionally, the techniques seen in existing "core monitoring" systems appear to be inappropriate as they are concerned with efficiency of the the nuclear process rather than the appraisal of physical condition.

For any system, the general effect of maintenance is depicted in Figure 3.2. This conceptualisation shows that system health is a downward trend. Once maintained, the condition (or system health in Figure 3.1) is reset to a new level and the system continues with a similar profile to before, although possibly with maintenance occurring at an increasing frequency. Cleaning and maintenance of the system or components may affect the shape of the curve, but the overall shape is representative of most maintainable systems.

The repeated degradation and maintenance can continue until the maintenance costs associated with the device make it uneconomical to continue operating. It may then be better to replace the device.

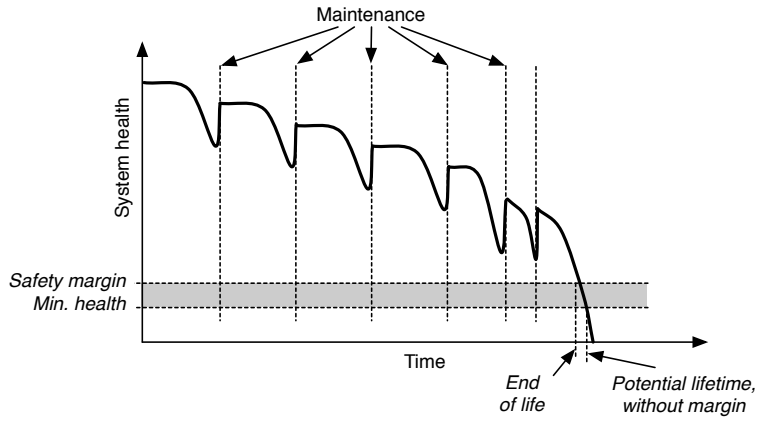

Figure 3.2: System health over time with maintenance

Process	*Transformation*
Radiolysis	$CO_2 \Longrightarrow CO + O$
Recombination	$O + CO \longrightarrow CO_2$
Surface oxide formation	$O + graphite \longrightarrow C(O)\ [surface\ oxide]$
Gasification	$C(O) \longrightarrow CO\ [gaseous]$
Overall reaction	$CO_2 + C \Longrightarrow 2CO$

Table 3.1: Principal processes in graphite corrosion

Having established that the approximate system health curve shown in Figure 3.2 is appropriate for most maintainable equipment, attention is now focused on the graphite cores for monitoring. As detailed in Section 2.2.1, the key roles of a graphite core are to moderate fast neutrons, to provide a physical support structure for the fuel and to provide entry routes for control rods used to both control and stop the fission process.

In order to appreciate the difficulties that may arise due to core damage, the underlying physical processes that result in core damage should be considered.

3.2.5 Damage in AGRs

The operation of the AGRs results in damage to the graphite. There are two primary mechanisms; corrosion of the graphite by the carbon dioxide coolant and direct changes to graphite properties due to irradiation. These are both described here.

The principal reactions which take place causing graphite corrosion due to carbon dioxide, CO_2, are shown in Table 3.1 [LL72]. Normally, the radiolysis and recombination takes place quickly, so the oxygen (O) and CO species are short-lived. It is only where the radiolysis takes place close to a graphite surface that surface oxides and gaseous CO form, removing carbon atoms from the graphite surface.

The removal of carbon atoms results in weight loss in the core graphite, particularly near the brick bore. Although weight loss can be combated by adding other gaseous species, such as methane, to the coolant, this is a key ageing process as weight loss could impact many operational issues. Since weight loss is considered such a key issue there are limits on maximum weight loss, after which generation operations cannot continue. This limit

is currently 40% of the graphite mass [Gee08].

In addition to the gradual graphite erosion due to the coolant outlined above, the graphite is itself affected by radiation [NJT87]. The radiation results in a stress profile, as shown in Figure 3.3, throughout the life of the bricks [Jon05]. The keyway root is the outer part of the brick and the bore is the inner part, this is also shown in Figure 3.4.

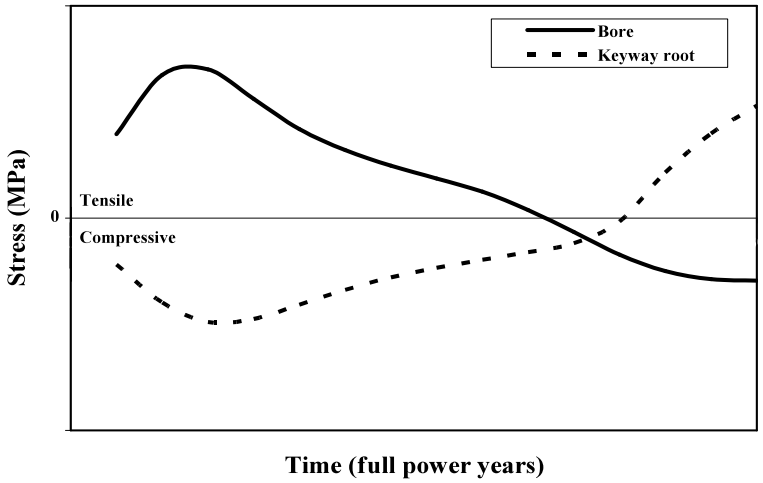

Figure 3.3: Graphite brick stresses at keyway root and bore [Jon05]

Where the bricks are experiencing a tensile stress, it is possible that the surface may open into a crack. Applying the information in Figure 3.3 to an individual brick, the forces at the beginning and end of life can be shown as in Figure 3.4. This shows that cracks are most likely to appear at the bore in the earlier part of their lifetime and in the outside of the bricks towards later life. The point at which cracks may initiate at the outer side of the bricks is after "stress reversal", where the bore and keyway lines cross in Figure 3.3.

Stress reversal is a significant point in the life of reactor graphite. Some of the AGR reactors are now approaching this point and this represents a key driver for monitoring to ensure that the changing stresses do not cause significant cracking which could lead to core distortion.

The most important result of these damage processes, in terms of continued operation, is that the structural strength of the core is weakened. The damage, particularly weight loss, also means that there is less moderation resulting in fewer thermal neutrons. This has a greater effect on

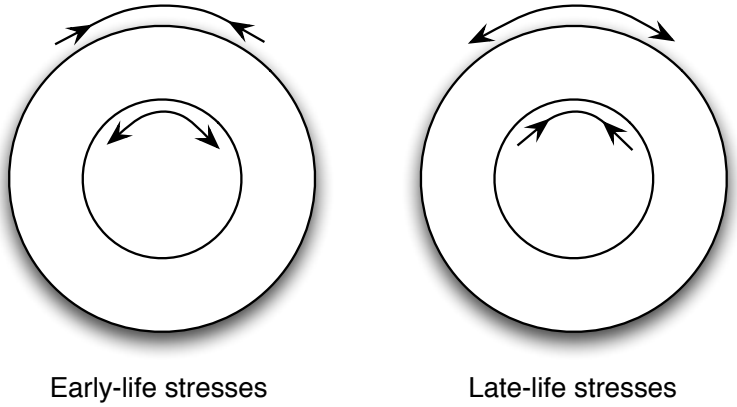

Figure 3.4: Early and late life stresses on bricks

the economics of operation than safety as the level of enrichment must be increased to maintain sufficient thermal neutrons to continue the fission reaction [MRM96].

In terms of repairing graphite, there are techniques that can reduce the level of damage caused by irradiation. Annealing is the principal technique that can be used but this requires heating the reactor beyond normal operational temperatures and no mechanism for this is in place. Even if this were possible, the components around the core have not been designed to operate in the higher-temperature conditions that annealing would require [Rey96]. As such, there is no practical method of repairing any of the damage to the graphite core, nor any means for combating the weight-loss due to oxidation that has already occurred.

Concern over the structural properties of the cores has led to consideration of condition monitoring techniques from the structural domain. The area of structural health monitoring is considered suitable for this application, and this is discussed in the next section.

3.3 Structural health monitoring

Structural health monitoring is a technique that has been applied to "single-use" structures including bridges, buildings and aircraft [FW07]. Whilst these structures may contain a great number of replaceable or serviceable components, in each structure there is a fundamental component, such as suspension cables in bridges, the reinforced concrete core in buildings or

the aluminium airframe in aircraft that cannot be readily repaired. These are all considered to be single use; after a component is judged to be incapable of continuing its role in the larger system, it is considered the end-of-life for the full system. At this time, the whole system – bridge, building or aircraft – will be replaced.

The graphite core within a nuclear reactor is a single-use system. It experiences variations in temperature, pressure and vibration due to the flow of the CO_2 coolant and, as noted in Section 3.2.5, there is continuous irreversible weight-loss. Viewing the core as a single-use, unserviceable component requires that any analysis technique must take account of the fact that the item being monitored is a continually deteriorating component rather than a maintainable or replaceable component such as those discussed in Section 3.2.2.

To consider this, another indicative health curve is shown in Figure 3.5. This curve show the system health against time for the life of a component or system which cannot be maintained. The dashed line shows that degradation will simply continue until the item can no longer perform its function; this is exactly the case within the car brakes example where the health is continually eroded with use, but where the component is discarded when no longer able to perform its role. The result is that those condition monitoring aspects relating to maintenance no longer apply for the component and the justification for the structural appraisal must be made solely on the grounds of risk, and particularly life safety risk.

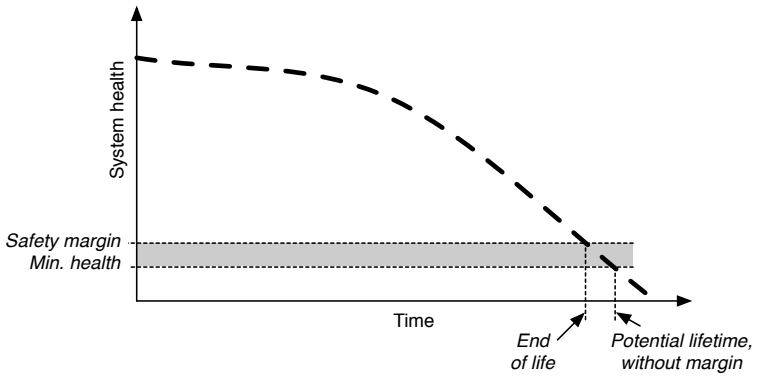

Figure 3.5: Health curve where maintenance cannot be performed

The introduction of risk is important. If there are no significant conse-

quences of a failure, there need not be extensive monitoring undertaken. If there is a risk of loss of life, large capital or remedial costs, or cascade effects to other plant, condition monitoring should be considered. This is the reason that SHM is typically seen in large systems such as buildings, bridges and aircraft; all have the potential for loss of life or significant economic impact.

It is important that structures such as these have a margin of safety to avoid potentially catastrophic situations. If the *minimum health* limit in Figure 3.5 is actually the lowest limit of health that the structure can reach and still perform its function, another line can be added above this representing a safety margin. As this safety margin will require a greater level of system health than the minimum level, the effect of this will be to remove the structure from use before reaching the lower limit. The effect of removing, or indeed any reduction of, this margin can be seen to be an increase in the usable life of the system.

Finally, these curves cannot be fully predicted in advance as the operating environment can change the degradation pattern. For this reason, most SHM systems include an element of modelling degradation and a testing element which allows the present condition to be compared against an end-of-life condition [FW07].

This scenario is comparable with that of the AGR power station. The graphite is weakening, just as suspension cables on a bridge corrode, and as it does, the Remaining Useful Life (RUL) of the reactor reduces. Furthermore, the safety aspects associated with bridges, buildings and aircraft are equalled, if not surpassed, by the potential for nuclear accidents. This means that safety margins, although conservative, cannot simply be reduced to prolong station life. The modelling in the SHM systems considered here, in terms of looking to identify key trends and changes to the operational state, also appear to map well onto the requirements for an AGR monitoring system. The justification for this will be presented in Section 3.3.1.

It is known, from inspection information, that the current condition of each AGR is different. This is even true for different reactors at the same station, and for those at sister stations – stations that were built around the same time to a similar design. The present condition depends upon the quality of the original materials used to construct the core and encom-

passes changes during operation due to operating conditions such as power output, coolant chemistry and operating pressures and temperatures.

The differences between reactors suggest that modelling should be supplemented with evidence gathered during operations and inspection, and SHM appears to be a valid technique for this purpose. To demonstrate the applicability of the SHM technique in this particular scenario however, the requirements for such a system must be considered and work undertaken in other areas examined. Considering ways in which a graphite core might be monitored using this technique is also necessary and the next section addresses these issues.

3.3.1 The SHM process

Structural health monitoring is defined by [FW07] to be a four-stage statistical pattern recognition process. These stages are:

1. *operational evaluation*: the process of deciding why SHM is suitable and how it can help with the operational issue,

2. *data acquisition, fusion and cleansing*: the process of choosing how to gather data and ensure that sensed data is relevant,

3. *feature selection and information condensation*: the process of choosing what should be trended and monitored, and

4. *statistical model development for feature discrimination*: the "decision factor" that prompts some action.

There are key challenges at each stage of applying this 4-stage approach to the nuclear domain. These include obtaining data in a suitable format for processing using a SHM-based approach and how to normalise and select an appropriate set of all available data. These challenges will be considered more in Chapter 4 but prior to this it is useful to look at three other areas which encompass similar investment, economic and safety concerns. These are the monitoring of bridges, aircraft and nuclear containment buildings using SHM.

3.3.1.1 SHM for bridges

Bridges represent a large capital cost and contribute to the economic prosperity of whole regions and as such they are designed to have a service life measured in decades, with some bridge designs now having expected lifetimes of over 100 years [LOZ+06]. Throughout their lifetime, bridge components will age and possibly be replaced; however eventually a life-limiting factor, such as the structural strength of the main towers, will result in it being necessary to replace the structure.

As is the case with other condition monitoring applications, there will be an optimum time where the structure should be replaced. This optimum time will be a function of safety margin, the residual strength (system health), the traffic volumes on the structure, and the risk posed by a catastrophic failure. The risk takes into account not only potential loss of life on the structure in the event of a sudden failure, but also the cost of using alternative routes whilst a replacement is designed and constructed; a process which may take many years. For example, when a short section of the MacArthur Maze freeway interchange collapsed in Oakland, California, the economic impact was estimated at between $4m and $6m per day [Oak07]. Given the economic consequences of a bridge collapse without alternatives, consideration may be given to extensive monitoring schemes for bridge structures to avoid such large costs.

Traditional means of ensuring continuous safety for such structures is routine inspection of key components [OBM06], as it is for nuclear reactor cores. With increasingly long structures, representing ever higher capital cost, the relatively small incremental cost of monitoring systems has considerable potential benefits. Furthermore, as the cost of sensing and data handling equipment has fallen, the case for fitting monitoring systems to smaller structures has become stronger [OBM04].

Techniques for monitoring bridge health include both system-wide monitoring, allowing movement in the bridge deck and towers to be monitored using Global Positioning Systems, and localised monitoring at key points using technologies such as strain gauges and vibration sensors [LOZ+06]. The Singapore-Malaysia Second Link is approximately 1.9km long with 27 individual spans and a main span length of 92m. The monitoring strategy employed here uses 12 strain gauges, 12 pressure cells, 44 thermocouples

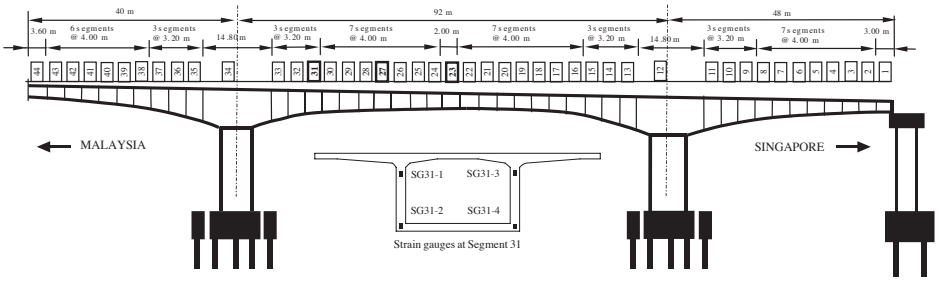

Figure 3.6: Example bridge monitoring system [OBM04]

and a triaxial accelerometer all monitoring the main span. A diagram showing the locations of the sensors in relation to the main span is shown in Figure 3.6 [OBM04]. Some of these sensors, such as the temperature sensors, exist to allow any changes in sensed data to be adjusted for ambient conditions.

It can be seen from the diagram that the monitoring has been spread across half of the main span, and from the description that several different data types are being collected.

After data collection, the analysis continues by processing the data initially in a training mode and subsequently in an operational mode. A multivariate statistical technique is commonly used for this analysis. The multivariate distance commonly used in SHM implementations is the Mahalanobis distance. This allows multivariate data with different scales and units to be analysed together. Having calculated the multivariate distances, the final stage is to apply a threshold level, calculated to give statistical confidence, to this distance to conclude whether the new observation is novel – beyond the threshold – or normal.

Omenzetter [OBM04] applied the SHM technique developed using measurements gathered during construction; this meant that there were significant sudden changes in the measured data as, for example, a new span was erected. Each of these was visible within the clustered data and the conclusion was drawn that the Mahalanobis statistical distance could be used to successfully detected abnormal post-construction events.

Brownjohn, in [Bro07], also addresses two other bridges which were instrumented to provide validation of a bridge simulation. This monitoring, on the Humber Bridge in the UK and the Bosporus Bridge in Turkey, re-

vealed that deck vibrations are sensitive to damage in bearings and hanger components. Brownjohn recommends that sensors could be added to detect this component failure.

The consideration of the bridge monitoring systems has some important insights; bridges have similar lifetimes to reactor cores and have several replaceable components in addition to the main structures. The techniques used in this domain use statistical distances to allow consideration of data from different measurement systems in a holistic manner and have shown that detection of novel modes of operation, possible indicative of failure, is possible.

3.3.1.2 SHM for aircraft

Whilst the economic case for undertaking monitoring on bridges may be high, the monitoring undertaken on aircraft has a higher life safety justification; the airframe of an airliner is ultimately responsible for all those inside. Airframe monitoring is currently undertaken in much the same manner as for bridges, with extensive inspection at periodic intervals. This is time-consuming and costly for operators [MKB+02].

Against the backdrop of a desire to reduce routine inspection, some aeronautical structures are now operating beyond their original design life [GZJB02]; again this is a good parallel with graphite cores case which are also operating beyond their original design life. Another similarity with nuclear operations is that airliners have used off-line models of degradation and operations to characterise likely deterioration in the airframe and, on this basis, the inspection schedules are devised [MKB+02].

SHM technology is far less mature in this area than for bridges, but is following a similar path to that described in the previous section. In this case, due to the ability to access the airframe, it has been suggested that new sensor technologies are integrated within existing aircraft. The data from the measurement systems can then be centrally collected, as shown in Figure 3.7.

The example airliner monitoring system is drawn from Giurgiutiu et al [GZJB02]. This work is interesting because, although demonstrating that cracks forming around rivet holes can be identified, it is based upon novel measurement and sensing technologies. Unfortunately, whilst this area has

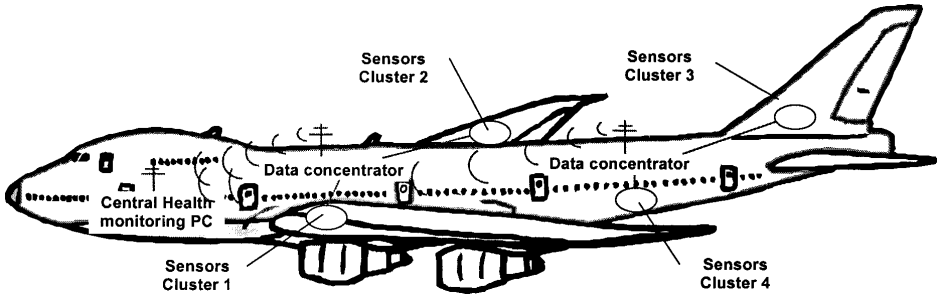

Figure 3.7: Example airliner monitoring system [GZJB02]

useful life safety implications for comparison, Section 2.4 discusses why the approach of adding new sensors is not suitable in the AGR case.

The NASA report produced by Munns et al [MKB+02] studies the subject more deeply, and provides a roadmap for the further development of their technique. This suggests several prerequisites of an SHM system and recommends that the monitoring and inspection be linked. Significantly, they also recommend studying additional sensor techniques to improve the capability.

3.3.1.3 SHM for nuclear buildings

A review of structural health monitoring for civil infrastructure [Bro07] considers several applications of SHM, including nuclear power plants. In this review, it is noted that the main measurand of interest is temperature. Other performance data, such as that from strain gauges, plays a less important role. The conclusion in this paper is that "in the UK at least...online monitoring of structural response so far does not play a major role in tracking the health of the [Pre-stressed Concrete Pressure Vessel]".

Despite this relatively negative statement on SHM in the nuclear domain, work has continued internationally to address ageing containment structures in the United States under the auspices of the Nuclear Regulatory Commission (NRC) [NOE+98]. This supports the view that although formal SHM techniques are not currently in common use within the nuclear industry, the general trend is to increase the monitoring of critical structures. This confirms that investigating the application of SHM tech-

niques is a reasonable and valid approach.

3.3.2 SHM and AGR monitoring

Having investigated several other areas of application for the SHM process, it is concluded that the process is suitable for structures such as graphite cores. The most significant challenge for the AGR monitoring is obtaining data and subsequently processing this data to a suitable format for monitoring.

The solutions presented in the SHM literature reviewed in Section 3.3.1 all used sensors capable of relatively continuous monitoring. Whilst these techniques are effective within bridge, building and aircraft monitoring, sensor technology within the particularly harsh environment of the nuclear reactor is not mature.

Despite the difficulty with gaining data from sensors, the use of disparate analysis techniques within SHM using measurands such as position, vibration and strain is analogous to the disparate manual techniques being used to assess the graphite reactor cores. This means that the technique can be applied to the data described in Section 2.5.1 allowing it to be analysed.

It is also recommended that an extensible strategy for analysis should be employed which can integrate the results of future research into the analysis of the currently available data.

3.3.2.1 Stage 1: Suitability of SHM

Having investigated the role of condition monitoring in both maintainable and non-maintainable plant items it is shown that there is value in the application of structural health monitoring in the context of the graphite cores structural properties.

It is suggested in [SFHC02] that four questions should be answered at this stage. These relate to how damage is defined for the system, the conditions of operation of the system, the limitations on data acquisition and the economic or life-safety implications for undertaking the monitoring.

The clearest definition of damage for the system is that the system has failed in the event that the core has distorted and control rods or fuel cannot be moved within the reactor system. Thus, the failure is not the damage to individual failure of components, such as bricks, through cracking, but

rather the effect of cracking allowing further movement in the system that prevents safe operation. The aim must therefore be to consider how the damage to component parts, that may not be threatening in themselves, relate to overall system damage.

The conditions of operation of the reactor system are relatively well known. As introduced in Chapter 2, the normal mode is for the system to have a thorough inspection and to then run through a pre-defined period until the next inspection period. Furthermore, the operational conditions have been identified as preventing the provision additional sensors in Section 2.4. This introduces difficulty in the next key operational evaluation question of data acquisition. This particular issue is described, and a solution is presented, in Chapter 4.

The final operational evaluation question of life-safety and economic justification is taken as a strong indication of the benefit of an SHM approach due to the increased risk to human life of operating nuclear reactors in the damaged conditions outlined previously.

Overall, the SHM technique has been applied with success to structures with high economic values, such as bridge structures, and those with high safety requirements, such as airframes. A key challenge is recognised as being able to acquire data that can be analysed in a suitable manner, but subject to being able to resolve this problem, SHM is a suitable technique for core structural assessment.

3.3.2.2 Stage 2: Data Acquisition, Fusion and Cleansing

As highlighted when considering the suitability of SHM, the data provided by the Monitoring Assessment Panels is relatively sparse compared with the data from sensors, such as vibration sensors, that can provide multiple data points every second. The most frequent data is channel power discrepancy values, which are carried out at least weekly, but not more than twice per week. Additionally, for observations such as Fuel Grab Load Trace, the same channel might only be checked every few years. In terms of frequency, this is considered to be analysis of discrete events rather than continuous data that an array of sensors might provide.

In order to perform data fusion on the discrete events, a method of combining the temporal and spatial location of the events should be used.

The selection of appropriate time windows and space is considered in Section 4.4.

The final aspect, that of data cleansing, is not considered as part of producing results that are conservative. Data could be excluded for a variety of reasons. One such example would be channel power comparison data that is incorrect due to a faulty thermocouple. In this case, removal of the data point through cleansing may be correct, but it is also possible that the reason the thermocouple is faulty is core-condition related. Thus, the inclusion of all available data is a more conservative approach to damage assessment at this time and the later removal of data should be considered carefully in this context.

3.3.2.3 Stage 3: Feature Selection and Condensation

As highlighted in the previous section, the available data limits the selection of appropriate features for analysis. Any analysis will be limited to the discrete event data that is available, though there are many aggregates that could be computed and trended. The choice of appropriate features, and of the statistical model to use, is developed in Chapter 4.

