THE UNIVERSITY *of* LIVERPOOL

INTELLIGENT INFORMATION RETRIEVAL AND FAULT DIAGNOSIS FOR THE ASSET MANAGEMENT OF POWER SUBSTATIONS

Thesis submitted in accordance with the
requirements of the University of Liverpool
for the degree of Doctor in Philosophy

in

Department of Electrical Engineering and Electronics

by

Zhen YANG, B.Sc. (Eng.)

December 2008

INTELLIGENT INFORMATION RETRIEVAL AND FAULT DIAGNOSIS FOR THE ASSET MANAGEMENT OF POWER SUBSTATIONS

by

Zhen YANG

To my families

Acknowledgements

First of all, I would like to thank my supervisor, Professor Q.H. Wu, not only for his invaluable support and intellectual guidance, but also for providing me with such a great opportunity to enrich myself on both academic and personal levels. He has made contributions to this research project, of which I am truly appreciative.

I am deeply grateful to Dr. W.H. Tang for many helpful discussions along the way. Moreover, my thanks goes to all the members of the Intelligence Engineering and Automation Group for their kind help during my studies. In particular, I would like to mention Mr. C. Ma, Dr. A. Shintemirov, Dr. W.J. Tang and Dr. Z. Lu for their valuable comments and ideas on experimental work.

I would also like to offer my thanks to the Department of Electrical Engineering and Electronics in