3.3.2.4 Stage 4: Statistical Model Development

The final stage in the SHM process is one which tries to answer the question of significance of findings. This aspect is particularly difficult when assessing a reactor core design where there is a worldwide population of fourteen, none of which have failed or been decommissioned.

Despite this, the inspection information that is provided every two to three years may give additional information against which the model can be tested. It is a recommendation from this work that the outcomes of the approach be considered against the inspection information when available to identify the quality and robustness of the approach.

3.4 Conclusion

This chapter examined the role that condition monitoring has traditionally played within the electrical power industry and compared the applicability of these condition monitoring regimes to the case of monitoring graphite

reactor cores. The inability to perform maintenance on these cores reduces the usefulness of more commonly used techniques within the domain, so alternative techniques from the structural domain have been investigated where such monitoring is more common.

One of these, Structural Health Monitoring (SHM), has been identified as a suitable technique for the analysis of the reactor core monitoring data. As such, an SHM system for AGR monitoring should allow for the existing monitoring techniques and the outcomes determined through the MAP process (Section 2.5.1) to be utilised and analysed but remain flexible enough that, as further techniques relating to core monitoring are developed, they can be added to the analysis system.

Despite it being possible to undertake health monitoring on the basis of the available data, there is a clear recommendation for additional sensing technologies within other areas and, despite the difficulty associated with the in-reactor environment, this is something which should be considered for future reactor systems.

Chapter 4

Graphite core data analysis

4.1 Introduction

Chapter 3 introduced the technique of structural health monitoring and claimed that this technique would be suitable for the development of a condition monitoring system for graphite cores.

The aim of graphite core monitoring is to provide insight into the current condition of the cores and to assist in assessing their ability to support fuel and provide safe access for the control rods.

The core monitoring is intended to supplement the existing inspection approach and as such the key aim is to detect changes in the operational state of the core. This approach is founded on the basis that where there is a significant change in the core structure, the change will be visible within the monitoring data and the MAP, described in Section 2.5.1, will appropriately grade the data. In the event of such a change being detected, a process is in place where the Technical Safety Support Manager (TSSM) at the site is notified and further action is taken as required. This may involve immediately calling a Monitoring Assessment Panel, recommending a channel for inspection at the next outage, or taking a reactor off-line for immediate inspection.

There are good reasons to support that distortion might be detected by the different analysis processes undertaken upon core data. A brief outline of the principal observations and how distortion may be detected is provided in the next section.

Having collected the observation data, the data must be trended in

some way to identify changes in the core state. This technique should not be prescriptive and search for specific signatures as this would require detailed knowledge of all fault conditions, which is not available. Instead, the analysis should detect significant changes as this allows detection of states which have not been hypothesised in advance. This is important as the ageing processes within the graphite change over time. These processes were outlined in Chapter 3.2.5.

4.2 Inferring distortion from observations

Section 2.5.1 introduced four Class 1 parameters that are collected and monitored as part of the Monitoring Assessment Panel. These are:

- Thermal-Neutron Channel Power Ratios,
- Fuel Grab Load Trace (FGLT) Analysis,
- Control Rod Braking Times and Control Rod Alarms, and
- Fuel Handling Observations.

Of these, the thermal-neutron channel power ratios are collected at least weekly during operation, the FGLT and fuel handling observations are collected on each refuelling and control rod brake times are collected at each reactor shutdown. Control rod alarms are generated by the control systems as an when required.

The primary observations to consider are the Fuel Grab Load Trace (FGLT) and channel power observations. These events are logged routinely during on- and off-load refuelling operations and during regular comparisons of thermal (output) power and neutronic (input) power. FGLT analyses can detect physical changes in the fuel channel geometry whilst the thermal-neutronic comparisons will be capable of detecting changes in gas flow caused by core distortion. FGLT data is gathered every 3 to 6 years per channel whilst thermal-neutronic comparisons are gathered at least once per week during operations. The bulk of available data relates to these observations, though the *blue* observations, as detailed in table 2.1, are only captured for fuel grab load trace events.

As noted above, additional data is also available from control rod drop tests. These tests are simply statistical tests undertaken on data gathered when rods are lowered into the reactor when it is being shut down. These tests will identify when a control rod is moving slowly, which may be indicative of contact with the channel wall. Since there should be no contact, this would suggest distortion. Although this data is not regularly available, unplanned trips do occur and if this data is available, it can be factored into the analysis.

The aim of the analysis, in line with the SHM approach described in Chapter 3, is to detect a change in the operational mode of the reactor by detecting changes in the pattern of the captured data. A significant shift in the distribution of the data would then suggest a change in core condition. A change in frequency of events, for example, may indicate that the core has started ageing and degrading at a different rate whereas a change in distribution, for example a clustering of observations in one part of the core, might suggest that there is a localised defect in that region.

Chapter 3 introduced the concept of structural health monitoring as being suitable for graphite core data, but it was noted that the available data would need to be represented in a suitable manner to allow the analysis. The next section examines various options for using the available data within a SHM framework.

4.3 Statistical analysis

Statistical measures of distance can be applied to many problems. Where there is data that has comparable units, the Euclidian distance is normally used. This allows a direct measure between any two points within the space to be made. A consideration of the available data reveals this to be an inappropriate measure though.

The data fields available upon which analysis can be undertaken are restricted as this is limited to those fields captured by the MAPs. The fields captured for monitoring data are listed below.

- Station and Reactor identifying the core
- The type of observation being made

- The channel number on which the observation was made
- The date on which the event occurred
- A MAP grading (Blue, Green, Amber, Red)

Considering the available data outlined above, the station and reactor are identifying variables and are thus not considered for analysis. These may however be useful in the future when comparing different reactors and stations. The type of the observation being made is a non-numeric value. This cannot readily be handled within multivariate space; any arbitrary assignment of values would be meaningless. One approach might be to consider each data type as a separate analysis, but this would not highlight situations where multiple measurement techniques support a distortion hypothesis.

The channel number could also be considered to be arbitrary, but since these relate to a position on a two-dimensional lattice, these have been resolved to an X and Y coordinate value. As the stations were built to different physical dimensions with different pitches between channel centres, the X and Y coordinate units are specified as a number of pitches, numbered from the centre channel being $(0, 0)$. This feature allows data to be compared between stations in the future should this be deemed meaningful and useful; a potential avenue for investigation being the fleet-wide effect of the increased neutron dose towards the centre of all AGR cores. An AGR core layout is shown in figure 4.1; in this diagram channel MN24 represents the $(0, 0)$ location and the distance between adjacent large circles, representing fuel channels, is two pitches. Even-numbered pitch distances are the smaller interstitial channels and odd-numbers represent the fuel.

The event date has also been normalised to be on a linear scale; for the purposes of analysis, the number has been converted to the number of seconds since the Unix epoch. This means that each time is expressed as the number of seconds since January 1^{st} 1970.

The MAP gradings, also arbitrary, cannot readily be given meaningful numeric values assigned for this analysis. Arbitrary values of 0 for blue, 25 for green, 50 for amber and 100 for red were chosen, but there is no equivalence in these values; for example 2 amber observations are not equal to a single red.

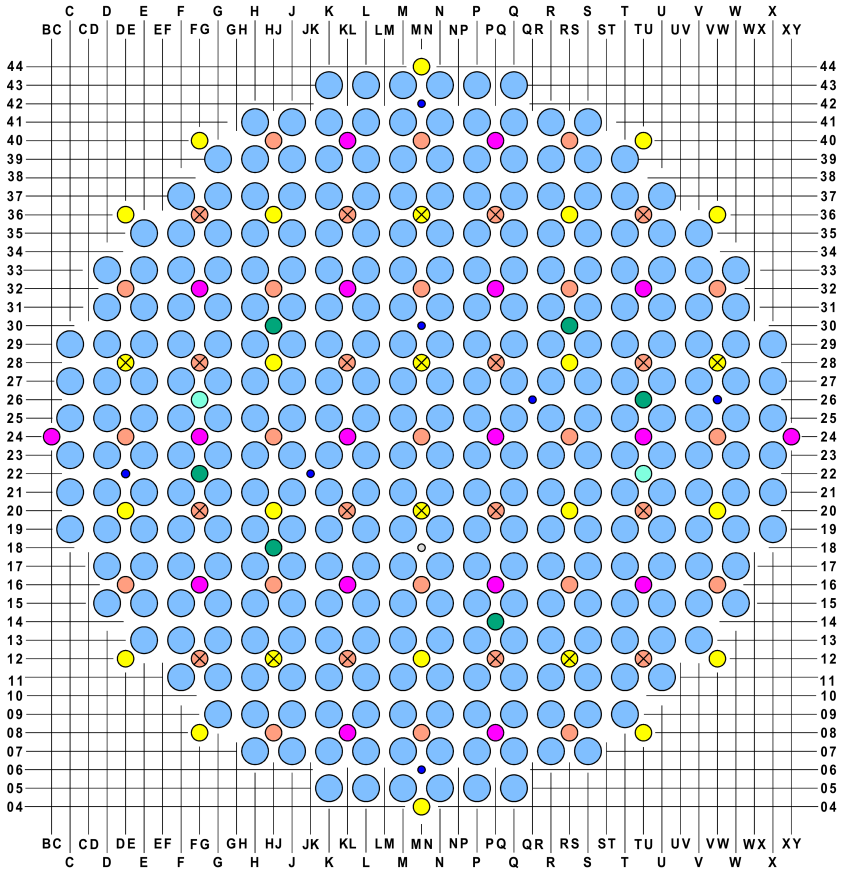

Figure 4.1: AGR core plan

Prior to analysis, the data have been plotted using a simple scatter plot. Figure 4.2 shows data from 18 MAP meetings plotted with the channel represented in the X and Y axes. The MAP number, incrementing from 1 to 18, represented in a shallow Z axis with the position on this axis also represented by various shades. The chart shows 958 monitoring observations in shades from red to black with redder observations occurring earliest. This dataset has been chosen as being from the reactor with the highest number of observations available for analysis.

There are no obvious trends in Figure 4.2; this distribution serves only to provide confidence that the majority of the core area has been examined and that there is a good distribution of data. It can be seen that the shape resembles that in the core diagram of Figure 4.1. In order to perform a rudimentary assessment of the core, all "negative" data are plotted.

Negative data has been assumed to be anything non-positive, or not

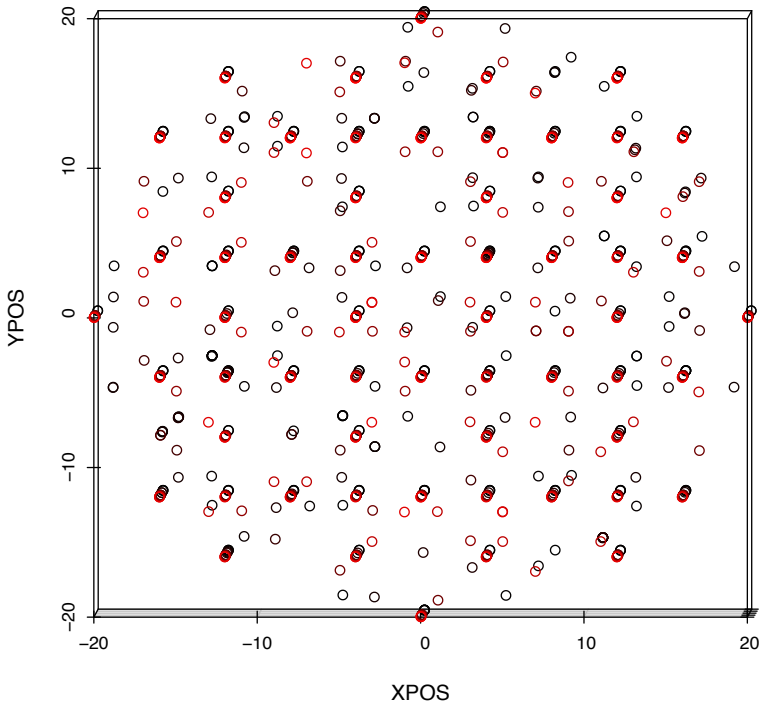

Figure 4.2: MAP observation data

graded *blue* even though the MAP grading system outlined in Table 2.1 grades negative events with known reasons as *green* and considers only *amber* and *red* events to be indicative of core distortion. Although many of the *green* observations could be due to well understood phenomenon, such as anomalous channel power discrepancy values shortly after refuelling in a nearby channel, continuing to count these "explained" observations is a conservatism that is more likely to detect anomalous behaviour.

The negative data, those items logged as *green*, *amber* or *red* observations, are plotted separately and can be seen in Figure 4.3. This is a subset of 154 out of the 958 total events corresponding to all those graded green or amber. There are no red events logged on this core.

Removing the positive data reveals, visually, that there are potentially a small number of clusters around which there are a significant number of negative events. This can be ascertained without considering what event caused the report and can be readily identified by performing a count of data at each location.

The result of plotting count data versus the X and Y position within

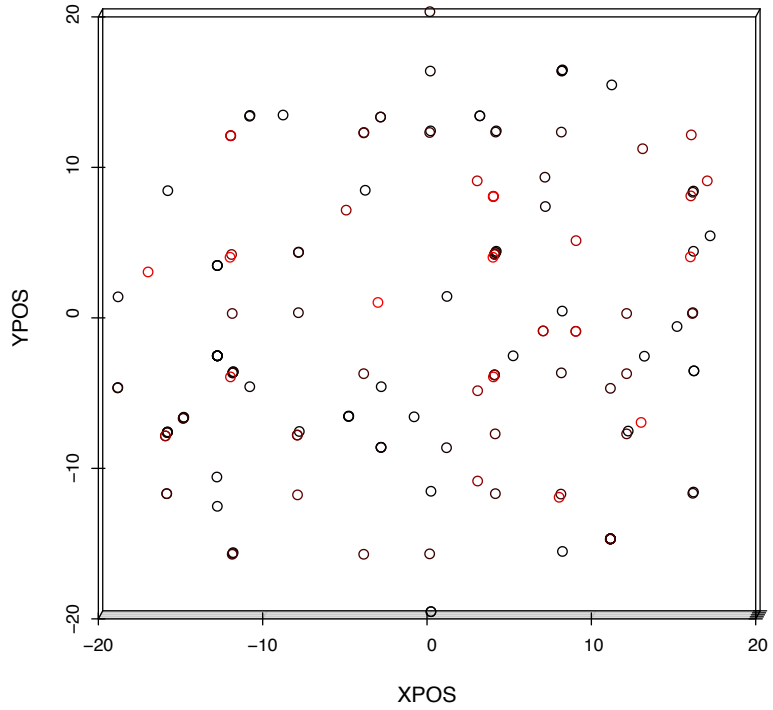

Figure 4.3: MAP "negative" data

the core can be seen in Figure 4.4.

Counting the number of negative observations on each channel is a potential means of assessing the core condition; there are readily visible peaks in the data which could correspond to distortion. Equally, however, there are channels which are inspected on every inspection outage in order to assess the effects of ageing on the same channel. These repeated inspections also give rise to repeated monitoring events as, upon each inspection, the fuel is removed and hence Fuel Grab Load Trace events can be assessed. These channels, if they have any defects, simply due to the increased surveillance, will see increased levels of observations which might not be representative of core distortion. The repeated inspection of channels would result in any analysis based solely upon count data being unreliable and inconclusive.

The counting method also considers each channel in complete isolation from all those around it. As the core is known to be an interlocking structure, as described in Section 2.2.1, this is not a robust means of assessing the damage at a holistic level. Core degradation resulting in distortion

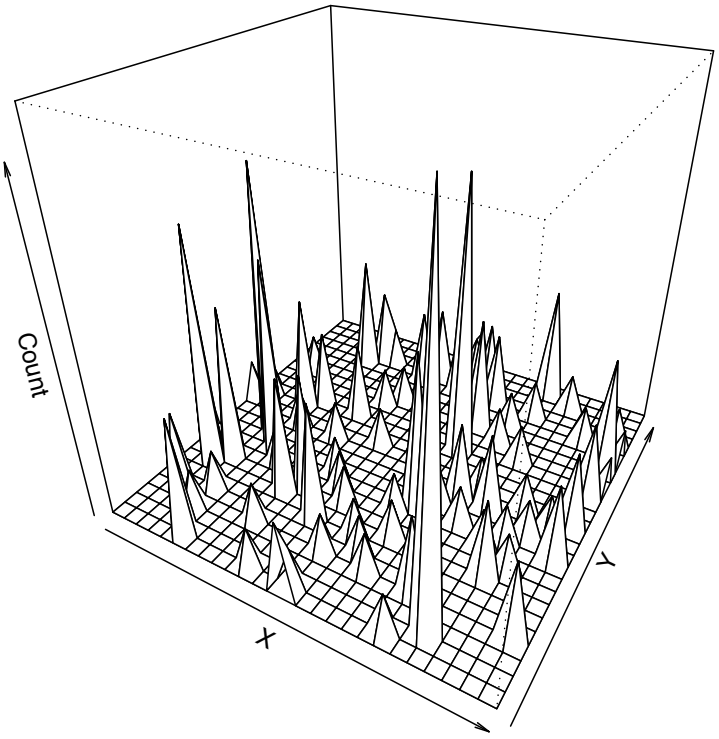

Figure 4.4: MAP "negative" count data

would affect channels within an area of the core. Channels around the centre of the distortion would also be affected until all movement can be accommodated by the slack in the keying system. Although distortion is not known to have occurred, modelling carried out suggests that distortion which may prevent control rod entry would be visible in many fuel channels. This is because the distortion required to prevent the articulated control rods entering the core is in the order of 2 to 3 centimetres based upon the clearances in the control rod channels. and the slack provided by a single keyway clearances is around 2 to 3 millimetres [Ahm87]. On this basis, distortion could be visible up to 10 channels away. Despite this, it is unlikely that distortion would be visible on data obtained from the principal monitoring techniques of FGLT and channel power comparisons at these distances as the effects will get smaller further from the centre of the distortion. A conservative estimate, allowing for lack of visibility in the monitored parameters, is that it may only be visible 1 or 2 fuel channels away. These values are used within this analysis.

The conclusion drawn from these considerations is that a combination of physical distance and the time between observations must be brought to the analysis; true core distortion is likely to manifest itself in FGLT and channel power comparisons simultaneously. The aim must therefore be to detect wherever there is a significant incidence of events within an area of the reactor; the area and period over which the events should be considered therefore depends upon the frequency of data collection and the area of detection.

4.4 Monitoring data coverage

In order to define a temporal and spatial window over which data can be detected, two issues must be considered: the frequency of data collection and the area of the core that can be concluded to be distortion-free based upon any analysis undertaken. As the channel power comparisons and FGLT were considered to be the most significant measurements, these are the most important for this analysis.

The channel power comparison is carried out at least weekly on all channels, so this indicator is readily available for monitoring. The FGLT analysis is less frequent however, and this will be the limiting factor for core monitoring coverage. As a result, the question of how much core distortion can be measured using FGLT in the period between inspections must be raised.

FGLT analyses, as detailed in Section 2.5.1, are undertaken during the inspection period when high-quality inspection data will be available anyway, and also during normal refuelling operations, usually on-load. Since the frequency of FGLT analysis is determined by the refuelling schedule, the rate of refuelling becomes significant. The frequency with which fuel stringers are replaced in an AGR core varies depending on the channel location, the power output and the fuel enrichment, but with a fuel stringer lifetime of around 5 years and with 308 channels, this corresponds to around 60 channels per year requiring refuelling. These are usually refuelled in batches of around 8 channels every 6 weeks.

Before considering the use of FGLT for monitoring purposes, it is useful to consider the monitoring coverage that analysis of FGLT can achieve. This

is a trade-off between the location within the core that is monitored, in this case the particular fuel channel being refuelled, and the amount of the core that can be considered to be healthy by checking that single channel. If the whole core can be considered healthy by checking a single channel then we can deduce whole-core health every time there is a refuelling. If, however, only the channel monitored can be considered healthy, coverage of the whole core will take around 5 years, a period longer than that between inspections and one that could nullify the benefit of performing monitoring in this way.

In order to assess the coverage provided, a number of refuelling campaigns were analysed to reveal that the time taken to monitor the whole core can vary significantly and is dependent upon the distance at which core distortion might be detected within adjacent channels. To analyse the coverage that FGLT alone can achieve, refuelling operations over a $2^1/_2$ year period looking back over 6, 12 and 18 months and assessing the number of channels *not* monitored within a 1 or 2 fuel channels distance were assessed. These correspond to a 3×3 and 5×5 area of visibility; these are shown in Figure 4.5.

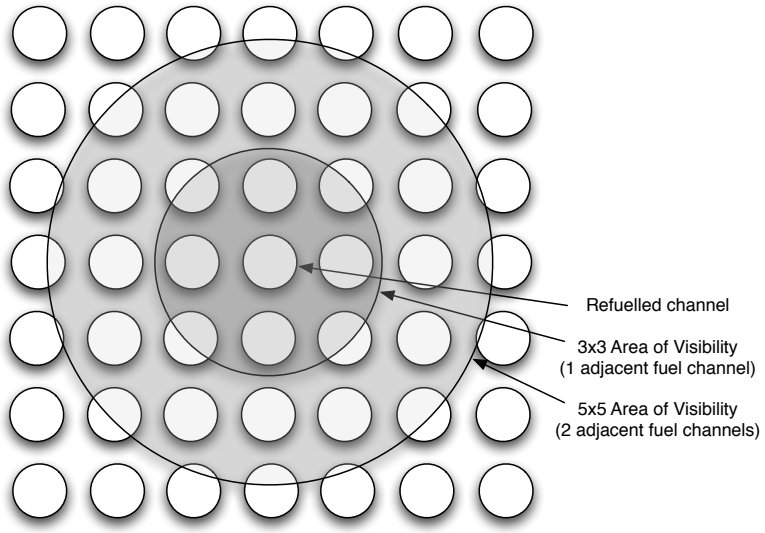

Figure 4.5: Area of visibility around a refuelled channel

The number of channels outside the 3×3 and 5×5 areas of visibility over different time periods is plotted in Figure 4.6. The chart shows that since around June 2008, 90% of channels are monitored within a "2-fuel

channel" distance looking back 6, 12 and 18 months. More recently, complete coverage has been attained of all channels within this distance. The data also shows that more recent data covered about 75% of the core within a single fuel channel, considering only 6 months of refuelling operations.

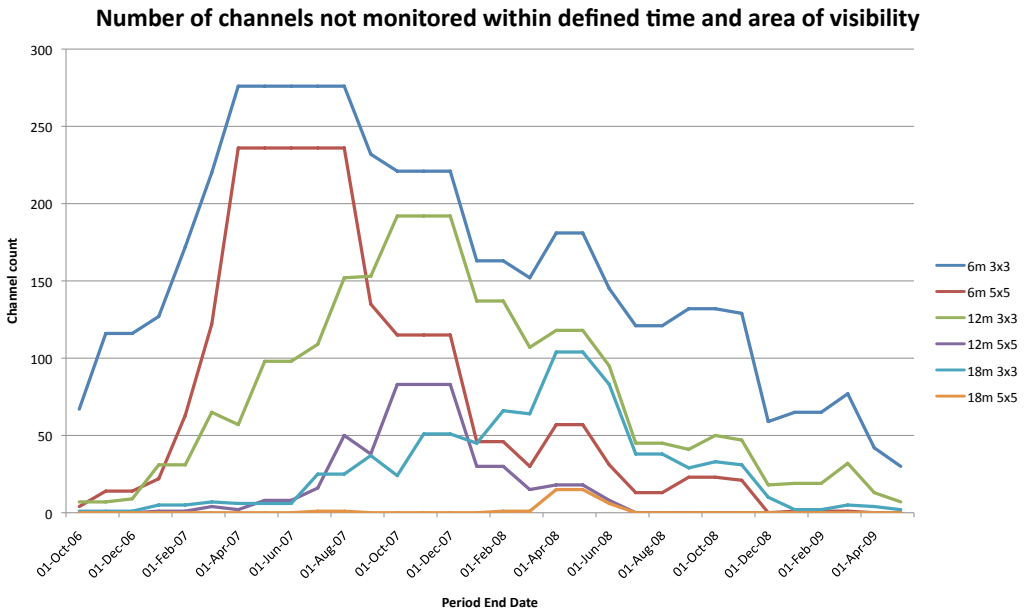

Figure 4.6: Number of channels not monitored against time and distance

These figures suggest that data from any rolling 6 month window covers a significant proportion of the core. This is an improvement upon the 3 year period between core inspections. Despite this result, there are peaks in each trace. Investigation revealed that these were caused by an outage for boiler-closure work which did not see any fuel movements over an extended period. During a normal inspection outage, however, fuel would be moved for inspection purposes. The exceptional event of an outage without core inspections could be accounted for by removing that period from the data and only considering "active" time. Despite this, such events may occur again and it is feasible that the cooling and heating effects of having a core on outage and restarting it causes damage. As such, it is more prudent to use a continuous period, but to realise that where there is a significant outage – of greater than 3 months – that the monitoring is degraded until refuelling operations recommence. This is considered acceptable as additional information from the control rod drop times will

be available from the shutdown process. This should mean that in the period immediately after the outage, the core is in a reasonably well-known state. Where inspections also take place during the outage, the knowledge of the core will be even better.

The discussion in Section 4.3 concluded that a reasonable distance over which to view the core data was up to 2 channels away in each direction, forming a 5×5 grid around any given channel. The discussion in this section has shown that 6 months is a reasonable time period over which to perform the analysis. It is therefore suitable and acceptable for the core monitoring application to use these as limits for detecting related features.

Having established these limits, consideration is given to detecting these within the data.

4.5 Implementing SHM for reactor core data

As noted previously the data in this application cannot readily be analysed using a Euclidian distance due to different units. An alternative was therefore sought and found to be the Mahalanobis distance [Mah36]. The Mahalanobis distance, D_M, is a probabilistic distance, in a multidimensional space, of the multivariate location $x = \{x_1, x_2, x_3 \dots x_n\}^{\mathrm{T}}$ from a group of values with means $\mu = \{\mu_1, \mu_2, \mu_3 \dots \mu_n\}^{\mathrm{T}}$ and covariances represented in the matrix S. The covariance matrix allows for a normalisation of each individual dimension and allows values of different units to be readily compared. The value is calculated according to the formula:

$$D_M(x) = \sqrt{(x-\mu)^{\mathrm{T}} S^{-1} (x-\mu)}$$

The covariances can be calculated by using the distances desired and the values for the X and Y location of the channel, along with the time, can be added to the multivariate vector. This information is available for all observations. For the time being, the type of observation and classification are not considered as these are arbitrary fields and data will naturally cluster around these aspects if included as arbitrary numeric variables. The MAP grading shall be used, however, to identify only the "negative" observations as noted previously.

This leaves the question of what are acceptable values to compare

within the multidimensional space. SHM techniques assess variables in the context of "normal" operation and finds those observations that exceed a confidence interval to be anomalous. This graphite core monitoring application has bounds on both the X and Y directions, due to physical constraints, and the time will only increase. It is therefore proposed that a suitable technique is to measure the distance between all observations and cluster those which are within a specified radius.

There are two ways of undertaking cluster analysis; one starts with data represented within the multidimensional space and combines clusters which are close to form larger, clusters whilst the other uses labelled data to create zones within the multidimensional space corresponding to each cluster. These are called "hierarchical clustering" and "partitional clustering" respectively [JMF99].

It is not possible to use partitional clustering in this case as there is no labelled data to identify when the core condition has reached an unsafe level. This is because none of the AGRs have encountered a significant issue resulting in permanent shutdown.

Instead an agglomerative hierarchical clustering approach should be used, where observations which are close in the multidimensional space are clustered [War63]. This approach starts off by considering each observation as a cluster and measuring the distance between each pair of clusters. The smallest distance is then located and, if below a threshold, the clusters which are closest are merged. At this point a new mean for the cluster is calculated and the analysis repeats until every cluster is separated by at least the threshold distance. Each cluster can then be analysed and assessed. Particularly large clusters would signify stronger evidence of an issue around that point, in space and time, in the core. These clusters can then be labelled and could themselves be assessed and graded.

The analysis process can therefore be summarised as follows:

1. Calculate the covariance matrix for the core data under analysis for use in the Mahalanobis distance calculation

2. Create a "cluster" for each observation in the core; initially each cluster contains a single observation

3. Measure the distance between all clusters

4. If the smallest distance is less than a threshold value, combine the two clusters taking the arithmetic mean of all observations within

5. Repeat until the closest distance between clusters exceeds the threshold value

An example of this, using the data shown previously, is presented in the next section.

4.6 Case study: Clustering using real data

This example uses the 154 observations plotted previously in Figure 4.4. The analysis is performed on all of these observations as a current view of the reactor core data.

The first step in this analysis is to calculate the covariances for all of the available data. The time and date data are converted into Unix timestamps and the channel numbers are converted into X and Y location coordinates, Loc_x and Loc_y. This gives a vector, for each measurement and hence each original cluster, of:

$$x = \{time, Loc_x, Loc_y\}$$

Having established these, covariances between each of the values are calculated. For the dataset used, the covariance matrix is:

$$S = \begin{pmatrix} 1.5049 \times 10^{15} & -4.6239 \times 10^{7} & -4.1460 \times 10^{7} \\ -4.6239 \times 10^{7} & 114.0659 & 10.1258 \\ -4.1460 \times 10^{7} & 10.1258 & 87.7409 \end{pmatrix}$$

Now, each of the clusters, initially of a single item, can be measured to find the distance to all others using the distance formula. Once clusters start to form, the distance will use the mean of each of the time and location data.

$$D_M(x) = \sqrt{(x-\mu)^{\mathrm{T}} S^{-1} (x-\mu)}$$

At this stage a list of distances has been generated and the closest items can be combined into clusters. This process would then continue incorpo-

rating increasingly large distances into the clusters until all items were in a single cluster representing all items. A threshold value must therefore be used, beyond which items are not considered related.

This is calculated using the $(x - \mu)$ term in the Mahalanobis equation. An appropriate time distance has been calculated to be 6 months, or 15,552,000 seconds (assuming 180 days), and both the X and Y locations can vary by up to 2 fuel channels or 4 pitches. Knowing this, a maximum bound for these limits can be calculated using the formula itself.

$$Threshold\ D_M = \sqrt{(15,552,000 \quad 4 \quad 4)^{\mathrm{T}} S^{-1} (15,552,000 \quad 4 \quad 4)}$$

$$Threshold\ D_M = 0.5322$$

For the given data, this value calculates to be 0.5322. It is again noteworthy that this is not a deterministic distance; it simply gives a value for the maximum allowable distance we have chosen when combined into the Mahalanobis distance for the given data. The means that some items, even though outside the 6 month limit could still appear in the data depending on the variance seen within the data set; particularly if there is little distance in the X or Y direction, for example with items on exactly the same channel, this temporal distance may be greater. It should also be noted that this distance is not biased in any direction so the 6 months time limit will be applied in both directions, allowing clusters to span up to around 1 year of data. This is a further conservatism in the analysis.

This fuzzy property is useful though as it allows the analysis to trade time versus distance granting some flexibility in determining the clusters.

When this process is undertaken on the dataset, the first distances calculated are found to have $D_M = 0$; as such the system proves immediately useful in detecting duplicate entries that have been entered onto the system.

The clusters found as part of this analysis are shown in Table 4.1. It can be seen that 137 out of the 154 observations have clustered in some way. Each of these should now be analysed for evidence of core distortion. As this data represents observations from mid-2005 until late 2009, it represents less than 10 potential clusters per year to check.

The largest of these clusters is shown graphically in Figure 4.7. This

Cluster Size	*Number of Clusters*	*Observations covered*
13	1	13
10	3	30
7	1	7
6	1	6
5	4	20
4	5	20
3	9	27
2	7	14
	31	137

Table 4.1: Clusters formed during data analysis

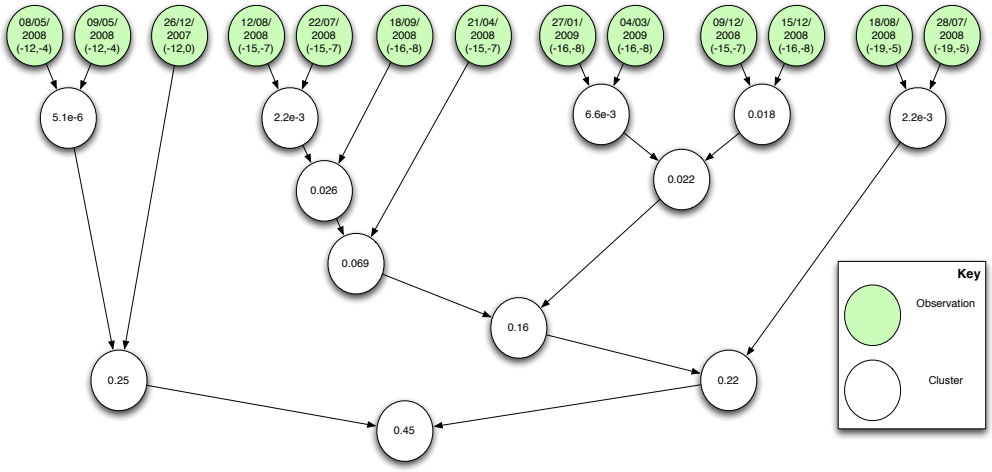

Figure 4.7: Graphical representation of largest cluster

diagram shows the different timescales and X and Y distances involved in each cluster. The mean X-Y location for the cluster is at $(-15.2, -6)$. When this is compared with the 3-dimensional count plot in Figure 4.4 it suggests that the most significant set of observations, including temporal information, is towards the left hand side rather the tallest peak in the foreground.

Section 7.2.4 provides a discussion on the clusters identified using this technique describing the main clusters identified by this approach and explores the possibilities of further analysing the clusters to filter those that are unlikely to indicate core distortion.

4.7 Conclusions

This chapter has shown how the Mahalanobis distance, commonly used in SHM implementations, can be used for the graphite core monitoring application and how the Mahalanobis technique can be configured through covariances to have an appropriate conservative sensitivity in each multivariate dimension.

This is the first time that an overall structural health assessment of the graphite core has been carried out using monitoring data collected whilst the reactor is running. For the generation company, there are considerable benefits in having this additional knowledge as it can contribute to safety and good management of the reactors.

It has also been concluded that that there is a trade-off between the frequency of refuelling, daily operating parameters of the reactor and fuel enrichment and that flexibility in analysis is required to ensure continuing relevance of the analysis.

Chapter 5 goes on to discuss relevant intelligent system techniques for the implementation of this system.

Chapter 5

Extensible graphite core analysis

5.1 Introduction

Chapters 3 and 4 have identified structural health monitoring as a suitable approach for analysing data from graphite cores. This chapter considers the most appropriate method for the integration of these techniques in a novel reactor monitoring system.

This chapter recapitulates the design aims for analysing graphite core data. Next, having discussed relevant techniques from the field of distributed artificial intelligence, the technique of multi-agent systems is selected for implementation.

After identifying multi-agent systems for this implementation, the chapter looks in-depth at the characteristics of these systems, their suitability for implementing a condition monitoring system and issues relating to their use within the nuclear domain.

The chapter concludes that the data storage aspects of current agent-based systems are insufficient for the flexibility and requirements of graphite core monitoring.

5.2 System options

Throughout the preceding chapters several requirements have been identified. Each of these requirements must be considered in choosing a suitable architecture for analysing core data. The requirements are listed below.

1. The system must be able to analyse of all available data captured during the Monitoring Assessment Panel process at each power station (Section 2.5.1),

2. The system must deal with uncertainty arising from monitoring data which cannot be verified until, at the earliest, the next periodic inspection (section 2.3),

3. Whilst sensor technology has not developed sufficiently yet, the system should be able to process additional data, with different characteristics, that may be made available through additional modelling, analyses or processes (Section 2.4), and

4. The multidimensional analysis techniques from Structural Health Monitoring will analyse the condition monitoring data. There may be further developments allowing the system to deliver additional information; therefore it should be possible to add to and extend the system to use these. (Section 3.3.2.1)

The first of the requirements highlights the overall problem. The remaining three requirements are concerned with providing a system suitable for the long-term analysis of available graphite core data. System design techniques vary in their ability to deliver on these requirements.

Literature in the field of distributed artificial intelligence has been reviewed to find the most suitable architecture for this application. Shaw and Fox [SF93] discuss several classes of system using Distributed Artificial Intelligence (DAI) for decision support. They identify that DAI systems generally consist of "problem-solving agents collaborating in finding the solutions to given problems" and go on to describe three types of DAI systems. The three types of DAI system are collaborative reasoning, distributed problem-solving and connectionist systems.

Collaborative reasoning systems have participants that act together to solve a problem, but can reason individually and independently. Distributed problem-solving systems are similar. Instead of each component reasoning independently, the problem is sub-divided and each agent can operate asynchronously on different parts of the problem. This adds flexibility to the system but there is less "emergent intelligence" with intelligence codified within the individual blocks and the problem scheduler.

The final system considered is the connectionist system. In a connectionist system independent agents operate on a problem, but each agent is sufficiently simple to be able to be described as a processing element. The multi-agent platform allows the order of processing to be altered by changing the connections between agents. This serves to alter the data flows through the processing elements. This approach can solve problems and produce coordinated actions but is viewed as the least "intelligent" system as it uses the distributed techniques only as an integration tool [SF93].

A disadvantage common to all of the DAI examples reported in [SF93] is that all of the systems examined utilise a shared knowledge-base, known as a blackboard. This technology was first used for speech recognition and later rationalised to three main components [Cra88]. These are:

1. A shared knowledge-base, or blackboard, which contains all of the knowledge relating to the reasoning process being undertaken,

2. A set of "knowledge sources" which apply knowledge and can change information on the blackboard, and

3. A controller, or scheduler, which decides which knowledge sources should be applied to the blackboard and in which order they should be applied.

The DAI system structure is analogous to experts sitting in a room using a blackboard to solve a problem. Such systems may seem suitable in this case as they are a conceptual extension of the "experts in a room with a whiteboard" strategy, outlined in Section 2.5.1, implemented within a computer system. For the core condition monitoring application, the blackboard would contain a representation of all know information about the reactor core. This could include knowledge such as one of the fuel channels being slightly narrower than others, or of a defect. With further reasoning, additional facts are added, such as degradation having been detected in a particular area of the core. Whilst the blackboard model allows the data to be represented, the subsequent processing steps used within blackboard systems does not fit the application. These issues are discussed below.

In the blackboard system, knowledge sources examine, when the controller allows, the information on the blackboard and apply their own

reasoning abilities. When a participant makes a conclusion, usually only part of the larger problem, they place their part of the overall solution on the blackboard. This new fact is then available to others. This approach breaks down the problem into a manageable size. Different analyses operate at different levels of abstraction with all of the data pertaining to each level available to all participants globally.

This model seems particularly good for the core monitoring application as key data passing through a central point – the blackboard – means that it can be archived at this point and at this time a record can be kept for auditing and verification purposes, in line with the safety requirements for the system.

Corkill [Cor91] claims that blackboard systems are also beneficial in uncertain circumstances and that although absolute determination of a solution is not possible, they can allow "progress to be made". This feature is also useful within the graphite monitoring case as good knowledge of ageing cores is being developed against accelerated ageing models derived from Materials Test Reactor (MTR) data obtained over a much more intensive radiation exposure; the uncertainty between modelled and actual behaviour can then be considered.

A drawback with the blackboard method, however, is that only a single processing element will execute at any given time and that the overall conclusion of the system is being created on a single blackboard. This can reduce conflict in the final system, as information expressed on the blackboard can be considered by the system to be "fact", but it also reduces flexibility. The control shell within the blackboard decides which element to execute, but this decision may be difficult to make and choosing the wrong element could result in an incorrect fact being asserted and all further conclusions being potentially incorrect.

In 2003, Corkill [Cor03] revisited the Blackboard concept and compared it to the emerging field of Multi-Agent Systems (MAS). The comparison between the two systems concluded that the two methods were at opposite ends of the spectrum in terms of processing concurrency, processing distribution and agent lifetimes and the conclusion presented was that where high performance systems were required, a blackboard system was the appropriate choice whereas large-scale distributed applications are more suited to MAS technologies.

Blackboard systems also have issues with communication between the participants. The knowledge base is shared between all participants so the blackboard must be capable of storing all data and assertions from each of them. Whilst this is, in part, an implementation issue, the corollary is that for the information to be useful, other participants must understand it. This means that whenever a new participant joins the system, all other participants must be capable of interpreting the newly available facts. Additionally, the controller must be made aware of the new abilities that have joined the system.

Overall, the blackboard system provides a good model for collaborating systems where a single problem is being solved and where the knowledge sources involved in the system have reasoning that is beyond doubt. They are designed to solve such problems quickly and correctly. The core monitoring system does, however, need to robustly handle the uncertainty, should readily allow the inclusion of new processing techniques as they become available.

Considering the overall requirements along with the relative inflexibility of the blackboard approach, it is believed that the MAS is a more practical approach that can overcome the difficulties encountered with blackboard systems. This method, using completely autonomous agents that are able to communicate with each other, can still deliver the logging through an "archive agent" that takes on the role of storing data for later auditing, similar to the blackboard. It is believed that the MAS approach will be able to meet the requirements outlined at the start of this section.

5.3 Multi-agent systems

In 1995, Wooldridge and Jennings summarised the theories and practice to date in the field of intelligent agents [WJ95]. This paper recognised Multi-Agent Systems (MAS) as an evolution of Distributed Artificial Intelligence and predicted an increasing use of the technology as computing systems evolved.

In this paper, Wooldridge and Jennings made no attempt to define the MAS itself, only the component parts, but the term Multi-Agent System (MAS) is now commonly taken as a system comprising two or more intel-

ligent entities (or agents) [RN03] that communicate to try and meet their own goals and, in doing so, it is typically intended that a system level goal will be met [MDC+07].

Each agent has roles and responsibilities within the MAS. The MAS model is based upon the concept of society; i.e. the designer will align their desired system-level goal with the societal goal of the system and its agents. This might be a competitive situation whereby costs can be minimised or a cooperative situation to improve robustness and redundancy.

Several intelligent agent attributes are outlined by Wooldridge and Jennings [WJ95]. This paper outlines both a *weak notion of agency*, but also some additional traits that together comprise a *stronger notion of agency*. They also define additional attributes that are sometimes discussed in the context of agents. These notions are both described below.

The weak notion of agency describes four key properties required of a software or hardware agent; these are autonomy, social ability, reactivity and pro-activeness.

The autonomy property requires that each agent be able to operate without intervention. The social ability property allows all of the autonomous agents to operate together in order to try to achieve their goal using an agent-communication language. These two properties give rise to a frequent claim about multi-agent systems that they are robust, reasoning that their autonomous nature allows agents to find alternative means of completing a task where the "normal" route is not available. Whilst this can be true, it is not something that is guaranteed by the mere use of agents; rather it must be considered by the system and agent designers to ensure that each agent be able to exercise this freedom.

The remaining pair of properties in the weak notion are reactivity and pro-activeness; two properties which are intrinsically linked. The concept of reactivity requires that all agents should be able to interact with their environment. This means the agents must be able to both sense the environment and affect it. Agent environments can be entirely virtual, entirely physical or anywhere in-between. The final property, pro-activeness, is defined by Wooldridge and Jennings as being "able to exhibit goal-directed behaviour by taking the initiative". For example, an agent notices a perturbation in its environment and takes some action that might prevent an escalation of the effect or a repeat occurrence.

The distillation of the four key properties of the weak notion of agency results in a simplification of this model; the concept of flexible autonomy can be seen in an agent which is able to reason and find means of carrying out a task. In the case of a failure, alternative methods may be found. This flexibility in planning task completion can be derived from the other three properties of social ability, reactivity and pro-activeness.

The attributes associated with a stronger notion of agency define agents as having mentalistic or emotional traits. These traits have not seen considerable use in condition monitoring systems.

One such approach is that of the belief, desire and intention (BDI) model [RG91]. In this formalism, each agent within a MAS has a number of *goals* or *desires* and can be stimulated either by the agent society and/or externalities to have a *belief*. Based on the beliefs and desires of the agent, it can then set itself tasks, or *intentions*, that it will try to effect.

The adoption of a mentalistic model such as this could be useful in modelling the uncertainty associated with assessing the core condition, primarily through the use of the *belief* aspect. There is, however, no *desire* or *intention* component allowing the agents to control the nuclear generation process. As such, the BDI model is an unnecessary complication for this work. Furthermore, other techniques exist to deal with uncertainty. For example, Catterson's work on evidence combination [Cat06] demonstrates one way in which uncertainty and competing conclusions can be rationalised within agent-based systems. Since such techniques are available without recourse to the stronger notions of agency, and because the stronger notions do not currently offer further benefits, the stronger notions of agency are not considered any further within this thesis. The reactor core monitoring system, as a multi-agent system, is conceived using Wooldridge's weak notion of agency.

The aim of the multi-agent system has been defined; by placing a collection of intelligent agents into a suitable environment, an intelligent system can be created which can achieve the collaborative reasoning detailed by Shaw and Fox [SF93]. The characteristics of these individual agents, particularly flexible autonomy, can help create robust distributed systems which are capable of operating in the suboptimal conditions found in real-world systems.

Multi-agent system designers have these general design aims, but no

other restrictions on the construction of agent systems. This could lead to competing implementations that are not compatible. As with many other industries, standardisation could help prevent such fragmentation and allow interoperability of agents. Consideration of the international standards in MAS is undertaken before discussing the system design.

5.3.1 MAS for nuclear operations

A review of the literature revealed two applications of multi-agent systems to nuclear plant. The first of these, from 1996, described the Advanced Plant Analysis and Control System (APACS) for Canadian nuclear reactors [WW96] and the second is a proposed system for "anticipatory control" generation IV plants that are yet to be built [WW96].

The APACS system is a demonstration system operating at the Bruce B nuclear plant by Lake Huron in Canada. This knowledge-based system uses MAS technology to monitor continuous variables from the plant. It will reason about potential causes of any deviations from normal operation. The system uses a repository for storing the data, similar to that proposed for the graphite core monitoring application. A commercial object-oriented database management system is used and this appears to be tightly bound to the individual agents with the MAS and therefore negates one of the key advantages of using MAS, namely extensibility. An ontology has been developed for this application which contains around 350 concepts covering many aspects of the station including boilers and reactors. No recent information can be located on this project, suggesting that it did not see use beyond a demonstration system.

In 2003, Uhrig and Tsoukalas [UT03] proposed that multi-agents be used for anticipatory control of generation IV nuclear plants. Their application, however, is to theoretical generation IV plants and relies on the ability to wrap "existing" computer code as agents that can operate the reactors semi-autonomously. In their work, they claim that "nuclear power plants are by their design well suited for the application of intelligent agents to carry out the safety assurance function as they are well instrumented for purpose [sic] of defining their safety status". This system aims to be able to predict forthcoming reactor conditions and proactively change the operation. In doing this it is planned that this system will be able to perform

the majority of control of a reactor, reducing the personnel required for operations.

Neither application of agents presented here provides robust evidence for the use of agent-based systems as a sound deployment mechanism, but do indicate that this area is worth exploring.

5.3.2 International standards

As communication is a key part of multi-agent systems, standards are essential for both interoperability between agent systems and the possibility of reusing agents within different agent-based societies.

The Foundation for Intelligent Physical Agents (FIPA) was set up in 1996 to assist developers in creating interoperable and reusable agents. FIPA's main aim is to standardise the messaging structures within and between agent-based systems. Anyone can code an agent according to the standards that will be able to operate within any FIPA-compliant multi-agent system conforming to those standards. Having the ability to reuse components should allow rapid development of new applications.

The FIPA standardisation process involved examining the processes associated with implementing intelligent agent-based systems and extracting those parts that are common for standard development. Their approach resulted in a "platform specification"; this specification defines features that any FIPA-compliant platform should provide for individual agents and sets out the high-level approach to inter-agent messaging.

The standards hierarchy is shown in Figure 5.1. This figure also lists all finalised standards in each area. Additional standards that are not finalised and are still at the prototype or experimental stage have been omitted from this diagram.

As can be seen in Figure 5.1, standards across the FIPA architecture have been finalised, from a high-level application to low-level communication standards. In addition to those standards shown in the hierarchy, there are several more proposed and experimental standards under development.

Some of the FIPA standards, such as the "Nomadic Application Support" and "Quality of Service" are not relevant to the field of condition monitoring as they are for specific applications. The Quality of Service (QoS) application standard, for instance, defines an ontology for repre-

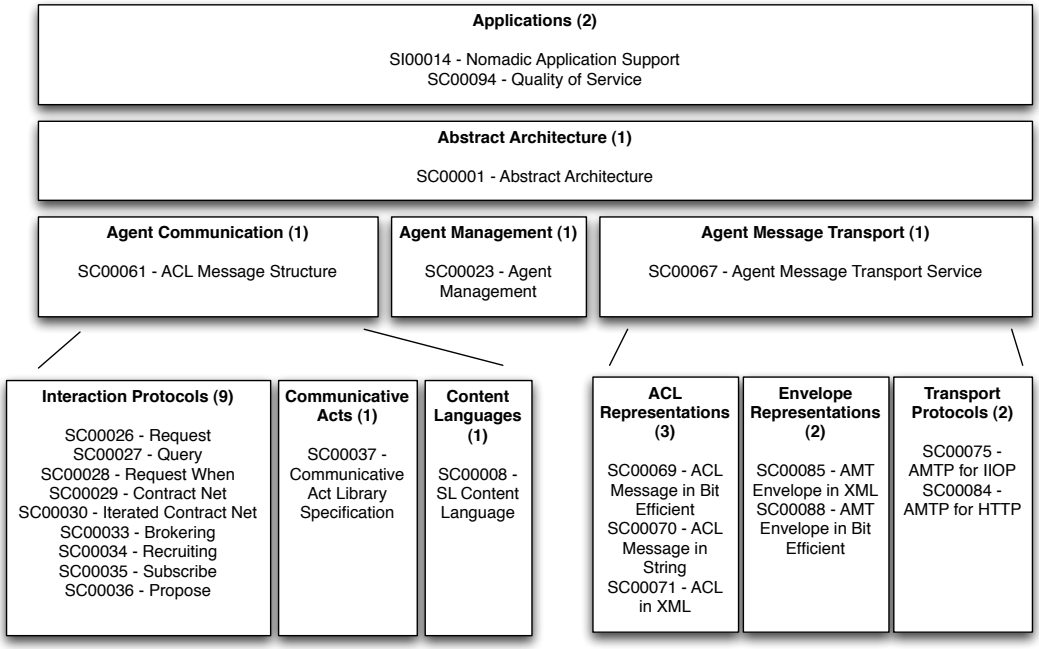

Figure 5.1: The FIPA Standards hierarchy

senting the Quality of Service of the FIPA Message Transport service. Ontologies are explained in more detail in Section 5.3.4.3, but this is merely a component of the Nomadic Application Support application and, as such, it is not relevant in this case. Those specifications which are relevant to the reactor monitoring application are described in Section 5.3.3.

In 2005, FIPA joined the IEEE Computer Society and became a Standards Committee, giving international recognition to this work. Since then, a number of platforms have been developed which comply to the FIPA standards; these platforms provide a "middleware" that is available to agent developers to build their agent-based system.

FIPA standards, although internationally recognised, are not used by all agent-based systems. Neither the Autonomous Learning Agents for Decentralised Data and Information Networks (ALADDIN) project based at Southampton University [RCJ09] or the APACS system for nuclear plant monitoring (see Section 5.3.1) use standardised communication protocols.

Ultimately, the choice of whether to use FIPA or an alternative suite of standards is the choice of the designer, but the international acceptance FIPA has now gained gives confidence that other standards-compliant sys-

tems will be developed, thereby allowing novel and beneficial interactions in the future.

5.3.3 Key FIPA components

This section outlines the key documents in the FIPA hierarchy that are relevant to condition monitoring. As noted in the previous section, the FIPA standards are written as a set of requirements for platform designers rather than a suite of guidelines for designers of individual agent-based applications or agents. A key part of being able to design a suitable agent-based system is understanding the platform upon which the agent will be executed.

Starting at the top of the standards hierarchy shown in Figure 5.1, the first relevant standard is that of SC00001 [Fou02a], the Abstract Architecture standard. This standard is relevant as it defines the basic minimum requirements of any system implemented using FIPA compliant agent-based technologies. It states that the "architecture must include mechanisms for agent registration, agent discovery and inter-agent message transfer". The standard also provides definitions of how FIPA agent platforms must allow the registration and discovery of other agents using a prescribed ontology. The ontology in this case is not chosen by the designer, but rather specified by FIPA for the purpose of allowing agents to access the platform. This specification requires that all agents have a host container, or Agent Platform (AP), within which they are executed. The AP must provide the inter-agent messaging and management functions which described in more detail below. This abstract standard does not require that containers be implemented in a specific programming language and neither does it preclude the addition of further platform-level components that the designer may require.

The remaining standards detailed in FIPA-SC00001 follow on from these basic requirements. There are additional sets of standards for the communication languages and the transportation of messages, described further in Section 5.3.4.

The final key standard is SC00023 [Fou04] covering the Agent Management Reference Model. This is shown in Figure 5.2. It can be seen here that the AP described above must contain an *Agent Management System* (AMS)

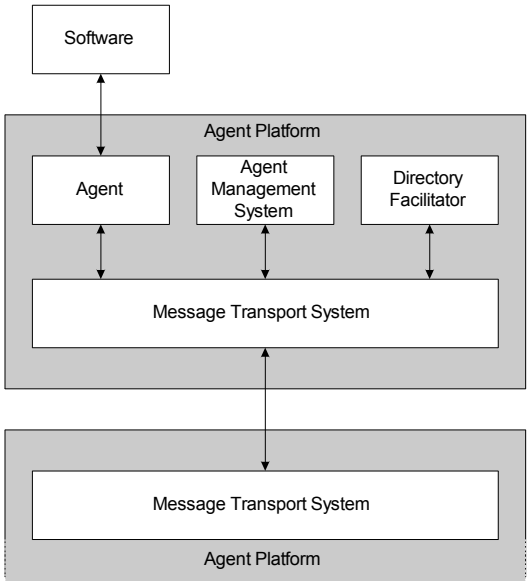

Figure 5.2: The FIPA Agent Management Reference Model

and, optionally, a *Directory Facilitator* (DF). It also shows graphically that each platform should have its own *Message Transport System* (MTS) which enables communication between both agents local to that platform and those located in other platforms. Within agent-based systems, the AMS is analogous to a "white pages" telephone directory where a number can be found for a known entity. The DF is analogous to a "yellow pages" directory where a specific service can be located without knowing who might be able to provide that service. The processes and agent-based interactions to support these procedures are detailed within the various FIPA communication standards.

Since communication is a key part of agent-based systems, these are dealt with in detail in the next section.

5.3.4 Communication between agents

A key part of the multi-agent paradigm is that of communication and proof of this can be found in the proportion of FIPA standards that relate to the communication aspects. Whilst part of the standards suite defines the encapsulation of a message and the mechanics of transferring the message between agents, the application-level concept definition is left to system

designers. Agents which do not share the same protocols and definitions are obviously unable to communicate.

Communication in all FIPA compliant systems is based upon speech theory. This breaks down any interaction between entities into actions. For example, one agent may request that another perform an activity; the other agent can then choose to either agree or refuse. These three actions, `REQUEST`, `ACCEPT` and `REFUSE` are all basic communicative acts and all agent interactions can be built from a small number of communicative acts. There are 22 communicative acts defined by FIPA. Since these are actions which agents can perform, they are also known as *performatives* [Fou02d].

In addition to specifying the communicative acts that can take place between agents, the FIPA standards also specify how domain information should be encoded. Within FIPA systems, this is known as the ontology and describes concepts within the domain and relationships between them that agents are able to use and reason with.

In terms of platform functionality required by the FIPA standards, there are 24 standards covering Semantic Language (SL) grammars, Agent Communication Language (ACL) protocols and higher-level considerations for agent platforms. ACLs are explained in more detail in Section 5.3.4.1.

There are two mid-level standards shown in the standards hierarchy in Figure 5.1, these are standards SC00061 [Fou02b] and SC00067 [Fou02c]. These deal with two different aspects of communication. SC00061, and those standards shown below it within the hierarchy, deal with how particular interactions should take place, such as the negotiation of a contract. This is analogous to what might be written between two parties in a letter- or e-mail-based communication. SC00067, and those standards shown below it, are concerned with the formation of ACL messages into transportable data and the subsequent carriage of that data across protocols such as Internet Inter-Orb Protocol (IIOP) or Hypertext Transfer Protocol (HTTP). Using the personal communication metaphor again, the SC00067 aspects are concerned with how letters or e-mails are addressed and transported from the source to the destination.

A FIPA-compliant message is defined by the Unified Modelling Language (UML) diagram shown in Figure 5.3[Fou02a]. This shows that a message comprises several components.

The components of the FIPA compliant message allow for flexibility in

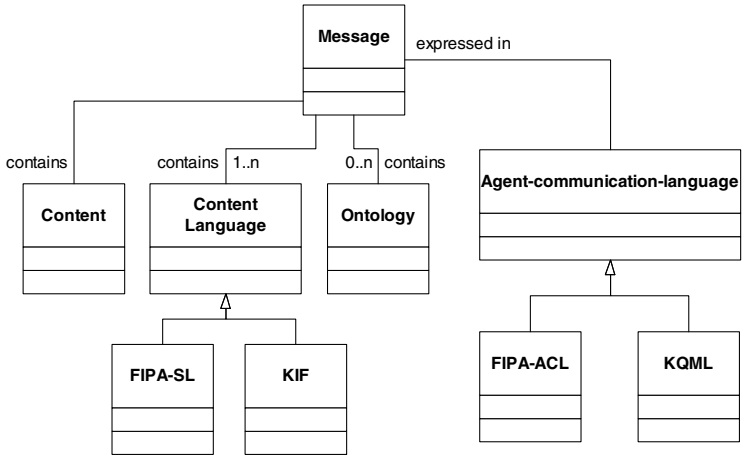

Figure 5.3: The FIPA message structure

the conversations between individual agents. Each component outlined above is considered in turn to show both the contribution to agent-based messaging and to consider appropriate implementations for the graphite core monitoring application. As can be seen in Figure 5.3, all content in the message is expressed in an agent communication language. These are described first.

5.3.4.1 Agent communication language

ACLs are structures in which communicative acts, and any necessary content, can be expressed. The only required information in an ACL message is the performative being used however, depending on the performative, additional information can also be added.

As previously noted, performatives are the basic actions that agents can undertake, and the current standards list 22 of them. The full list of performatives is shown in Table 5.1.

This feature of FIPA is required in any agent-based system, but there are additional standards for bit-efficient, string and eXtensible Markup Language (XML) based transmission of this information [Fou02a]. As the ACL is used for the actual transmission of messages, the choice of technique should be made based upon any constraints on data passing, such as limited bandwidth. In this application, there are no such constraints so the data will be passed using the default string-based ACL structure.

It would be relatively simple to change this in the future if required, similar to changing an Internet connection from wired to wireless; the application will work regardless of the method used to transport messages.

ACLs are a mandatory part of the FIPA standards and are usually provided by the FIPA platform upon which agents are implemented.

5.3.4.2 Content language

Content languages, like ACLs, are arbitrary and simply need to be understood by both participants conveying a message. Content languages are used to express the propositions, actions and terms used by individual agents. The FIPA standards list a number of content languages including the FIPA Semantic Language (SL) and the Knowledge Interchange Format (KIF) [Fou02a].

Although the designer can choose the content language and it can be readily changed, the standards process has only standardised the FIPA SL content language. As such, it can be concluded that FIPA SL is an appropriate language for use in a core monitoring application.

5.3.4.3 Ontology

The FIPA model allows for application-specific communication by requiring system designers to implement the semantics to be shared between agents within an *ontology*. Only those agents sharing the same ontology will be capable of communicating.

The ontology is defined by FIPA, in [Fou02a], as:

> "A set of symbols together with an associated interpretation that may be shared by a community of agents or software. An ontology includes a vocabulary of symbols referring to objects in the subject domain, as well as symbols referring to relationships that may be evident in the domain."

Within FIPA compliant systems, there are three basic ontology items, the `Concept`, the `AgentAction` and the `Predicate`. All domain-specific items should extend one of these three types. Each `Concept` is an item which represents physical items, such as a reactor or a power station, or data, such as an interpretation. An `AgentAction` is something which one

Act	*Summary*
Accept Proposal	The action of accepting a previously submitted proposal to perform an action
Agree	The action of agreeing to perform some action, possibly in the future
Cancel	The action of one agent informing another agent that the first agent no longer has the intention that the second agent perform some action
Call for Proposal	The action of calling for proposals to perform a given action
Confirm	The sender informs the receiver that a given proposition is true, where the receiver is known to be uncertain about the proposition
Disconfirm	The sender informs the receiver that a given proposition is false, where the receiver is known to believe, or believe it likely that, the proposition is true
Failure	The action of telling another agent that an action was attempted but the attempt failed
Inform	The sender informs the receiver that a given proposition is true
Inform If	A macro action for the agent of the action to inform the recipient whether or not a proposition is true
Inform Ref	A macro action for sender to inform the receiver the object which corresponds to a descriptor, for example, a name
Not Understood	The sender of the act (for example, i) informs the receiver (for example, j) that it perceived that j performed some action, but that i did not understand what j just did. A particular common case is that i tells j that i did not understand the message that j has just sent to i
Propagate	The sender intends that the receiver treat the embedded message as sent directly to the receiver, and wants the receiver to identify the agents denoted by the given descriptor and send the received `propagate` message to them
Propose	The action of submitting a proposal to perform a certain action, given certain preconditions
Proxy	The sender wants the receiver to select target agents denoted by a given description and to send an embedded message to them
Query If	The action of asking another agent whether or not a given proposition is true
Query Ref	The action of asking another agent for the object referred to by a referential expression
Refuse	The action of refusing to perform a given action, and explaining the reason for the refusal
Reject Proposal	The action of rejecting a proposal to perform some action during a negotiation
Request	The sender requests the receiver to perform some action. One important class of uses of the request act is to request the receiver to perform another communicative act
Request When	The sender wants the receiver to perform some action when some given proposition becomes true
Request Whenever	The sender wants the receiver to perform some action as soon as some proposition becomes true and thereafter each time the proposition becomes true again
Subscribe	The act of requesting a persistent intention to notify the sender of the value of a reference, and to notify again whenever the object identified by the reference changes

Table 5.1: The 22 FIPA Communicative Acts, or Performatives

agent can ask another agent to do and each `Predicate` is a function which will evaluate to either `true` or `false`. Each of these are arranged hierarchically and can utilise multiple inheritance. These are related to the performatives described in 5.3.4.1; some performatives, such as `REQUEST`, utilise an `AgentAction` whilst others, such as `QUERY-REF`, utilise `Predicates` to identify `Concepts`.

There are two common ways to express an ontology, as Frames and using the Web Ontology Language (OWL). Both of these model concepts in the ontology and allow each to contain properties. Within Frames, the properties are known as slots. As an example, a `Reactor` entity could have a station property, or slot, containing a `Station` entity along with an `id` slot containing the reactor identifier. These objects could then be used within `Channel` objects to indicate which reactor and station a channel is within.

A single ontology representing all concepts from all domains is simply not practical. This is because a predicate, such as `IsCloseTo` can have different meanings in different domains. For example, in the graphite core case adjacent channels may be considered to be close. Equally, if the same predicate is applied to a geographical problem, London may be considered far away from Glasgow in a UK-based application but close when considered internationally. Predicates can have properties or slots in exactly the same way as Concepts and these are generally used to pass parameters; in this example searching for places close to other places would require specifying one of the locations. As noted above, the application of meaning to the predicates is something that is necessarily domain specific.

The requirement to be able to express concepts, actions and predicates to an appropriate degree within a domain means that the designer of any agent based system must either find an existing ontology which expresses all domain-specific definitions they wish to use, or design an ontology specifically for the application. An appropriate ontology will therefore be required for the core monitoring application.

5.3.4.4 Content

The final part of the FIPA-compliant message is the Content. This is the main payload of the message. Having chosen the content language and ontology, the creation of the message content can be automated.

The JADE platform enables this by providing the functionality to create content programmatically. Instances of these items in the ontology can be created and each of the slots can be set and modified before populating an ACL message and sending it to the recipient.

5.3.5 MAS platforms

With the key properties of multi-agent systems defined, and having decided upon using a FIPA compliant platform, consideration turns to the concrete implementation to be used.

Two agent platform reviews, both from 2004, considered the optimum agent platform. There are ten FIPA-compliant Agent Platforms listed on the FIPA website [fip10] and these were all considered by Lesczcyna [Les04].

The findings of this report were that, of the ten platforms, only three were still maintained. These were the Tryllian Agent Development Kit (ADK) [try10], JADE [jad10], and JACK [jac10]. At the time, the Tryllian and JACK options were both commercial-only products whilst JADE was the released under an open-source licence. Lesczcyna concluded that if the need was a well maintained, supported and free platform, JADE was the only option. The other non-maintained and paid-for options were discounted. Lesczcyna offered no additional benefits to using the commercial software though the desire for support and service level assurances would surely be the primary reason.

In Shakshuki [SJ04], JADE, JACK and a third platform, the Zeus Agent Building Toolkit, were considered. The latter of these was identified in Lesczcyna as having not been updated for some time and, as of 2010, the project website appears to have completely closed down. Shakshuki ignored licensing costs and focused on run-time performance, specifically comparing the message passing times between individual agents as this is seen as a key performance metric for agent platforms. The times were considered for different population sizes of agents and compared the case where agents were in the same and different agent platforms. The conclusion from this work was that where agents reside in the same AP, JADE was superior and where agents are located in different containers, JACK was superior. In both cases, the Zeus Agent Building Toolkit was the worst at handling messages.

Since 2004, the landscape for agent platforms has not changed significantly. Those platforms considered abandoned by Lesczcyna have not been updated or have disappeared completely. No new implementations have been found. As of late 2005, the Tryllian ADK is now dual-licensed and is available under an open-source as well as a commercial license.

These platforms both claim FIPA-compliance and, as such, there is now little to choose between them. Having considered this, JACK has been discounted on the basis of cost, preferring an open-platform option which will remain available in the long-term. Since there remains little difference between Tryllian and JADE, JADE is selected due to the significant community of users and that it has been used to successfully demonstrate engineering solutions in areas such as digital control system simulations for nuclear plants [SMC+10], naval power systems [CS07], fault diagnosis within incineration plants [WLJL10] and traffic management [CC10].

5.3.6 MAS for condition monitoring

Intelligent multi-agent systems have been in development for condition monitoring purposes for some time and have developed from rudimentary message-passing entities seen in early examples such as Mangina's COndition Monitoring Multi-Agent System (COMMAS) for Gas-Insulated Substations (GIS) [MMM01] to the later development of the same system by Catterson for transformer monitoring [Cat06], allowing multiple analysis techniques to act upon the same data to improve overall system reasoning. Both the COMMAS systems and the Protection Engineering Diagnostic Agents (PEDA) system [CDM05], for monitoring protection equipment, are designed to monitor specific plant-items within electrical networks. Additionally, they are concerned with GIS, transformers and protection devices, all serviceable, replaceable or tunable if there are found to be particular issues with operation.

These systems were designed to analyse data "of the moment" and make the results of this analysis available to the engineer. Both early COMMAS GIS and transformer systems operated in this way without any persistent storage between operations. The PEDA system performed its analysis in this way, but the conclusions drawn were output to a relational database for later retrieval by engineers [CDM05][MD06].

In the COMMAS and PEDA systems, it is important to note that the present state of the monitored item can vary due to many factors. This means that each analysis is dependent on ambient conditions. The results of the analyses should be archived for later use to ensure that even short time-scale faults are captured. This situation is complicated further when the maintenance regime is considered. The "step-changes" in condition and hence data that can be expected as a result of maintenance means that all data from these sources could have an implicit "shelf-life" beyond which it may no longer be applicable. These aspects have been considered by the system designers in their analyses.

The analysis approach used in the early COMMAS system and PEDA is reasonable for systems which, when defective, are likely to repeatedly generate monitoring data for the same defect. Alternatively where data is sparse, as in the graphite core application, information must be retained for future use.

5.3.7 MAS for Structural Health Monitoring

The structural deterioration process is different to both of the PEDA and COMMAS cases in the previous section. If a crack appears within the structure after 2 years, that crack will remain in the structure for the rest of its life; there is no condition under which that crack can heal. It is possible that due to material changes and internal stresses crack dimensions change and open wider or become closed again. Nevertheless, it would remain the case that there is a structural discontinuity at that point which could contribute to structural failure.

The nature of continual degradation results in different data currency; data gathered remains valid forever. This has a cascade effect on the design of systems to analyse the data as the information base upon which these analyses are based also grows continuously.

As a result of the number of events involved and the desire to take a structure-wide view of the data, storing all of the data within each agent is impractical. Instead, an alternative data storage design which did not restart analysis every time an agent was executed was sought.

The PEDA approach, using a database for data storage, is an obvious option. Where there is substantial data this approach allows the data to

be managed by existing data management processes within organisations. It also allows fast retrieval and analysis using a plethora of data analysis tools. This could allow future data-mining and more advanced techniques to be developed before deployment within the agent-based system. A significant drawback to this technique is that there is no extensible way of storing arbitrary agent-based data within a database. This was seen in the PEDA application where the output was stored using hand-coded SQL queries in one interface and retrieved using hand-coded queries within a separate interface. Consequently new analyses have no way of querying the previous knowledge base for information; this is not desirable for the core monitoring application.

Guidance was taken from the blackboard systems detailed in Section 5.2; the factual information should be stored in a single, central repository and all agents should be able to retrieve any data they require. Appropriate design of these agents will allow flexibility at all levels as clustered databases can allow several agents to provide the data storage services. Interpreted information can then be generated by each agent as required.

The data storage aspect of the multi-agent systems is found to be lacking in the existing literature. Storage of information within each agent normally takes place only in memory and scanning these facts for those which match content-language queries is not currently easy to perform. This is a significant challenge, one to which this thesis offers a resolution.

Chapter 6 explains this problem in more detail and outlines the current approaches to resolving the issue. A novel method for integrating large-scale storage within agents which is capable of inter-operating with any FIPA SL query is then presented. The remaining aspects of the design of the multi-agent system design are considered within the IMAPS case study in Section 7.2.

5.4 Conclusion

This chapter has outlined the basic concepts of the multi-agent system and shown their suitability in condition monitoring applications.

There has been no previous implementation of a multi-agent system to help address the structural deterioration in nuclear reactors, but this novel

technique helps in providing a system which can be rolled out without requiring revalidation of those parts of the system which have already been implemented, installed and tested. This will allow the integration of further data and structural evidence in the future, without requiring that the whole system be re-engineered.

The development of a structural health monitoring system based on this technology would prove useful and extensible provided it was possible to store the data. This would require the correct ontology for the system, but also the development of a robust data store that could be extended if and when additional data types are created.

Chapter 6 goes on to address the challenge of implementing flexible storage of data in agent-based systems, which supports the aims of the wider agent community.

Chapter 6

Flexible data storage for intelligent agents

6.1 Introduction

Chapter 5 introduced the need to be able to undertake long-term storage of information within agent-based systems as part of condition monitoring applications. This chapter describes the problems with the current data storage mechanisms used within agent-based systems by investigating the underlying location of data, the structure that data should actually be stored in and the interaction between long-term storage and the agent-based programming model, with particular note to system and ontology evolution.

The chapter proposes three different approaches to storing arbitrary data within a data repository. The repository is designed for multi-agent system (MAS) architectures and this chapter explains the advantages and disadvantages of each.

In order to evaluate the approaches, an understanding of the data storage technology is required. This is addressed in the following section explaining the operation of database systems.

6.2 Database systems

There are several available approaches that can be applied to the storage of data within any computer system. These are retaining information in com-

puter memory, or storage within files and databases. The most appropriate technology depends upon the application. Data which has short currency and will be required quickly is best held in memory whilst larger and persistent datasets, of the type under discussion within this chapter, are best stored within a database. Storage of data within files is becoming an anachronism; the advent of multi-user systems, concerns over incompatible file formats and the isolation of useful data have all served to ensure that useful data is increasingly stored in databases. As such, file-based storage is not considered further [CB05a].

Unfortunately, the concepts and requirements of MAS, and the data expressed by an ontology, vary to that of the relational database. In order to successfully store data in a way that it can be readily reused, a consideration of the capabilities and operations of standard relational database systems is required. Additionally, the standard design approaches used within database theory must be considered as this work is fundamental to the construction of fast and efficient databases for use with agents.

6.2.1 Relational database model

The relational database model was introduced by Codd in 1970 [Cod70]. The aim of the relational data model is to store data within databases using the "natural structure" of that data only and "without superimposing any additional structure for machine representation purposes". This model is now standard in database systems; database users are comfortable with modelling data within tables and creating relationships between them. Following the modelling stage, row data can be created, read, updated and deleted; these are the four basic operations of any persistence approach and are often referred to as CRUD operations. Codd also recommends that, in order to allow data management using these operations and to improve data integrity, each row in a relational database table should have a key which can be used to uniquely identify that row.

Each table in a relational database has several columns which can store data values or a reference to a key in another table. The structure of the table therefore defines the relationship between each of the columns; for this reason a table is also known as a relation.

In designing a relational database, it is feasible to store all data in an

Student No.	*Name*	*Address*	*Degree*	*Contact*
199612345	Alice Smith	22 Argyle Avenue	BA English	1234, 8765
199712345	Bob Jones	99 Bishop Street	BSc (Hons) Agriculture	4000
200212345	Alice Smith	22 Argyle Avenue	MBA	1234, 8765

Table 6.1: Example database table in the unnormalised form (UNF)

Student No.	*Name*	*Address*	*Degree*	*Contact*
199612345	Alice Smith	22 Argyle Avenue	BA English	1234
199612345	Alice Smith	22 Argyle Avenue	BA English	8765
199712345	Bob Jones	99 Bishop Street	BSc (Hons) Agriculture	4000
200212345	Alice Smith	22 Argyle Avenue	MBA	1234
200212345	Alice Smith	22 Argyle Avenue	MBA	8765

Table 6.2: Example database table in the first normal form (1NF)

Unnormalised form (UNF); in this case the table can have repeating groups of data between rows. This situation means that if there needs to be an update to any values, several rows in the table must be modified otherwise the state of the database would be inconsistent. As an example, consider the table shown in Table 6.1 showing student registration information; if Alice Smith updates his contact number, two rows in this table must be updated or the data is inconsistent with two different entries contradicting each other.

Within database systems, the solution to this inconsistency is not to update multiple rows as this is inefficient. Instead the database is normalised. There are several *normal forms* in which data can be expressed. The process of normalising the data starts with the first normal form (1NF), where data and relations must meet a minimum set of rules. As an example, 1NF would require that the two contact values for Alice Smith be expressed in another way as there should only be a single value from the domain in each table cell. To make this a 1NF table, the remaining data in the row would be duplicated with one telephone number in each row as shown in Table 6.2.

Unless the two numbers have special significance, such as being a main and secretarial number, the normalisation would usually involve remov-

Name	*Address*
Alice Smith	22 Argyle Avenue
Bob Jones	99 Bishop Street

Table 6.3: Example 2NF database table – Student names

Name	*Contact*
Alice Smith	1234
Alice Smith	8765
Bob Jones	4000

Table 6.4: Example 2NF database table – Contact numbers

Name	*Student No.*	*Degree*
Alice Smith	199612345	BA English
Bob Jones	199712345	BSc (Hons) Agriculture
Alice Smith	200212345	MBA

Table 6.5: Example 2NF database table – Degrees

ing this column from this table and placing them into a separate "Phone Numbers" table as two rows. Each would then have a unique identifier, the student number to refer back to this table, and the phone number.

Moving to the next level, additional rules are added. Adhering to the second normal form (2NF) meets all the requirements of 1NF, plus the additional 2NF rule [CB05b] – that any non-key attributes within a table are dependent on the whole of the candidate key and not just a part of it.

The original table is not 2NF because the only candidate key that can uniquely identify relations is *{Student No., Name, Contact}*. Since the *Degree* attribute is then dependent on only the *Student No.*, the table is not 2NF.

The aim of 2NF is to reduce redundancy by splitting data into separate tables and would, in this case, involve three tables for the person, their contact numbers and their degree information. These are shown in Tables 6.3, 6.4 and 6.5.

There are many normal forms and all of the requirements need not be reiterated here; the point is that a generic archiving agent should be able to store data in as normalised a fashion as possible. This will allow the benefits of database technology to be applied to the archive agent. Storing in a completely normal form may not always possible though as relationships can be complex. For example, all items in a JADE ontology extend the `Concept` type. If a slot is defined as containing a `Concept` rather

than a concrete subclass, there is no way for the database to model which table that slot value can be found in. The differences between ontologies and database models are further discussed in [RTV06].

The desire to store data in a form normally used for databases is important, as it allows database features beyond the basic data storage role to be utilised. A key consideration, especially for large databases, is data access times. Database queries are generally written in a standard Structured Query Language (SQL). SQL queries are decomposed by the database engine and optimised to yield the quickest search of the database. Delays can be introduced when the database plans the appropriate method of executing the query and during data retrieval whilst executing that plan. The slowest means of querying, for example, would be to examine every row in a database to check whether it meets the constraints within the query. This would be time-consuming, so databases use indexes to allow them to quickly look up data. The planner is aware of each index available and chooses the best one, if possible, to use. Whilst indexes can increase the speed of querying data, they must be updated whenever data is created, updated or deleted. This means that any indexing undertaken becomes a trade-off; increasing the speed of querying slows down the speed of changes. The trade-off that indexing enforces is something which is normally tuned by experienced database administrators (DBAs) to the specific requirements of the database and the underlying hardware that the database is running upon. The exception to automatic indexing is primary keys. Where a value in one table has a relationship with one in another table, such as the name in the 2NF contact table (Table 6.4) being linked to the name in the student names table (Table 6.3), a foreign key constraint can be set up to ensure that contact details are only retained for people listed in the student names table. Since primary keys are so frequently used during lookups, and because they must always be unique and therefore describe a unique record, these are always indexed.

Automation of the indexing aspect is not considered in this thesis, but an awareness that they are available to accelerate information retrieval is required during the following discussion.

The goals of database normalisation are, however, worth considering in the context of fact storage for intelligent agents. The various theories of database normalisation serve to produce a coherent and readily maintain-

able information store; indeed, the database schema for a fully normalised database will bear a resemblance to that of an ontology. In many cases, the relations between the tables may map directly to predicates; to reuse the preceding example, placing telephone numbers into a separate table and relating this to a person means that the relation could be expressed within the domain as "phoneNumber" and "belongsTo".

The various approaches to storage and the compatibility of ontology schemas with database schemas will be therefore be considered within the next section.

6.3 Storage approaches in agents

A commonly used approach within agent-based systems for condition monitoring is the "wrapping" of existing systems and databases and exposing properties of those systems to the agent society [Jen01]. This involves encapsulating a pre-existing system within an agent-based wrapper which makes that whole system appear as a single agent. Inevitably, this process requires that data be extracted from the encapsulated system and represented within the ontology. In the case of databases, information from the database must be able to be assessed against predicates and the mappings between predicates and database queries tend to be hand-coded. Beyond these wrapped sources of knowledge, most documented agents operate on a small number of facts held in volatile computer memory.

The approach of using existing data sources serves particularly well where there is a pre-existing database; it makes little sense in replicating data in another format as it is both a waste of valuable resources and may lead to inconsistencies between multiple repositories. The issue with wrapping these other databases arises out of the implementation of predicates. If there are a large number of predicates, the degree of translation required between the representation in the wrapped system and that in the external system grows rapidly. This effect is considered by Giampapa et al in [GPS00] where it is argued that wrapping is only useful in certain circumstances where the cost of performing all of the translations required for interactions is less than the cost of simply making the systems interact fully.

The graphite core monitoring application has no existing database so there is no need to wrap an existing database. This gives the opportunity to design the optimum data-storage technique for agent-based systems and use this from the outset.

One option for storing the information is to retain it within the working memory of each agent, or shared between several agents. Several agents each holding a part of the overall condition could thus provide a distributed memory. For applications which aim to analyse lots of information over a long period this leads to the problem of how to ensure that the information remains available to all agents in the agent society over a prolonged period. This can be likened to the collective memory of peer-to-peer file distribution systems where there is no guarantee that all the parts of a file will be available at any given time due to systems or networking issues. The loss of a small part of a file can then render the remainder useless. Within an agent-based system using solely distributed memory for storage, the loss of any one agent could result in the unavailability, even temporarily, of critical information. For any application which relates to safety and security, this is untenable.

The PEDA application described in Section 5.3.7 relies upon two relational databases, but neither is an integral part of the agents knowledge base, and neither makes the information readily accessible by all agents [CDM05]. In PEDA, one database is read-only and is the source of SCADA data for analysis and the other is a database for storing the outcomes of the analysis. Code for retrieving and storing this information was hand-coded for the task. Whilst this technique is capable of delivering a highly-efficient database which can be readily optimised for the queries being undertaken by the various agents, it is concluded that due to the lack of reusability of this information, it is incompatible with the flexibility often claimed of agent-based systems.

In order to deliver the flexibility promised of agent-based systems, any multi-agent data store must be able to store any fact that an agent can itself use as part of the reasoning for a problem. The reasoning, however, within a multi-agent system will be dependent upon the current ontology being used. It is therefore possible that data stored within a data store using one version of an ontology will not be compatible with future versions of that ontology; for example, the addition of a single mandatory field would

have this effect, rendering all previous facts in the old format unusable with the new format.

This discussion leads to a number of key conclusions about the agent-based usage of relational databases. Firstly, there exists no simple way to store any fact used within the agent-based system in a manner that can be readily queried by standard agent-query languages. Secondly, the queries that are generated by a series of predicates in a conversation could result in complex database queries that may be difficult for the Database Management System (DBMS) to parse and execute, resulting in slow query times, because optimising agent-based queries can be very difficult due to the unpredictability of the conversations and resulting queries which will emerge as the system is used. This problem of slow queries could cause extensive table locking and resource exhaustion on database servers; in instances where an agent is used to wrap an existing corporate resource, which by its existence as a corporate resource is probably being used for other duties, this may cause extensive disruption to other parts of the business.

This discussion leads to five core requirements for an intelligent agent which can store arbitrary agent-based data.

- It must be capable of storing any fact or frame that the agent itself is capable of interpreting,
- It should be able to be queried readily by the interpretation of FIPA compliant query languages,
- It must support object hierarchies as these are commonly used within ontologies,
- It should have an appreciation of different, and different versions of, ontologies and either be able to handle different ontology versions or to flag where stored data is no longer compatible with an ontology, and,
- The format of the underlying data store should readily support optimisation of queries.

Additionally, although not direct requirements, there are two further items for consideration in implementing agent-based data access:

- The database form adopted should be readily recognisable to Database Administrators as this will facilitate further use of the datasets collected via the agent-based system, and
- Agent-based system designers should take care in their use of pre-existing databases to ensure they are not causing problems for other users; this is considered in relational databases as logical data independence.

As there is no perfect solution to the problem of storing data for agent-based systems in a way which is readily searchable and compatible with agent-based systems, this area requires further investigation.

6.4 An archive agent

Section 5.3.2 introduced the Foundation for Intelligent Physical Agents (FIPA) standards. It was noted that one of the key aims of this organisation, and one of the claims of agent-based technologies, is that agents can be deployed to undertake roles thereby adding functionality.

Reusable agents that can be placed within any domain, other than those which are part of the platforms and deal with the FIPA ontologies, have not been encountered during literature reviews. The requirement to be able to handle the different predicates within an ontology have always resulted in agents being tied to a specific ontology and, hence, a specific application. Where a database is used it is possible to circumvent this restriction by encoding the meaning of the predicates in terms of SQL queries. Queries can then be combined using set-handling operations within the database to locate appropriate matching facts.

The main design of the archive agent requires only as many interactions as required to implement the CRUD operations. Other agents must be able to query for an arbitrary set of facts including finding out whether a specific fact has already been stored, and be able to request the storage, update or deletion of a fact. This requires the implementation of several of the performatives listed in Table 5.1 in Section 5.3.4.1. The performatives required for an archive agent are listed in Table 6.6.

Of the communicative acts listed in Table 6.6, all will be required with the exception of `SUBSCRIBE`. `SUBSCRIBE` is considered optional for this agent

Act	*Use in archive agent*
QUERY-IF	Used to find out whether a fact has been stored; replies with CONFIRM or DISCONFIRM
REQUEST	Used to request a modification of the knowledge base – an AgentAction from the ontology should be used to specify whether this is creation, update or deletion of a stored record
QUERY-REF	Used to make a query to retrieve information from the archive, will reply with INFORM
SUBSCRIBE	Could be used to request that relevant changes are notified
INFORM	Used to respond to QUERY-REF messages
CONFIRM	Used to respond to REQUESTs
DISCONFIRM	Used to respond to REQUESTs

Table 6.6: Communicate acts for the archive agent

because it is not good for frequent use as, in the event of an agent terminating abnormally and losing the subscription information, the subscriber has no way of knowing that it is no longer receiving updates. As a result of this, SUBSCRIBE has not been considered in prototyping this agent.

The standards provide the high-level protocols for each of the QUERY and REQUEST interactions. These are defined in [Fou02e] and [Fou02f] respectively and are shown in Figures 6.1 and 6.2. These standards dictate that the interactions with the archive agent will, at the performative level, be identical for every application.

As well as the performatives being identical regardless of application, content languages will also be standard. As noted in Section 5.3.4.2, the FIPA Semantic Language (SL) is the only standardised content language. This is therefore used for the prototyping and testing; additional content languages could be added to this agent later as a further development.

The communication part of the archive agent, being fixed, makes for a good delineating point in the design of the agent; this part of the archive agent can be written to be completely application independent.

These design features mean that a generic architecture for an archive agent can be constructed. An appropriate high-level architecture is shown in Figure 6.3. This figure shows that there are three levels within the archive agent design; one which allows the interactions shown in the agent-interaction diagrams, one which can translate messages from the ontology used to construct the message into a format usable by the data store, and the data storage part itself.

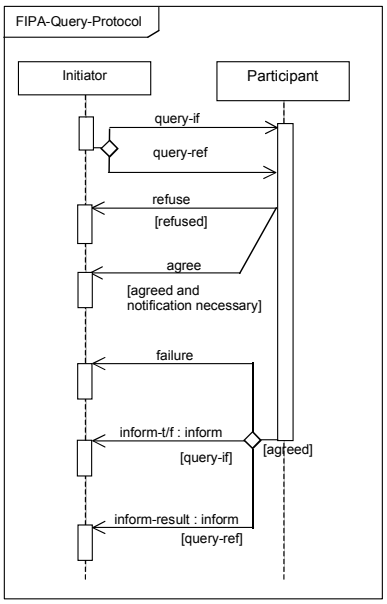

Figure 6.1: FIPA `Query` Interaction Protocol

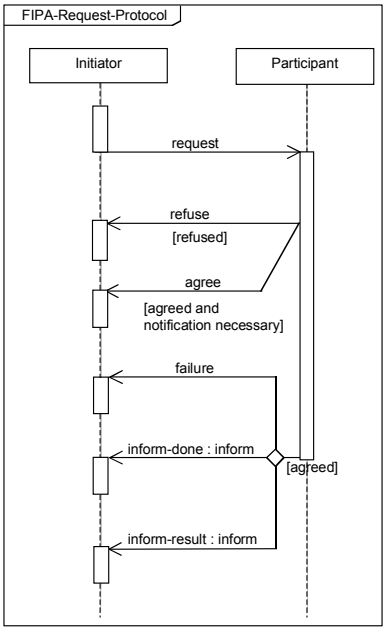

Figure 6.2: FIPA `Request` Interaction Protocol

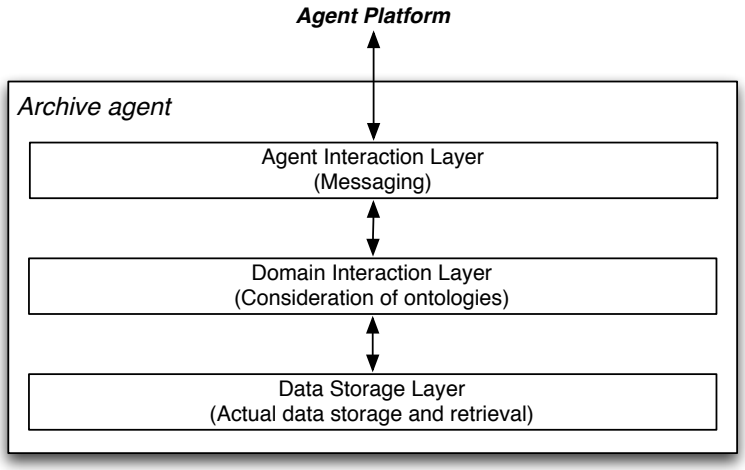

Figure 6.3: High-level archive agent architecture

The "domain interaction" and "data storage" layers are now considered separately.

6.4.1 Content translation

A key challenge in providing an agent which is capable of operating within any domain is being able to deal with the ontology for that domain. Recalling the example from Section 5.3.4.3, a predicate, such as `IsCloseTo`, can have different meaning in different domains. It was shown that when applied to a geographical problem, London may be considered far away from Glasgow in a UK-based application but close when considered internationally. The application of meaning to predicates is something which is necessarily domain specific.

Allowing an agent to assess these predicates in order that correct response can be generated is therefore a key requirement. It is also a feature dependent on the ontology and the application. As such, this can only be handled by configuration of the archive agent relating the predicate in the ontology to the information within the stored facts.

The approach taken is one that does not depend on modelling complete requests that may come from other agents, rather it is configuration on a predicate by predicate basis for each supported ontology. Thus, the agent is configured by specifying each predicate in the ontology that the agent can use to reason and how to use that within the storage engine. This

allows the queries to be generated, meeting the read component of CRUD. Additionally, actions relating to the three data modification operations – create, update and delete – are required.

The syntax used for archive agent configuration is shown in Figure 6.4; the format of this file is a series of key-value pairs. The keys are made up of the agent name, `archive`, an ontology identifier to configure each ontology supported by the agent (in this case `beontology`), and then settings for the implementing class (`ontologyClass`), a slot within the frame which can be used to communicate the database primary key (`identifierSlot`) and each of the predicates. This allows one of the normal form requirements to be met, allowing every fact stored within the database to be uniquely addressed.

This example also shows the `OccurredBefore` and `InReactorRegion` predicates and, for each of these, an SQL fragment is required along with mappings from the slots within the predicate to the SQL. When a query is encountered which uses these predicates, the archive translates the predicate into SQL. If the query is for something that `OcurredBefore` a certain date and is `InReactorRegion` of a specific channel, the agent requests the intersection of the sets and can then return those items matching.

As an example of how the predicates are processed, the directives state that where an `OccuredBefore` predicate is encountered, matching facts can be identified by the SQL fragment, but that the `#startTime` variable in the SQL should be replaced with the value of the `OCCURREDBEFORE_TIME` slot within the predicate object.

One drawback of this method of translation between the agent messaging layer and the data storage layer is that it depends on the features available in the data storage layer. This means that the designer must choose how the data will be stored in order to codify the translation. The work undertaken on this agent shows that even the decision of how to store the data is complicated by trade-offs.

6.4.2 Data storage

As detailed in the previous sections, it is feasible to code parts of an archive agent which are relevant to all applications though some features will require configuration for the ontology used. The ultimate aim of the archive

```
archive.beontology.ontologyClass=
        uk.ac.strath.eee.be.ontology.BeOntology
archive.beontology.identifierSlot=internalIdentifier

archive.beontology.predicate.OccurredBefore.sql=SELECT
        concept_id FROM slot_values WHERE slot_name = 'startTime'
        AND slot_abs_value < '#startTime'
archive.beontology.predicate.OccurredBefore.startTime=
        OCCURREDBEFORE_TIME

archive.beontology.predicate.InReactorRegion.sql=SELECT
        concept_id FROM slot_values WHERE slot_name=
        'reactorComponent' and slot_concept_value IN (SELECT
        concept_id FROM slot_values WHERE slot_name='position' AND
        slot_concept_value IN (SELECT concept_id FROM slot_values
        WHERE slot_name='positionX' AND slot_abs_value < (#xLoc +
        #pitches) and slot_abs_value > (#xLoc - #pitches)) AND
        slot_concept_value IN (SELECT concept_id FROM slot_values
        WHERE slot_name='positionY' AND slot_abs_value < (#yLoc +
        #pitches) and slot_abs_value > (#yLoc - #pitches)) AND
        slot_concept_valueIN (SELECT concept_id FROM slot_values
        WHERE slot_name='positionZ' and slot_abs_value < (#zLoc +
        #pitches) and slot_abs_value > (#zLoc - #pitches)))
archive.beontology.predicate.InReactorRegion.xLoc=
        HASPOSITION_POSITION.POSITION_POSITIONX
archive.beontology.predicate.InReactorRegion.yLoc=
        HASPOSITION_POSITION.POSITION_POSITIONY
archive.beontology.predicate.InReactorRegion.zLoc=
        HASPOSITION_POSITION.POSITION_POSITIONZ
archive.beontology.predicate.InReactorRegion.pitches=
        INREACTORREGION_NUMBERPITCHES

...

archive.beontology.replacePredicateName=ReplaceIdentifier
archive.beontology.replacePredicateNewValue=variable
archive.beontology.replacePredicateIdentifier=internalIdentifier
```

Lines beginning with a tab continue the previous line

Figure 6.4: Configuration file extract

agent, however, is to store the information within a database. There are several options for realising this part of the agent, and, as will be seen, there are advantages and disadvantages to each technique.

The first consideration in choosing a storage technology was that of compatibility between data types in the ontology and the database. An ontology, is usually represented as a hierarchy of concepts much like an object hierarchy in object-oriented (OO) computing languages. OO programming is now routinely taught and is a well-understood technique; there are commercial databases available that can directly store objects and their relationships. Such systems are known as Object Oriented Database Management Systems (OODBMS) or Object Relational Database Management Systems (ORDBMS). The principal issue with these is that most OODBMS and ORDBMS systems depend upon single inheritance where each concept can extend only a single, more abstract, concept in a tree. Ontologies, however, support multiple inheritance where each concept can have multiple superconcepts. The use of multiple inheritance is an issue because all of the candidate FIPA-compliant platforms use the Java programming language, which does not support multiple inheritance natively. Additionally, a review of database systems reveals that support for multiple inheritance is limited; only one database engine, by the open-source PostgreSQL project, supports multiple inheritance. This support is being developed to support entity models, but there remain practical issues with regard to indexing these tables [Pos10] and, as such, there is no readily available database facilitating direct OO storage within an ORDBMS or OODBMS. Despite this, ORDBMS or OODBMS storage would be suitable for the majority of applications encountered within the condition monitoring domain which use only single inheritance ontologies. As database engines develop, this may expand to providing full multiple inheritance support.

This section considers three techniques for storing information within agent-based systems. The first is a simple schema based upon a standard Relational Database Management System (RDBMS), the second is based upon a commercial object-oriented database and the third is a set of standards designed to store objects within a standard RDBMS.

All of these techniques utilise an additional surrogate key which is added to the data stored in each table. This is added to allow updating and removal of facts by reference rather than searching by all fields and is

a standard technique used within databases storing objects [WdJ92].

6.4.2.1 Simple data storage

A fact that an agent can reason with has two parts to it; the structure defined in the ontology and the information which populates the fields in that ontology. The ontology may not require that all fields be populated however, so the data stored need not represent every single slot within the fact.

The first approach to storing multi-agent data within a database was therefore to treat both the fact schema and information as data to be stored within the database. This called for a database structure which would allow the storage of both the information and structure within the ontology. The devised database structure stores facts and each of the values associated with that fact in a simple tree-like structure; each fact is related back to the ontology it exists within using the `Ontology` table.

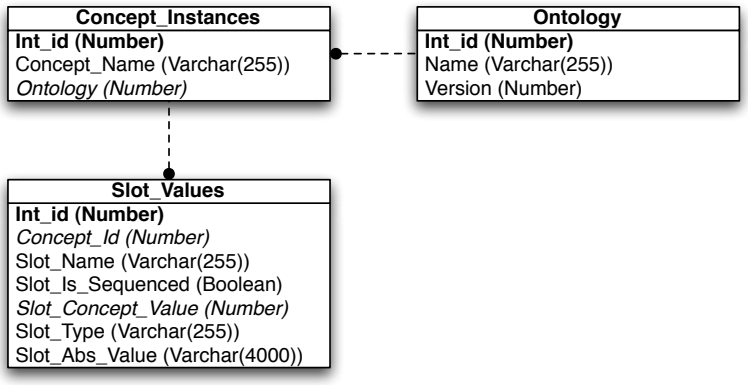

Figure 6.5: Simple Archive Agent Schema

The schema is shown in Figure 6.5; this schema will always store data in the First Normal Form. The `Concept_Instances` table stores the name of the concept and, in so doing, creates an identifier `Int_id` for each instance. A number of entries in the `Slot_Values` table are then added for each fact. Each has its name stored in the `Slot_Name` field and literal values, such as text or numbers, are stored within the `Slot_Abs_Value` field. Where the slot is a fact itself, it is also stored within the `Concept_Instances` table in the same way, but in this case the `Slot_Concept_Value` field in the

Slot_Values table is set to the Int_id of that fact. Where the data is stored as an absolute value, the type – such as String, Date or Integer – is also stored in the Slot_Type field. The remaining field is Slot_Is_Sequenced for cases where the slot value is an array of and there may be several slot values for the same name. The Slot_Is_Sequenced field signifies that the slot is an array type and should be expressed in FIPA messages as such. Arrays are ordered using the identifier which is numerical. The natural ordering of the identifier therefore corresponds to array ordering.

This structure is capable of storing all of the fact-based data that can exist within any agent-based system and reconstructing them to send them to other interested agents. There are clearly limits on the size of literal values, in this case 4 kiloBytes, but subject to these restrictions, this is a suitable method of storage. Another potentially valuable property of this technique is that because the structure is also stored in the database, the structure does not need to be known ahead of data storage taking place. This simplifies deployment with there being no requirement for database customisation for the ontologies it will be used with.

One important consideration with this storage method is that of handling polymorphism. Since only the concrete type for the concept is stored within the database, searching for supertypes will not automatically find subtypes. This is handled by providing a processor. This line, from the IsA predicate, shows how this is achieved:

```
archive.predicate.IsA.permissibleTypes.processor=
    uk.ac.strath.eee.be.util.PermissibleTypesProcessor
```

Here, a Java class implementing an AttributeProcessor interface is specified and the value for the permissibleTypes variable of the IsA predicate can be transformed in any arbitrary way by the specified handler, PermissibleTypesProcessor. In this case, inheritance is handled by creating an appropriate SQL query. The PermissibleTypesProcessor uses the class hierarchy, which mirrors the ontology hierarchy, to construct a list of possible types. Since this processor uses, as its source, a set of Java classes, multiple inheritance cannot be covered. However, an alternative processor could be constructed to do this and the storage aspects of this technique therefore meet all of the requirements of the application.

This technique is portable between relational database systems; it is sufficiently simple to be supported by all databases, the only requirement being that the SQL fragments in the archive configuration file are correct for the platform in use.

In order to show the operation of the simple storage technique, consider the following message sent via FIPA-SL:

```
(HasIdentifier :variable (FGLT_Observation :logLevel 75
    :description (sequence #0 Test Fuel Grab Load Trace)
    :startTime 2009-01-01 :reactorComponent (FuelChannel
    :reactor (Reactor :belongsTo (NuclearPowerStation
    :componentIdentifier TST) :componentIdentifier 1)
    :componentIdentifier AB12)))
```

Within this SL, there are four concept instances: `FGLT_Observation`, `FuelChannel`, `Reactor` and `NuclearPowerStation`. The `FGLT_Observation` contains a `logLevel` slot, a `description` sequence, a `startTime` and a `reactorComponent`. It is the `reactorComponent` slot that contains the `FuelChannel` concept instance and this instance itself contains a `reactor` instance and a `componentIdentifier`. These instances have been passed along with the predicate `hasIdentifier` that the archive agent interprets as a request to store the concept.

The simple storage technique extracts all of this information and stores it in the Concept_Instances and Slot_Values tables as shown in 6.7 and 6.8 respectively. The ontology table will contain a single entry with the name of this ontology and the Int_id assigned is "1".

These tables now contain all information from the original SL message, but in a structured data store that can then be queried. The archive agent is then configured with a series of predicate SQL fragments – these the meaning of a predicate to be implemented as a query within the RDBMS storing the data.

```
archive.predicate.OccurredAfter.sql=SELECT concept_id FROM
    slot_values WHERE slot_name = 'startTime' AND
    slot_abs_value > '#startTime'
archive.predicate.OccurredAfter.startTime=OCCURREDAFTER_TIME
```

Int_id	*Concept_Name*	*Ontology*
1	NuclearPowerStation	1
2	Reactor	1
3	FuelChannel	1
4	FGLT_Observatin	1

Table 6.7: Example Concept_Instances table

Int_id	*Concept_Id*	*Slot_Name*	*Slot_Is_Sequenced*	*Slot_Concept_Value*	*Slot_Type*	*Slot_Abs_Value*
1	1	componentIdentifier	0	NULL	BO_String	TST
2	2	belongsTo	0	1	NULL	NULL
3	2	componentIdentifier	0	NULL	BO_Integer	1
4	3	reactor	0	2	NULL	NULL
5	3	componentIdentifier	0	NULL	BO_String	AB12
6	4	reactorComponent	0	3	NULL	NULL
7	4	startTime	0	NULL	BO_String	2009-01-01
8	3	description	1	NULL	BO_String	Test Fuel Grab Load Trace
9	3	logLevel	0	NULL	BO_Integer	75

Table 6.8: Example Slot_Values table

```
archive.predicate.AtStation.sql=SELECT concept_id FROM
    slot_values WHERE slot_name='station' and slot_concept_value
    IN (SELECT concept_id FROM slot_values WHERE slot_name =
    'componentIdentifier' AND slot_abs_value = '#stationID')
archive.predicate.AtStation.stationID=
    ATSTATION_STATION.POWERSTATION_COMPONENTIDENTIFIER
```

The archive agent can the process *QUERY-REF* SL message using the predicate definitions in the configuration file. Take the case of the following SL message:

```
((all ?x (and (OccurredAfter 20090101T000000000Z ?x) (AtStation
    (SteamCyclePowerStation :componentIdentifier TST)))))
```

This is processed by using standard set handling operations available within database systems to implement the *and*, *or* and *not* set handling functions along with the predicate fragments described above. This results in the following SQL query:

```
SELECT int_id FROM concept_instances WHERE int_id IN (SELECT
    concept_id FROM slot_values WHERE slot_name = 'startTime'
    AND slot_abs_value > '20081231T000000000Z') AND int_id IN
    (SELECT concept_id FROM slot_values WHERE
    slot_name='station' and slot_concept_value IN (SELECT
    concept_id FROM slot_values WHERE slot_name =
    'componentIdentifier' AND slot_abs_value = 'TST'))
```

When processed against the tables 6.7 and 6.8, this query return the event previously stored.

6.4.2.2 Object-oriented data storage

Object oriented data storage is possible in modern relational databases and standards for these exist. To use these features, the object types used within an application need to be created within the database and the software can then map between objects in the ontology and the database.

As previously noted, concepts within an ontology are analogous to objects within object-oriented programming languages thus attempting to store this information within an object-oriented database is a reasonable approach. As already discussed, the biggest difference is that within an ontology classes can have multiple superclasses. Despite this, experience with the COMMAS and PEDA applications described in Section 5.3.7 suggests that many ontologies do not feature multiple-inheritance and, as such, an attempt was made to use this technology.

The ORDBMS data storage layer for the archive agent operates in a different manner to the simple storage implementation of the storage layer described in Section 6.4.2.1. The ORDBMS technique requires that a dedicated schema be exported for the data store and, once created, the system is capable of storing objects within the schema.

There are two options for creating the ORDBMS schema; it could be created in advance using the ontology sources, or it could be generated from the data being written to the database as messages are received by the archive agent. However, the latter method is not considered to be robust. This is because when data is sent by an agent, it is not a requirement that all slots be sent. As such, empty slots are not communicated and this would mean that the assumed schema may be incorrect. Consequently every fact arriving for storage or for querying would require checking against the current schema and schema changes would be required where there were incompatibilities. This would be very slow; schema changes can take several seconds per change and the checking process would be an unnecessary overhead.

Given that the predicates that would be configured would also be dependent upon the schema, resulting in a firm tie between schema gener-

ation and archive configuration. The recommended option is therefore to generate the schema at the same time as the ontology is generated. This may result in tables that are never used for storage, but this is a small issue compared to the benefit of storage being available quickly at runtime. Due to the tight binding between the ontology and the database schema, multiple ontologies are not supported by this storage technique.

This technique is less portable between relational database systems than the simple data storage implementation due to a lack of object-relational support in many database systems. Whilst the overall approach used here is straightforward, SQL which specifically supports object-oriented databases must be used, and the syntax of these vary more between vendors than relational SQL.

The ORDBMS approach differs from the other two approaches in that only a single table is used, and that table is declared as being able to store all concepts. The data is then presented as a set of nested tables. Thus, the example data from the previous section would be stored in as shown in Table 6.6 way in for the ORDBMS backend.

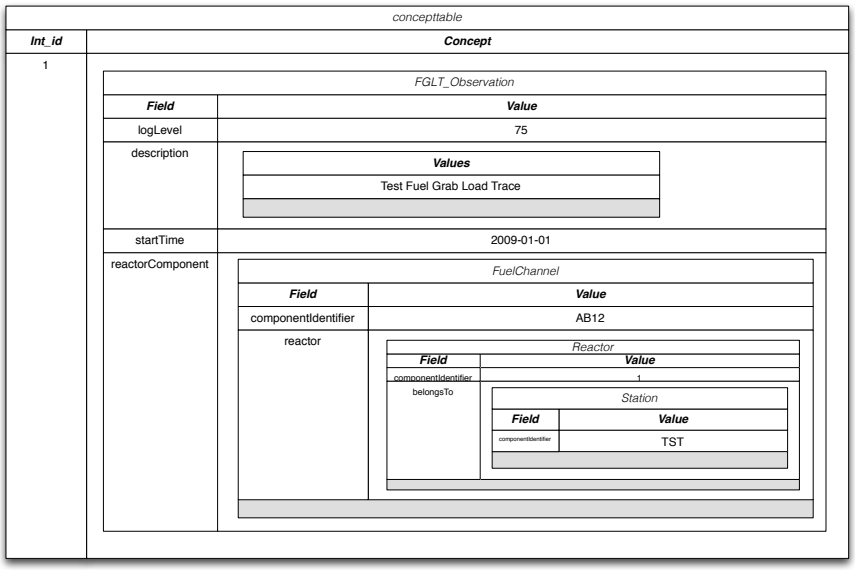

Figure 6.6: ORDBMS Nested Data Storage

In addition to the different underlying storage, this system is also queried in a different way. Example configurations for handling the predicates in this case are:

```
archive.predicate.AtStation.oql=select value from concepttable
    where TREAT(value AS Condition_Monitoring_Observati).startTime#
    >‘#startTime’
archive.predicate.OccurredAfter.startTime=OCCURREDAFTER_TIME

archive.predicate.AtStation.oql=select value from concepttable
    where TREAT(value AS Reactor_Observation)
    .station#.componentIdentifier#=‘#stationID’
archive.predicate.AtStation.stationID=
    ATSTATION_STATION.POWERSTATION_COMPONENTIDENTIFIER
```

In this case, the syntax for the queries is quite different; instead of using subselect queries as in the previous case, the . operator is used to check fields within subclasses. Apart from the syntactical differences though, the operation is identical with an SQL query representing the SL query being built using the predicate fragments.

6.4.2.3 Using the Java Persistence API

The object-oriented databases detailed in the previous section are a potential solution to storing object-based data, but these databases are far from common. Support for object-based storage is limited and the structures are not always readily usable for other applications. An alternative to storing object-relational data using standard relational databases rather than object-oriented databases was therefore developed by an expert group as part of the Java Community Process.

The Java Persistence API (JPA) is an application programming interface for the Java platform which aims to simplify the mapping of objects to standard relational databases. This is carried out by creating a model of the object hierarchy to be stored within the persistence engine. This engine is then responsible for the four CRUD operations of reading, persisting, merging and removing objects from the underlying datastore. This is similar to the model described above for the ORDBMS data store.

Within this API, it is left to developers to make decisions on the best way to store certain data types. For data types declared as a string, for example, the maximum length of the string is required and this is not declared in the ontology. Similarly, indexing is not a concern within an

ontology, but can have a significant effect on performance, particularly on large databases.

The storage of a class hierarchy is also a concern in mapping various subclasses into a relational database, particularly in the multi-agent application where the varying levels of abstraction offered by hierarchies are frequently used. JPA provides three strategies for handling such hierarchies – one table per subclass, one table per class hierarchy and joined table per subtype. These allow for all common class hierarchy structures to be stored within the datastore.

The benefit of the JPA is that the database schema built and used can be implemented in a standard RDBMS which can be accessed by other clients, and where querying and modifying the database is well understood. The main drawback is that the table structures and mappings to classes must be known in advance. This is identical to the problem encountered in prototyping the object-oriented model. The solution is slightly different in that the code generator which creates the ontology objects can annotate the generated classes, making them immediately ready for use within the database. This information can be appended on to the ontology classes generated by the JADE BeanGenerator [jad09]. JPA compatible classes can then be readily generated by adapting the open-source JADE BeanGenerator tool. Some JPA implementations can also use this information to directly generate the database schema.

There are a number of JPA implementations including Oracle Toplink, Hibernate and OpenJPA. The sample implementation of a JPA-based storage engine for the archive agent was based upon the open-source Hibernate [hib09] storage engine. This implementation is capable of generating the database schema meaning that the original ontology definitions also define the database with no additional effort. This means that the initial prototype of this technique was implemented in the same manner as the ORDBMS technique (Section 6.4.2.2), where the schema for the database is fixed, and the compatible items to be stored are defined ahead of time. Again, this restricts the system to a fixed version of the ontology.

This technique is very portable between relational database systems; even more so than the simple approach outlined in Section 6.4.2.1. Hibernate, for example, uses an intermediate query language rather than database specific SQL. This Hibernate Query Language (HQL) is database

agnostic and various dialects translate from HQL into each databases variant of SQL. Hibernate also supports database optimisations for the retrieval of data which can further increase system performance.

In this case, there is a single database table for each concept hierarchy within the ontology, so in the previous example, there will be a standard database table each for stations, reactors, components and observations, with each containing a field allowing the particular sub-type to be identified. This functions as a hybrid between the simple storage and ORDBMS storage techniques.

Once again, this data storage technique builds queries using the incoming SL. In this case, a query language specific to JPA is used, but following the same method as the simple storage and ORDBMS examples.

6.4.3 Comparison of storage options

The storage options outlined in the previous section are all capable of storing the data. This section compares the theoretical advantages and disadvantages of each of these options.

The conclusions from this section are summarised in Table 6.9 which shows that there is no single storage option which solves the problem of both storing ontology facts which may have multiple inheritance and doing so in a way that can be readily queried, taking into account the polymorphism inherent in all ontologies.

A final consideration when comparing these potential storage approaches is configuration and portability between approaches. As noted in each section, the query language used varies; in one case SQL is used, object-oriented SQL in the next and HQL in the final case. These query languages are all different. This means that the choice of storage also affects the translation layer between ontology and data storage.

The conclusions, from this analysis, is that wherever multiple inheritance is used within an ontology, the only compatible solution presented is the simple data storage technique. In other circumstances, the object-relational or JPA methods are suitable.

When the programming language is Java or .Net, it is further recommended that the JPA engine, or Hibernate's port of JPA for .Net, is used. This provides a database which is portable between vendors. For all other

Storage option	*Advantages*	*Disadvantages*
Simple tables	• No need to define schema in advance • Can store multiple inheritance readily • Compatible with all relational databases • Available across programming languages	• Queries can be slow/optimisation difficult • Difficult to re-use data • Polymorphism difficult to query
Object-relational	• Quick to store and query • Polymorphism handled natively • Available across programming languages	• Few implementations • No multiple-inheritance support • Schema required in advance
JPA	• Many implementations and good database support • Quick to store and query • Query optimisation built-in • Polymorphism handled by JPA providers • Easy to re-use data	• No multiple inheritance support • Schema required upfront • Only available in Java (though Hibernate ported to .Net)

Table 6.9: Comparison of storage options

instances, object oriented storage is recommended as the best option.

6.4.4 Case study: COMMAS SuperGEN V [1]

In order to prove the operation of the generic archive agent that was developed after research in terms of database and MAS theory and operations, the archive agent was handed over to another research project to assess the suitability and the claim that this was a generic archive agent. The project, COMMAS SuperGEN V, utilised the archive agent and is described in [CRMM09] although this paper does not cover the implementation aspects of utilising the archive agent.

Whilst the SuperGEN COMMAS application is also a condition monitoring application, it differs from the core monitoring application as it does not use a bespoke ontology but instead uses an ontology based upon the Common Information Model (CIM). CIM was designed for information interchange between energy management systems [McM07] and as such contains many concepts. As it is a generic model, several concepts are required in order to represent a single item. The use of CIM as an ontology should allow COMMAS to be readily applied across power system equipment. A case study explaining how an earlier incarnation of COMMAS was extended by using CIM as an upper ontology can be found in [CDM05].

Using CIM as an ontology is beneficial as it promotes standardisation allowing interoperable agents to be created. If a library of interoperable agents were created, it should then be possible to quickly create condition monitoring solutions by creating a MAS comprising the appropriate generic agents. Additional agents may be required at the periphery of the system in order to import data into the system ontology and to present the information in an appropriate manner, but the reasoning agents could be re-used across many domains.

As an example, a small part of this ontology is shown in Figure 6.7. It can be seen here that each item within a power system is represented as a `PowerSystemResource` and these are identified not through slots directly in

[1] The application-specific work undertaken in this section was carried out by Dr Victoria Catterson. This section supports that the research work carried out and detailed in Chapter 6 in respect of a generic archive agent is generic and applicable to other agent-based systems.

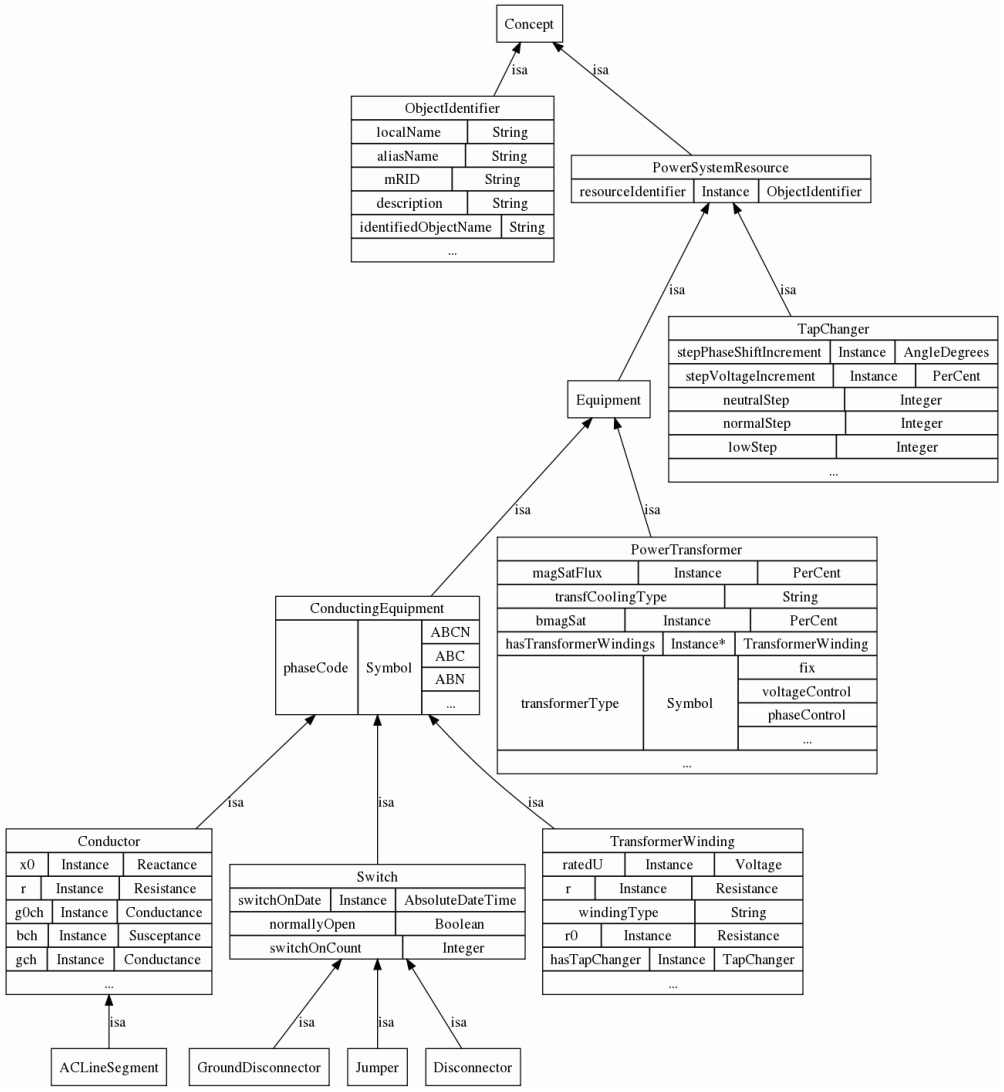

Figure 6.7: Excerpt from SuperGEN V Ontology

this instance, but rather through a `resourceIdentifier` slot which contains an instance of an `ObjectIdentifier`. To search for a single object therefore requires looking up a `PowerSystemResource`, then the `resourceIdentifier` slot for that resource, followed by the slots in the `ObjectIdentifier` object which allow the `PowerSystemResource` to be identified. This process involves a minimum of 3 database joins to simply identify a resource. Deeper queries require additional joins and the system limit can be readily reached.

CIM's flexibility resulted in a problem for the original archive agent implementation when used in this application; the extensive structures required to represent a relatively simple object meant that the two-table simple storage mechanism employed required several entries in the concept instance table and several entries in the slot values table. Since querying across several concepts requires that the same database tables are used to find the properties of each object, database join operations must be used. Each join uses database resources, so there is a limit to these and the open source MySQL database engine used in this project supports a maximum of 61 join operations. As the simple backend requires join operations to both query structure and data values, this limit can be reached very quickly, particularly where the archive agent was trying to query against every slot in a fact in order to check whether it had already been stored via the `QUERY-IF` performative.

In order to overcome the limitations in the database and to enhance the speed of querying whether something had already been stored, the system was altered to store a hash value along with each concept. This hash value is computed when the concept is stored and allows new concepts to be quickly compared with all values in the system.

Overall, the SuperGEN V COMMAS implementation helps prove that a generic archive agent is feasible and that it can be transported between different applications.

6.5 Conclusion

This chapter has described how database systems are commonly used for the storage of data. It has described the current storage techniques used in agent-based systems and shown how database features can be utilised

to perform this storage on a larger scale, along with the similarities and dissimilarities between ontology and database schemas.

A generic archive agent has also been described showing how different aspects of this agent can be generalised and what aspects are application-specific. Three different solutions to persistent storage have been described and compared, concluding that there is no perfect solution to agent-based data storage. A series of recommendations have been made to indicate when different options are suitable.

In the case of the core monitoring application where the ontology can be constructed without multiple inheritance, any of the storage mechanisms would be suitable. Polymorphism, for the multiple observation types, is a key part of the core monitoring application though which makes the simple storage technique less suitable than either the object-oriented or object-relational methods. Due to the widespread support for object-relational storage in common databases and the limited number of object-oriented databases, object-relational is the most suitable for the core monitoring application.

This chapter has shown one use of the archive agent whilst Chapter 7 further demonstrates this agent within the core monitoring context. These two case studies use very different ontologies and the queries the agent is subjected to by other agents in the multi-agent system have different levels of complexity. Together, these case studies give a comparison of each of these storage technologies within the context of real-world agent-based applications.

Chapter 7

Deployment case study

7.1 Overview

This thesis has explained the need for condition monitoring within the nuclear domain with particular reference to AGR reactors, which are the predominant design in use in the UK. Chapter 4 showed how available data could be analysed using statistical techniques adapted for the application, and Chapter 5 considered the use of multi-agent systems for the implementation. Chapter 6 showed the design of a generic agent that could be used to archive data for any application. This chapter describes the core monitoring system that has been been built using the novel aspects designed and refined within these preceding chapters.

The system, IMAPS, implements a full graphite-core monitoring application as considered throughout the thesis using, as a platform, agent-based systems and the archive agent design introduced and assessed for flexibility in Chapter 6.

7.2 Case study: IMAPS

The Intelligent Monitoring Assessment Panel System (IMAPS) was built upon the JADE agent-based framework, as recommended in Section 5.3.5, and using the archive agent described in Chapter 6. The system can assess the monitoring data captured at Monitoring Assessment Panels using the agglomerative clustering techniques described in Chapter 4. Figure 7.1 shows how the concepts from the preceding chapters are brought together

in the resulting system.

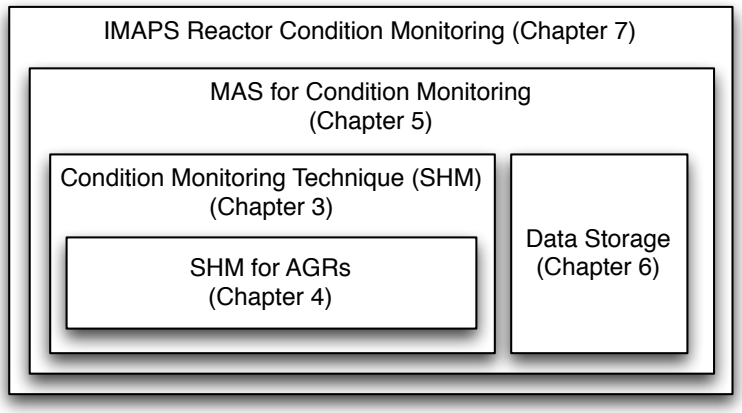

Figure 7.1: Thesis arrangement with respect to IMAPS

The original vision for this multi-agent system (MAS) is shown in Figure 7.2.

Most of this original design has been implemented; those portions not yet implemented are the portions which would interface directly with data sources within the power station and feed data to the MAS for automated analysis. As outlined in Section 2.3, access to this data is restricted on a segregated network in order to protect the operational networks at the plants.

7.2.1 IMAPS agents

IMAPS is a multi-agent system comprising the following agents:

- An `ArchiveAgent` responsible for storing data for the system
- A `ClusterAnalysisAgent` which can analyse data stored within the system
- Several `UserAgent` instances can be started; each one stores the data and privileges for a user and allows them to communicate with other agents. This also allows data to be entered into the agent-based system.

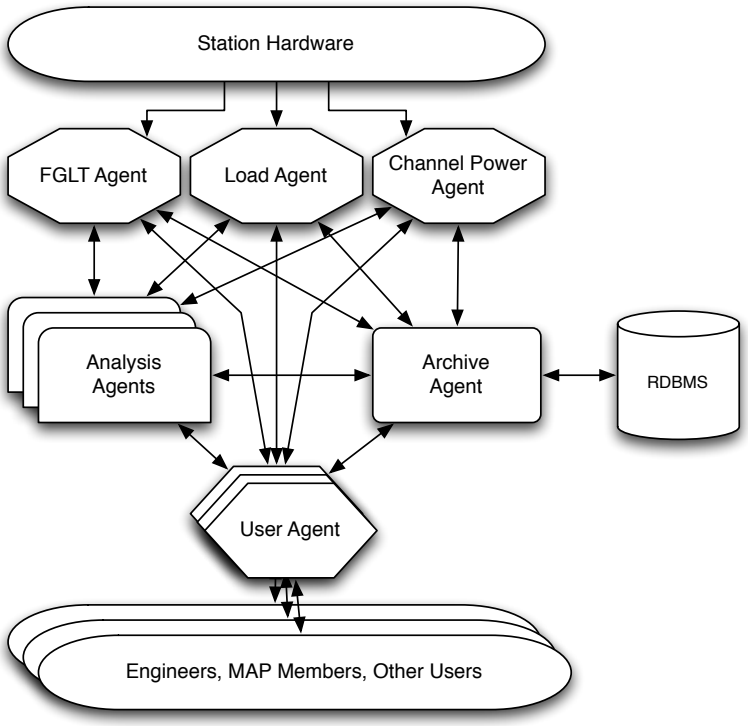

Figure 7.2: Original IMAPS vision

7.2.1.1 Archive agent

The archive agent was extensively described in Chapter 6 and was used initially with the simple storage technique. The key responsibility of this agent is to store data and to provide access to that data using standard agent queries. This involves interfacing between information expressed in an ontology and one of several relational database options. The agent is configured with a single behaviour:

- `FactStorage`: responds to all communications regarding the storage and retrieval of facts from the underlying storage

This agent must be configured prior to use, and has been configured to respond to several agent `Actions` which allow other agents to instruct the storage, updating and deletion of information. In addition to these information maintenance abilities, the behaviour is also configured for the all predicates within the ontology. This allows the agent to find matching facts for any arbitrary query.

7.2.1.2 Cluster analysis agent

The cluster analysis agent implements the clustering techniques described in Chapter 4. This agent is also implemented with a single behaviour:

- `CoreClusterAnalysis`: responds to requests to undertake cluster analysis

This single behaviour implements the cluster analysis, and the analysis is configurable through the requests sent by other agents. Once there has been a request from another agent to undertake an analysis, this agent will contact the data provider, normally the archive agent, and gather the required data for analysis.

Having collected the data, the behaviour will perform the analysis and respond to the initiator with the result of the analysis.

7.2.1.3 User agent

The user agent is, at the present time, the route by which all data and information is entered into and retrieved from the system by engineers. An instance of the IMAPS user agent is started for each user that logs into the system and behaviours are added to this agent based upon their permissions.

- `AnalysisManager`: performs discovery of analysis techniques available and provides an interface to request analyses to be undertaken and view the results
- `EventLogger`: allows a user to log new information within the system by creating ontology objects and submitting them to the archive
- `FactManagement`: allows a user to request that data be changed or removed by the archive
- `FactSearch`: allows a user to perform a search for data; these are executed as one-off searches based upon input parameters and the output is returned to the client as tabulated output
- `InteractiveView`: used when an engineer requests a customised view of the data to retrieve applicable data in a format for graphical rendering

- `LogFromFile`: allows a user to upload a file containing multiple events and have them all entered into the system
- `MapManager`: allows users to record information pertaining to the MAP meetings

The behaviours available within this agent were developed over several years. The system started with the `EventLogger` and `FactSearch` behaviours to allow logging of data into the system as early as possible. As MAPs completed further analysis of potential issues, there was a need for editing to update previous observation details, and this required the addition of the `FactManagement` behaviour. Bulk data logging was later added, by the `LogFromFile` behaviour, followed by flexible means of visualising the data in the `InteractiveView`. This allow engineers to treat each observation type as a layer and they can see the contribution of different data to the predicted core state. A `MapManager` behaviour was then added allowing data to be searched based upon MAP periods and finally, when the analysis capability was available, the `AnalysesManager` behaviour was added to allow the user to request that the analyses be undertaken.

7.2.2 Ontology

In choosing a suitable ontology for the application, consideration was given to using a pre-existing ontology that may exist in some other standards. The only candidate, however, was the Common Information Model. Whilst this provides the capability to represent generation characteristics, it does not model nuclear components and is therefore unsuitable.

As no suitable domain model exists, IMAPS uses a bespoke ontology. This was created with consideration for several monitoring projects being undertaken which could later allow extension of the agent-based system to include vibration analysis on other plant within the station. As such, it contains concepts representing conventional steam-cycle stations and the nuclear-specific concepts such as reactor and fuel channel. The main `Concepts` used in the ontology are shown in Figure 7.3.

A more detailed view showing the monitoring observation hierarchy, representing each of the manual analyses that are undertaken to create the different observation types, is shown in Figure 7.4.

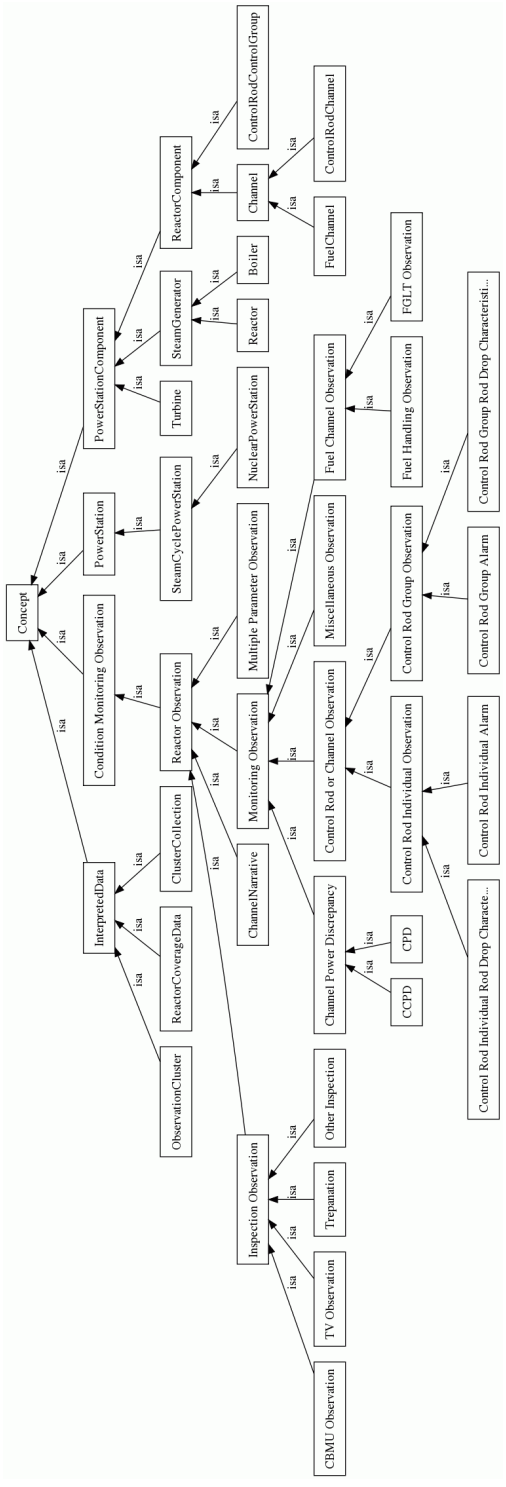

Figure 7.3: IMAPS main concepts overview

Figure 7.4: IMAPS ontology extract

Operation	*Storage*	*Engine 1 time*	*Engine 2 time*
Store Data	Simple	16m 09s	11m 39s
	Object-relational	*Not compatible*	3m 15s
	Object-oriented	6m 05s	*Not compatible*
Load Data	Simple	1m 04s	1m 14s
	Object-relational	*Not compatible*	1m 13s
	Object-oriented	0m 17s	*Not compatible*

Table 7.1: Time to store and load data using each backend

The IMAPS ontology additionally has a number of `AgentActions` defined. These allow the implementation of the CRUD operations defined in Chapter 6.

7.2.3 Data archiving within IMAPS

The data archival operations required for IMAPS allow testing of the three different data storage options outlined in Chapter 6. All three storage options were tested for this application and experience with each is reported below.

Prior to describing each of these, timed statistics are presented for storing 3,340 observations using each technique. Timings are also provided for the retrieval of data for a single station and are presented in Table 7.1. It was intended that the tests should all use the same database engine, but incompatibilities between the storage backend and the chosen engines meant that not all backends were compatible with all databases. The data in this table were therefore collected using two different database engines but in each case running on the same hardware to give, as much as possible, a comparison of only the storage technique. The archive was modified only to switch configuration between the storage backend and no other agents within the system were modified. The ontology initially generated and used was designed to be compatible with all backends, so this did not change either.

As can be seen, object-oriented storage is clearly the quickest method for storage. It is likely that this can be directly attributed to a single database query being required to store or load all the required data. Both the simple storage and object-relational methods can require multiple queries depend-

ing on the the ontology and queries used.

7.2.3.1 Simple storage backend

The simple storage backend was the first implemented for this system and was used extensively during the trial. The principal concerns with this method of storing data were retrieval times and the opaque manner in which data is stored.

In practice, the system initially performed reasonably well, but soon slowed as more data was added to the system. The concern over the data storage method leading to data that was difficult to view was found only to be an issue during development. The technique was found to require the least configuration and, as noted in Section 6.4.2.1, is the most flexible being able to operate on a wide range of database engines.

In order to resolve the issue of debugging the data during development and to speed up querying the database, accelerating code was added to the front-end agents and a database view was used. This allowed the front-end agents that present the data to the user to retrieve data directly within the database instead of obtaining it via the archive agent using FIPA messaging. This was only available for a small set of queries. The custom view in the database, which presented the data from the underlying structure as a standard structured table, is a standard technique used in database design for presenting data stored in different ways. FIPA messaging to the archive agent was used for all other interactions, including where analysis agents retrieve data.

The acceleration method used successfully reduced the times required to render core maps. In the previous example, only the data retrieval times are counted; by moving to custom front-end code, retrieval times for a whole page on the same systems were reduced from tens of seconds to around 3 seconds; this is considered to be an acceptable time to load and display the information.

The same method of providing different views could be used to make the data available to other applications as necessary.

7.2.3.2 Object-relational backend

The object-relational data store used the open source Hibernate implementation.

The first impression of the object-relational backend, on starting the archive agent, is that a significant degree of processing is being undertaken. This is because the query engine builds an internal model of the data prior to allowing queries to be undertaken; this is then used to aid query parsing. This does lead to longer startup times than either of the other techniques but since it only occurs when the agent starts, which is relatively infrequent, it is not considered to be a concern.

Moving from the simple backend to the object relational backend did require additional configuration; all of the queries from the simple backend had to be rewritten using the Hibernate Query Language (HQL) in place of SQL used previously. It was immediately apparent, however, that it was far simpler to express the queries in this language, reducing the burden on the system designer and the likelihood of mistakes.

The object-relational backend successfully stored the data very quickly when compared with both alternative techniques. In testing, it was noted that the object-relational technique could also store circular references successfully as Hibernate features circular reference detection and automatically resolves these. These were later removed to allow a fair comparison with the object-oriented backend which could not handle these.

A useful feature of Hibernate was also used for the testing; Hibernate can generate a database schema based upon the object-model that it will store. Since the object model was defined within the ontology, and this was subsequently exported to the class files used in the implementation, it was also possible to generate the full database schema from the ontology without manual intervention.

The querying aspects of this storage technique were reasonable. More configuration was required for this backend than for the simple model, and part of this could limit the querying ability of this backend. This is the only backend which utilises separate database tables for each Concept in the ontology. This means that the engine needs to be told which table to use for the top-level query and the need to provide this information could prove time consuming and difficult in large systems. Consideration of this

difficulty could be undertaken in future work.

The potential for reuse of data stored with this backend is great; this model provides standard database tables that most developers will be familiar with. The tables are not completely normalised, as schema information to allow this isn't necessarily available, but it is all stored and accessible.

It is also noted that Hibernate was designed to allow existing databases to be mapped onto an object schema. This means that if an existing database can be represented by an ontology, using it in this way may be a better solution to accessing existing databases than wrapping the database in an agent, as has been seen in systems to date. This approach should allow greater flexibility and access to the data within that store.

7.2.3.3 Object-oriented backend

The object-oriented backend posed the most problems for implementation. The storage method used requires that, as well as having simple inheritance between classes in the concept hierarchy, there are no circular references. In order to enforce this, the ontology was generated without circular references. Frequently ontologies do feature circular references like "contains" and "belongsTo" so either further development of this backend would be required to support this, or there would be a major restriction on its use. The development to support these may result in having to use multiple queries which may impact on the transaction times.

The requirement for class definitions in the database meant that this system did need an additional step to prepare the database for use; there is no simple manner to do this as for Hibernate. Additionally, the definition of the types within the database meant that, on storing, parameters had to be passed in a specific order, with no exclusions. This required additional coding and configuration so the archive agent was able to make appropriate requests to the database.

After these initial difficulties, the object-oriented backend did successfully store all of the data. The storage time was second best and retrieval times were the best when tested.

In all, the object-oriented technique is interesting and functional, but the consideration from this test was that it is not yet suitable for regular

use across multi-agent systems due to the limitation placed upon the agent system developer.

7.2.3.4 Storage recommendations

Having tested the various storage technologies, it is clear that there is a trade-off between the features of the ontology used and the speed and features available in the storage engine. The simple storage technique, storing both structure and data within the database is relatively slow, but requires the least configuration, is the most flexible in terms of data that can be stored and works with the widest array of database engines.

Mapping ontology structure to database structure is demonstrated in both the object relational and object oriented techniques. There are two problems with this; one is that type names within the ontology must conform to the limitations of the database engine and the other is that the mapping incurs an additional configuration burden. Despite this, the approach is capable of producing a very fast storage solution for a multi-agent system. Where there are many facts to be stored and queried, working within the limitations of the object-oriented approach could be very beneficial.

The object-relational data storage approach is also seen to work, and particularly for storing data, it performs reasonably well. This approach still requires configuration, but in the case of pre-existing databases or where the data store will need to be reused, it appears to be a suitable choice.

Following the experimentation undertaken with the three different data storage layers for the archive agent, and by considering the fundamental data storage properties it is concluded that the simple storage model is the most compatible and therefore appropriate for most agent systems. This choice is a trade-off, but the simple storage model poses no restrictions on the agent system designers, provides for simple configuration, but does require some care in the creation of predicate mappings. Where multi-agent system designers have specific requirements, such as using an existing database or requiring the greatest degree of performance rather than flexibility, the alternatives presented may prove better.

7.2.4 Data analysis within IMAPS

The first analysis agent in IMAPS performs the cluster analysis described in Chapter 4. To perform the analysis on all available data, the engineer simply needs to choose the station and reactor upon which to run the analysis. Data can also be analysed over any time period by selecting start and end dates.

The analysis was tested by automatically performing the same analysis on the same data as shown in Section 4.6. The result is computed and the clusters are displayed in size order as shown in Figure 7.5.

Analysis Result

Cluster 0

26/12/2007	Control Rod Individual Alarm	FG24	Green
09/05/2008	Control Rod Individual Alarm	FG20	Amber
08/05/2008	Control Rod Individual Alarm	FG20	Amber
28/07/2008	CCPD	C19	Green
18/08/2008	CCPD	C19	Green
12/08/2008	CCPD	E17	Green
22/07/2008	CCPD	E17	Green
18/09/2008	Control Rod Individual Alarm	DE16	Green
21/04/2008	CCPD	E17	Green
09/12/2008	CCPD	E17	Green
15/12/2008	Control Rod Individual Alarm	DE16	Green
27/01/2009	Control Rod Individual Alarm	DE16	Green
04/03/2009	Miscellaneous Observation	DE16	Green

Cluster 1

12/05/2008	Control Rod Individual Alarm	VW12	Green

Figure 7.5: IMAPS cluster analysis results

The clusters are identical to those found by performing the analysis manually, though IMAPS presents these with the original grading colours assigned by the MAP process described in Section 2.5.1. Analysis of the clusters that form show that some contain only observations graded green whereas others contain amber graded observations. These can now be manually analysed to assess which cause concern and may be related to core distortion.

In order to test this approach, all of the clusters identified for the provided data were assessed to identify whether the analysis technique is capable of finding potential damage within the reactor cores.

For the data analysed in the Chapter 4 example, and displayed above, 30 clusters were identified by the software using the statistical limits chosen. These clusters vary from having 13 observations, including 2 amber grad-

ings, down to clusters containing just 2 green-graded observations. The top 14 clusters, all those containing more that 3 observations, are described in Table 7.2 along with the number of amber and green observations identified in each cluster.

The data from the cluster analysis proves that the technique is identifying where data is collocated within space and time. The further analysis conducted in order to populate Table 7.2 has provided further insight; clusters containing several observations on exactly the same channel from a single analysis are likely to indicate either a sensor failure or a fault within the analysis process such as that encountered with the PHOENIX code.

Despite this, there are some clusters that contain a mix of observation types, and deeper analysis and investigation is required to identify the root cause. Once further experience is gained with the analyses, it is recommended that further intelligent analysis agents can be added to analyse and track changes within the clusters, be it by the requirement for a minimum number of observation types, a minimum area of impact or different grades of observation. Knowledge engineering may be able to provide insight into this as experience with the clustering approach is gained, and a rule-based approach to the analysis could be readily developed. This would give a filter identifying only the important clusters to the engineers.

Whilst there is the possibility of this technique providing false positive results, as the entries described here appear to be, the original aim of this work was not to improve or automate an existing process, but rather to identify potential structural issues with the core. It is argued that the identification of these clusters succeeds in that goal, and that the methodology presented is capable suitably aggregating the data from the various processes, despite the need for further analysis of the clusters.

7.2.5 Appraisal

Whilst the academic aspects of this work are reported in this thesis, Section 2.5.1 explained the MAP process within EDF Energy and the resulting system was always intended to support that industrial work. The industrial need for an operational system that would allow data to be captured, stored and displayed initially, with further developments to add deeper insight and analysis of the data, directly led to the choice of an agent-

Cluster size	*Ambers*	*Greens*	*Description*
13	2	11	Around DE16 control channel, during the latter part of 2008. The majority of the observations were traced to a control rod actuator problem whilst there was a collocated thermocouple fault. No FGLT issues were included over the period, and it is suspected that this is not a structural issue.
10	0	10	All CPD observations, problem traced to an issue in the PHEONIX model due to different conditions caused by the reactor running at reduced power. Code corrected.
10	0	10	Included FGLT observations, but mainly Channel Power discrepancies caused by maintenance disabling autocontrol in that core area.
10	0	10	Predominantly channel power errors; caused by faulty thermocouple disrupting modelled values.
7	0	7	Control rod actuator problems, ultimately resolved by maintenance on the control rod assembly.
6	0	6	Identified as an instrumentation issue that caused rods to be placed under manual control. Resolved when instrumentation corrected.
5	3	2	Various contributing causes for this cluster; chain alarms mid-core may indicate core distortion, but a thermocouple fault and shuffled fuel causing CCPD modelling issues suggest this is not distortion related.
5	2	3	Various issues caused this item; a chain fault alarm that may indicate snagging or slow mechanism, but equally a gag indication failure and shuffled fuel contributed. Overall judgement suggests this is not distortion related.
5	0	5	Two separate issues, a wiring fault and shuffling of fuel stringers in the same area of the core caused this cluster.
5	0	5	All control rod alarms between 2 rods in the same area over an 11 month period. No other observations though; unlikely to indicate distortion.
4	2	2	4 chain fault alarms on control rods over an 8 month period. Restricted to a small area of the core and no other observation types suggest no distortion.
4	0	4	2 gag errors, and a rod fault contributed, though the FGLT analysis suggests minor cracking in the core.
4	0	4	3 control rod faults, and a small crack in the core; unlikely distortion related.
4	0	4	One note added to the channel and an FGLT with known cracks in addition to 2 mid-core issues with control rods over a 3 month period. This could potentially be core-related, but all observations are considered minor.
4	0	4	Various rod alarms on the same channel suggesting a control rod issue, with minor cracks noted in the FGLT.

Table 7.2: The 14 clusters with > 4 observations

based system that would allow analysis extensions to be added later. This approach of using the multi-agent approach to allow a staged implementation meeting the industrial needs is reported in [JMRT07]. A discussion of the future ideas for the system, such as automating the data collection process and allowing analysis to be undertaken on individual channels and quadrants as well as the reactor-wide example demonstrated here, are contained in [WJM11].

The IMAPS implementation has been used in two separate modes within EDF Energy, at all times supporting the MAP process. The two-stage approach was required due to a lack of prior data; the system was developed whilst the MAPs were being initiated so monitoring data was not immediately available and having the system available quickly allowed data to be gathered to develop further aspects of the system.

As a result of the lack of monitoring observations, the system was created initially as a data storage system allowing users to enter and view the MAP information. Restrictions were placed on who could add data by limiting the agent behaviours available to each user; these were managed separately from the agent system.

This version of the software was in use within EDF Energy from 2006 until 2009. It was recognised that the software and information should be available widely within EDF Energy and a project to "productise" the initial version was undertaken. A screenshot from the original system, showing a core map output, is shown in Figure 7.6.

The most significant issue found with the prototype software was that the data took considerable time to be displayed on users' computers. The software is web-based, but the time taken to query the archive agent, get the response and then send it to the client was unsuitable; too much information was being retrieved at each stage simply to render channels a particular colour. This process meant that a complete set of data was being retrieved for each reactor, encapsulated in an agent message, extracted and then encapsulated in another format for delivery to the web browser. This was found to take too long for a production system.

The issue of efficiency was addressed by removing the agent-based components from the front end in the production version of the code. Communication takes place directly with the backend database, as it did with PEDA [CDM05]. In order to resolve this problem, but maintain the

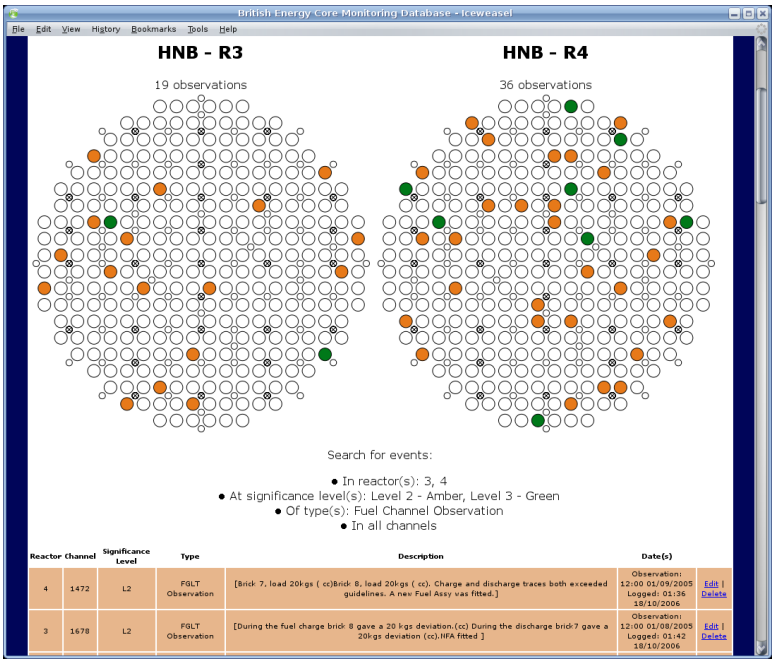

Figure 7.6: IMAPS prototype version screenshot

option of applying intelligent agents in a production system, the schema used within the database was that from the Hibernate model of the ontology. This means that the database can be used by an agent-based system once the technology has been developed to production standard.

The IMAPS system as currently installed does resolve the speed of access, but also adds support for validation of entered information, auditing processes to give confidence in the data stored within it and links to the corporate document management system giving rapid access to official documents to support each of the observations stored within the system. This aspect, even without the analysis component, makes for a compelling decision support system availing the inspection and monitoring information to those who need it.

Overall, IMAPS is considered to be a successful project; it has shown that the data gathered at MAPs can be successfully stored and retrieved within a database, but that this can be carried out quickly allowing various hypotheses and "what-if" scenarios to be considered. The system is now in regular use at two stations and has been trialled at another two that are still in the process of setting up their MAP processes. Support for the whole fleet of reactors has also been provided, ready for when the

remaining stations decide to engage in monitoring activities.

The implementation of the IMAPS system highlighted some shortcomings of the agent-based approach and these have been addressed in order to design an implement a working system. Through the IMAPS system, the operator has been given the ability to look at different aspects of the monitoring by configuring various views using the IMAPS software, a valuable feature even without the full SHM approach embedded.

The industrial assessment of the prototype also concluded that additional features were required to ensure quality data is stored in the system. This involved additional tracking during the logging process and ensuring that no single person should be able to add or modify data. This ensures that the data in the system, upon which operational decisions can be made, can be relied upon. Whilst providing this, the system also needs to record what each user has done to allow actions to be audited.

Despite these points for further development, the system has demonstrated that agent-based systems can deliver the data storage and analysis required to provide decision support for the operational management of nuclear reactors.

Today, IMAPS runs as a production system with the graphite core operators collecting, storing and presenting data and is valuable in this role. Work is being undertaken to deliver the structural monitoring approach reported here into the company in a manner that supports the operations at the appropriate quality level.

The operator has commented that the production version of this system is delivering benefits by making all relevant core data available to everyone in the company and that the verification of data within the system to their QA Grade 2 standard means that the data can be trusted for decision-making purposes. The ability to navigate, filter and export data, along with links directly into the document management system is beginning to find use in other areas beyond core monitoring where channel histories can be checked prior to maintenance operations and inspection campaigns. Further developments to the system have begun to address workflow, reducing the administrative burden of running the MAP meetings.

7.3 Conclusion

This chapter has shown how the generic archive agent described in Chapter 6 has been used within an agent-based condition monitoring system, as outlined in Chapter 5.

This chapter has also shown the effects of the different database backends upon the performance of information retrieval. It is concluded from these experiments that where retrieval performance is important, the object oriented backends are superior to the simple backend. In terms of storage, there is little to choose between the backends with each taking about the same amount of time. As is common with databases, however, the most common transaction is expected to be the querying.

The IMAPS project also implemented the structural monitoring techniques described in Chapter 4. This has proven the capability of agent-based technology to implement this type of analysis and this will be implemented with reactor operators in the future. The staged-implementation of the IMAPS system has also shown that agents are well-suited to building systems which will gain additional features over time.

Chapter 8

Conclusions, further work and recommendations

8.1 Conclusions

Condition monitoring has seen increased prevalence across the electrical generation industry and continuing pressures to increase safety, equipment availability and to reduce risk whilst improving overall financial performance show no signs of abatement. Chapter 2 introduced the motivation for this research highlighting that whilst many areas of the electrical industry have benefited from increased monitoring, graphite reactor cores have proved difficult to monitor due to the challenges of building sensors that operate within the harsh environmental conditions of the reactor.

Chapter 3 introduced the structural health monitoring (SHM) approach that has been demonstrated within other fields with life-safety considerations and showed that the damage in AGRs could be identified by collating data from multiple independent analysis processes using a statistical distance measure and explained how the four-stage SHM approach could be applied to the reactor core monitoring case.

Following the identification of a suitable approach to analysing the available monitoring data, Chapter 4 explored the limits that exist in terms of data collection and the underlying physical processes involved in core distortion. Consideration of these allowed the identification of limits that could reasonably be placed on the statistical approach for hierarchical clustering.

Chapter 5 looked at various system options that exist for the implementation of a prototype core monitoring system and concluded that, given the limited understanding of the monitoring data at the current time, that a Multi-Agent System (MAS) would be a suitable deployment mechanism. The MAS approach would then allow for future enhancements and expansion.

A discussion of MAS suitability identified a specific difficulty with the approach and Chapter 6 proposed a technique for interfacing any MAS with a relational database utilising a predicate mapping approach to handle the flexible and changing queries that could be experienced in a MAS environment.

Finally, Chapter 7 demonstrated how the overall agent-based technique implemented the various strategies developed throughout this thesis and showed how identification of spatially and temporally collocated events was realised.

It is ultimately concluded that the research has delivered the following novel contributions:

- An approach for the use of monitoring data collected during routine nuclear plant operation to identify structural defects within the reactor,
- Specification of how to use established structural health monitoring techniques with the discrete data available from reactor analyses using a cluster analysis approach,
- Identification of appropriate limits for the identification of structural defects within graphite reactor cores,
- A proposal to implement the core monitoring system as a multi-agent system allowing for the later integration of further analysis techniques,
- An implementation of a flexible and generic archive agent that can store arbitrary data within a multi-agent system,
- A case study demonstrating the use of the flexible archive agent within a transformer monitoring system,

- A case study of the use of the flexible archive agent within a multi-agent system for the detection of structural defects,
- A case study of the clusters identified by the process to demonstrate the effectiveness of the approach to detecting defects,
- A series of recommendations for how the cluster analysis process itself might be automated to further filter the information presented to end-users.

8.2 Further work

This thesis has been undertaken at a time when graphite reactor structural monitoring is a new topic. Additionally, the graphite cores being monitored are a small population, none of which has had to be decommissioned. However, it is essential to have confidence in the results, allowing systems to be placed in daily use for decision support. Therefore, and further work is recommended in a number of the key areas.

First, although the statistical analysis technique that has been used is sufficiently powerful to detect clusters of data, the data which is operated upon could be further preprocessed to remove the known issues. Such preprocessing has not been carried out at this stage as a conservatism, but a knowledge-based analysis system may be able to reduce the number of "negative" observations leading to clusters which could more often indicate core distortion.

The analysis has also been based upon simple measurements of the clearance distances involved. There are several projects underway to produce representative finite element models for the core. It may be possible to use these to search for reactor configurations that might give rise to observed data. This model-based approach could potentially give on-line core distortion information but would be based upon knowing the effects of various defect types and how these could manifest themselves in the data.

The cluster analysis could also be enhanced by providing a level of automatic analyses of the clusters. As identified in Chapter 7, the contents of the clusters is significant in terms of gradings, observation types and

which specific channel or channels the cluster contains data for. Automated analysis of these could assist engineers in focusing on particularly relevant clusters.

Additional work could also be undertaken on the statistical distances used. The distances measured are not directed within the existing model, as can be seen in the example in Section 4.6; the 6 month limit applies both before and after incidents resulting, in the example, in a 14 month period of data. Equally, additional data may be able to be added as further dimensions or analysing data at each observation level may yield more relevant results.

8.3 Recommendation

A final recommendation is now offered after reflecting upon the work reported in this thesis.

The requirement to infer the condition of nuclear reactors by using existing monitoring data is ultimately required due to lack of direct measurement techniques for the cores. Whilst there are unlikely to be more reactors of the AGR design built, it would have to be recommended that additional monitoring equipment be provided wherever possible on all reactor designs. As an example, vibration sensors mounted close to the reactor may be able to more closely mirror the processes undertaken in SHM for bridges and aircraft and allow those monitoring techniques to be directly applied to reactors.

It should be relatively simple to add such sensors to reactors, and doing so early in the operating lives of the reactors would provide a wealth of information about different normal operating modes, which should then improve the detection of defects in later life.

Bibliography

[Age92] Nuclear Energy Agency, editor. *In-core instrumentation and reactor core assessment: Proceedings of a Specialists' Meeting*. OECD, 1992.

[Age96] Nuclear Energy Agency, editor. *In-core instrumentation and reactor core assessment: Proceedings of a Specialist Meeting*. OECD, 1996.

[AGS+05] N. Achatz, J. Gorablenkow, U. Schichler, B. Hampton, and J. Pearson. Features and benefits of UHF partial discharge monitoring systems for GIS. In *Electrical Insulating Materials, 2005. (ISEIM 2005). Proceedings of 2005 International Symposium on*, volume 3, pages 722–725 Vol. 3, June 2005.

[Ahm87] K.M. Ahmed. The dynamic response of multi-layers AGR core brick arrays. *Nuclear Engineering and Design*, 104(1):1 – 66, 1987.

[Bar96] Ron Barron, editor. *Engineering Condition Monitoring: Practice, Methods and Applications*. Longman, 1996.

[Bea96] S. Beattie. Circuit breaker condition assessment by vibration and trip coil analysis. In *Monitors and Condition Assessment Equipment (Digest No. 1996/186), IEE Colloquium on*, pages 9/1–9/5, Dec 1996.

[Ben71] M. Benedict. Electric power from nuclear fission. *Proceedings of the National Academy of Sciences of the United States of America*, 68(8):1923–1930, 1971.

[Ben96] C. Bengtsson. Status and trends in transformer monitoring. *Power Delivery, IEEE Transactions on*, 11(3):1379–1384, Jul 1996.

[Bin90] F.J.L. Bindon. Sizewell B – the first of the UK's PWR power stations. *Power Engineering Journal*, 4(3):113–117, May 1990.

[BKB06] J. Bugge, S. Kjær, and R. Blum. High-efficiency coal-fired power plants development and perspectives. *Energy*, 31(10-11):1437–1445, 2006.

[Bow82] J. M. Bowerman. *The Safety of the AGR*. Central Electricity Generating Board and South of Scotland Electricity Board, 1982.

[Bri06] British Energy plc. Different types of nuclear power, 2006. `http://www.british-energy.com/opendocument.php?did=533`.

[Bro07] JMW Brownjohn. Structural health monitoring of civil infrastructure. *Philosophical Transactions A*, 365(1851):589, 2007.

[Cat06] V.M. Catterson. *Engineering Robustness, Flexibility, and Accuracy into a Multi-Agent System for Transformer Condition Monitoring*. PhD thesis, University of Strathclyde, December 2006.

[CB05a] Thomas. Connolly and Carolyn Begg. *Database Systems*, chapter 1, pages 12–14. Addison Wesley, 4 edition, 2005.

[CB05b] Thomas. Connolly and Carolyn Begg. *Database Systems*, chapter 13.5, pages 401–422. Addison Wesley, 4 edition, 2005.

[CBR07] A. Cole-Baker and J. Reed. Measurement of AGR graphite fuel brick shrinkage and channel distortion. *Management of Ageing in Graphite Reactor Cores*, page 201, 2007.

[CC10] B. Chen and H. H. Cheng. A review of the applications of agent technology in traffic and transportation systems. *Intelligent Transportation Systems, IEEE Transactions on*, PP(99):1 –13, 2010.

[CDM05] V.M. Catterson, E.M. Davidson, and S.D.J. McArthur. Issues in integrating existing multi-agent systems for power engineering applications. In *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*, pages 6 pp.–, Nov. 2005.

[CJRW96] R.F. Curtis, S. Jones, J. Reed, and A.J. Wickham. The Development of Direct Core Monitoring in Nuclear Electric plc. In *IAEA-TECDOC-901*, pages 91–104. IAEA, 1996.

[Cod70] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970.

[Cor91] D.D. Corkill. Blackboard systems. *AI expert*, 6(9):40–47, 1991.

[Cor03] Daniel D Corkill. Collaborating Software: Blackboard and Multi-Agent Systems & the Future. In *Proceedings of the International Lisp Conference*, New York, New York, October 2003.

[Cra88] I.D. Craig. Blackboard systems. *Artificial Intelligence Review*, 2(2):103–118, 1988.

[CRMM09] V.M. Catterson, S.E. Rudd, S.D.J. McArthur, and G. Moss. On-line transformer condition monitoring through diagnostics and anomaly detection. In *Intelligent System Applications to Power Systems, 2009. ISAP '09. 15th International Conference on*, pages 1–6, Nov. 2009.

[CS07] D.A. Cartes and S.K. Srivastava. Agent applications and their future in the power industry. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1 –6, 2007.

[Dav96] M. W. Davies. Graphite Core Design in UK Reactors. In *IAEA-TECDOC-901*, pages 47–56. IAEA, 1996.

[Dep08] Department of Business, Enterprise and Regulatory Reform. Digest of United Kingdom Energy Statistics: 2008, 2008.

[EGS+08] J.J. Erbrink, E. Gulski, J.J. Smit, P.P. Seitz, and R. Leich. Experimental model of aging mechanisms of on-load tap changer contacts. In *Condition Monitoring and Diagnosis, 2008. CMD 2008. International Conference on*, pages 247 –250, 2008.

[FBB+02] A.F. Fernandez, B. Brichard, P. Borgermans, F. Berghmans, M. Decreton, P. Megret, M. Blondel, and A. Delchambre. Fibre bragg grating temperature sensors for harsh nuclear environments. In *Optical Fiber Sensors Conference Technical Digest, 2002. OFS 2002, 15th*, pages 63–66 vol.1, 2002.

[fip10] FIPA Publicly Available Agent Platform Implementations, 2010. `http://www.fipa.org/resources/livesystems.html`.

[Fou02a] Foundation for Intelligent Physical Agents. FIPA Abstract Architecture, 2002. Available from http://fipa.org/specs/fipa00001.

[Fou02b] Foundation for Intelligent Physical Agents. FIPA ACL Message Structure Specification, 2002. Available from http://fipa.org/specs/fipa00061.

[Fou02c] Foundation for Intelligent Physical Agents. FIPA Agent Message Transport Service, December 2002.

[Fou02d] Foundation for Intelligent Physical Agents. FIPA Communicative Act Library Specification, December 2002.

[Fou02e] Foundation for Intelligent Physical Agents. FIPA Query Interaction Protocol Specification, 2002. Available from http://fipa.org/specs/fipa00027.

[Fou02f] Foundation for Intelligent Physical Agents. FIPA Request Interaction Protocol Specification, 2002. Available from http://fipa.org/specs/fipa00026.

[Fou04] Foundation for Intelligent Physical Agents. FIPA Agent Management Specification, 2004. Available from http://fipa.org/specs/fipa00023.

[FW07] Charles R Farrar and Keith Worden. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):303–315, 2007.

[Gee08] P. Geering. Underwriting the Weight Loss Limit for Graphite in AGR Power Stations. In *Securing the Safe Performance of Graphite Reactor Cores Conference Proceedings*. Royal Society of Chemistry, 2008.

[GPS00] J.A. Giampapa, M. Paolucci, and K. Sycara. Agent interoperation across multiagent system boundaries. In *Proceedings of the fourth international conference on Autonomous agents*, page 186. ACM, 2000.

[GZJB02] V. Giurgiutiu, A. Zagrai, and J. Jing Bao. Piezoelectric wafer embedded active sensors for aging aircraft structural health monitoring. *Structural Health Monitoring*, 1(1):41, 2002.

[Hea07] Health and Safety Executive. The licensing of nuclear installations, 2007.

[hib09] Hibernate Website, 2009. `http://www.hibernate.org`.

[How06a] How a PWR power station works, 2006. `http://www.british-energy.com/opendocument.php?did=528`.

[How06b] How an AGR power station works, 2006. `http://www.british-energy.com/opendocument.php?did=529`.

[Hun96] Trevor M. Hunt. *Condition Monitoring of Mechanical and Hydraulic Plant*, chapter 7, pages 73–87. Chapman and Hall, 1996.

[Int06] International Atomic Energy Authority. Nuclear power reactors in the world, April 2006.

[jac10] JACK, 2010. `http://www.agent-software.com.au/products/jack/`.

[jad09] The JADE BeanGenerator, 2009. `http://protegewiki.stanford.edu/index.php/OntologyBeanGenerator`.

[jad10] JADE - Java Agent DEvelopment Framework, 2010. `http://jade.tilab.com/`.

[Jen01] N.R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

[JLB06] A.K.S. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483–1510, 2006.

[JM08] Gordon J. Jahn and Stephen D. J. McArthur. Imaps – a system for managing graphite core information. In *Securing the Safe Performance of Graphite Reactor Cores Conference Proceedings*. Royal Society of Chemistry, 2008.

[JMF99] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[JMR05] Gordon J. Jahn, Stephen D. J. McArthur, and J. Reed. An integrated architecture for graphite core data analysis. In *Ageing Management of Graphite Reactor Cores Conference Proceedings*, pages 224–231. Royal Society of Chemistry, 2005.

[JMRT07] G. Jahn, S. McArthur, J. Reed, and D. Towle. Staged Implementation of an Agent Based Advanced Gas-Cooled Reactor Condition Monitoring System. In *IEEE Power Engineering Society General Meeting, 2007*, pages 1–4, 2007.

[Jon90] G.R. Jones. Power switchgear monitoring with optical fiber systems. In *Developments Towards Complete Monitoring and In-Service Testing of Transmission and Distribution Plant, IEE Colloquium on*, pages 3/1–3/2, Oct 1990.

[Jon05] Chris Jones. Predicting the stresses and deformations of irradiated graphite. In *Proceedings of the Ageing Management of Graphite Reactor Cores 2005*, Serco Assurance, 2005.

[Les04] R. Leszczyna. Evaluation of agent platforms. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, 2004.

[LL72] R. Lind and H. G. Lewis. Graphite corrosion in advanced gas-cooled reactors. In *Graphite Structures for Nuclear Reactors*, pages 59–72, March 1972.

[LOZ+06] H. Li, J. Ou, X. Zhao, W. Zhou, H. Li, Z. Zhou, and Y. Yang. Structural health monitoring system for the shandong binzhou yellow river highway bridge. *Computer-aided civil and infrastructure engineering*, 21(4):306–317, 2006.

[Mah36] P.C. Mahalanobis. On the generalized distance in statistics. In *Proceedings of the National Institute of Science, Calcutta*, volume 12, page 49, 1936.

[MCdPF10] E.A. Mackenzie, J. Crossey, A. de Pablo, and W. Ferguson. On-line monitoring and diagnostics for power transformers. In *Electrical Insulation (ISEI), Conference Record of the 2010 IEEE International Symposium on*, pages 1 –5, 2010.

[McM07] Alan W. McMorran. *An Introduction to IEC 61970-301 & 61968-11: The Common Information Model*. University of Strathclyde, 2007.

[MD06] S.D.J. McArthur and E.M. Davidson. Automated post-fault diagnosis of power system disturbances. In *Power Engineering Society General Meeting, 2006. IEEE*, pages 6 pp.–, 0-0 2006.

[MDC+07] S.D.J. McArthur, E.M. Davidson, V.M. Catterson, A.L. Dimeas, N.D. Hatziargyriou, F. Ponci, and T. Funabashi. Multi-agent systems for power engineering applicationsÑpart i: Concepts, approaches, and technical challenges. *Power Systems, IEEE Transactions on*, 22(4):1743–1752, Nov. 2007.

[MKB+02] T.E. Munns, R.M. Kent, A. Bartolini, CB Gause, JW Borinski, J. Dietz, JL Elster, C. Boyd, L. Vicari, K. Cooper, et al. Health monitoring for airframe structural characterization. *NASA Contractor Report*, 211428, 2002.

[MMM01] E.E. Mangina, S.D.J. McArthur, and J.R. McDonald. The use of a multi-agent paradigm in electrical plant condition monitoring. In *Power Engineering, 2001. LESCOPE '01. 2001 Large Engineering Systems Conference on*, pages 31–36, 2001.

[MRM96] N. McLachlan, J. Reed, and M.P. Metcalfe. AGR Core Safety Assessment Methodologies. In *IAEA-TECDOC-901*, pages 125–36. IAEA, 1996.

[NJT87] D.J. Norfolk, G.O. Johnson, and M.O. Tucker. Achieving Optimum Graphite Performance in AGR Core and Fuel. In *IAEA Specialists Meeting on Graphite Component Structural Design, September 8-11, 1986*, pages 176–81. IAEA, 1987.

[NOE+98] D. Naus, CB Oland, B. Ellingwood, WE Norris, and HL Graves III. Management of aging of nuclear power plant

containment structures. In *4. international conference on engineering structural integrity assessment, Cambridge (United Kingdom), 22-24 Sep 1998*, 1998.

[Oak07] Oakland Tribune. An a-maze-ing response to MacArthur mess. Newspaper, 10 May 2007.

[OBM04] P. Omenzetter, J.M.W. Brownjohn, and P. Moyo. Identification of unusual events in multi-channel bridge monitoring data. *Mechanical Systems and Signal Processing*, 18(2):409–430, 2004.

[OBM06] P. Omenzetter, W. Brownjohn, and J. Mark. Application of time series analysis for bridge monitoring. *Smart Materials and Structures*, 15:129–138, 2006.

[Ped98] Tor Pedersen. BWR 90–the advanced BWR of the 1990s. *Nuclear Engineering and Design*, 180(1):53 – 66, 1998.

[PMG+09] I.E. Portugues, P.J. Moore, I.A. Glover, C. Johnstone, R.H. McKosky, M.B. Goff, and L. van der Zel. RF-based partial discharge early warning system for air-insulated substations. *Power Delivery, IEEE Transactions on*, 24(1):20 –29, 2009.

[Pos10] PostgreSQL Global Development Group. PostgreSQL 8.4.2 Documentation, Section 5.8: Inheritance, February 2010.

[RCJ09] A. Rogers, D.D. Corkill, and N.R. Jennings. Agent technologies for sensor networks. *Intelligent Systems, IEEE*, 24(2):13–17, March-April 2009.

[Ren08] Renewables Advisory Board. 2020 VISION – How the UK can meet its target of 15% renewable energy. Technical report, Renewables Advisory Board, 2008.

[Rey96] W.N. Reynolds. *Chemistry and Physics of Carbon*, volume 2, chapter Radiation Damage in Graphite, pages 170–6. Edward Arnold (Publishers), Ltd, London, 1996.

[RG91] A.S. Rao and M.P. Georgeff. Modeling Rational Agents within a BDI-Architecture. 1991.

[RN03] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*, chapter 2, pages 32–44. Prentice-Hall Inc., second edition, 2003.

[RTV06] Timothy Redmond, Tania Tudorache, and Jennifer Vendetti. Tutorial on application development with Protégé, Part I. In *9th International Protégé Conference*, July 2006.

[SF93] M.J. Shaw and M.S. Fox. Distributed artificial intelligence for group decision support. *Decision Support Systems*, 9(4):349–367, 1993.

[SFHC02] H. Sohn, CR Farrar, FM Hemez, and JJ Czarnecki. A Review of Structural Health Review of Structural Health Monitoring Literature 1996-2001. Technical report, Los Alamos National Laboratory, 2002.

[SJ04] E. Shakshuki and Y. Jun. Multi-agent development toolkits: An evaluation. *Lecture Notes in Computer Science*, pages 209–218, 2004.

[SJMM03] S. M. Strachan, G. Jahn, S. D. J. McArthur, and J. R. McDonald. Intelligent Diagnosis of Defects Responsible for Partial Discharge activity Detected in Power Transformers. In *Intelligent System Applications in Power Systems Conference (ISAP2003)*, page ISAP03/109, 2003.

[SMC+10] Cheng Shouyu, Peng Minjun, Gong Cheng, Zhao Qiang, Tian Zhaofei, Zhu Haishan, and Liu Zhongkun. Research on digital control system simulation for nuclear power plants. In *Power and Energy Engineering Conference (APPEEC), 2010 Asia-Pacific*, pages 1 –4, 28-31 2010.

[SRG03] R.E.H. Sims, H.H. Rogner, and K. Gregory. Carbon emission and mitigation cost comparisons between fossil fuel, nuclear and renewable energy resources for electricity generation. *Energy Policy*, 31:1315–1326, 2003.

[SSM+07] B. Stephen, SM Strachan, SDJ McArthur, JR McDonald, and K. Hamilton. Design of trip current monitoring system for circuit

breaker condition assessment. *Generation, Transmission & Distribution, IET*, 1(1):89–95, 2007.

[Ste05] Alan G. Steer. AGR Core Design, Operation and Safety Functions. In *Ageing Management of Graphite Reactor Cores Conference Proceedings*, pages 11–18. Royal Society of Chemistry, 2005.

[SWG+09] B. Stephen, G.M. West, S. Galloway, S.D.J. McArthur, J.R. McDonald, and D. Towle. The use of hidden markov models for anomaly detection in nuclear core condition monitoring. *Nuclear Science, IEEE Transactions on*, 56(2):453 –461, apr. 2009.

[Tra06] Trade and Industry Committee. New nuclear? examining the issues, fourth report of session 2005-06 volume i. Technical report, House of Commons, UK Government, 2006.

[try10] Tryllian Agent Development Kit, 2010. `http://www.tryllian.org/`.

[TSB+02] S. Tenbohlen, T. Stirl, G. Bastos, J. Baldauf, P. Mayer, M. Stach, B. Breitenbauch, and R. Huber. Experienced-based Evaluation of Economic Benefits of on-line Monitoring Systems for Power Transformers. *CIGRE Session 2002*, pages 12–110, 2002.

[UT03] R.E. Uhrig and L.H. Tsoukalas. Multi-agent-based anticipatory control for enhancing the safety and performance of Generation-IV nuclear power plants during long-term semi-autonomous operation. *Progress in Nuclear Energy*, 43(1-4):113–120, 2003.

[Var02] A. Varl. On-line diagnostics of oil-filled transformers. In *Dielectric Liquids, 2002. ICDL 2002. Proceedings of 2002 IEEE 14th International Conference on*, pages 253–257, 2002.

[War63] Jr. Ward, Joe H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):pp. 236–244, 1963.

[WdJ92] R. Wieringa and W. de Jonge. The identification of objects and roles. *Faculty of Mathematics and Computer Science, Vrije Universiteit (working manuscript)*, 1992.

[WH72] D. F. Willson and J. M. Harte. Practical experience in the design and erection of the Dungeness 'B' core. In *Graphite Structures for Nuclear Reactors*, pages 147–166, March 1972.

[WJ95] Michael Wooldridge and N. R. Jennings. Intelligent agents: Theory and Practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[WJM11] Christopher J. Wallace, Gordon J. Jahn, and Stephen D.J. McArthur. Multi-Agent System For Nuclear Condition Monitoring. In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, May 2011.

[WLJL10] Hairui Wang, Ya Li, Lin Jin, and Yuping Liu. Multi-agents based fault diagnosis systems in msw incineration process. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, volume 2, pages 721 –724, 13-14 2010.

[Wor09] World Nuclear Association. Nuclear Power Reactors, 2009.

[WW96] H. Wang and C. Wang. APACS: a multi-agent system with repository support. *Knowledge-Based Systems*, 9(5):329–337, 1996.

[Yeo80] R.M. Yeomans. Advanced gas-cooled reactors (AGR). In *Specialists' Meeting on Gas-cooled Reactor Safety and Licensing Aspects*, pages 27–38. International Atomic Energy Authority, September 1980.

[YXQ01] Cunzhen Yu, Jie Xiao, and Yuehui Qin. Analysis of the superheated steam cycle of a PWR secondary system. *Heat Transfer – Asian Research*, 30(3):185–194, 2001.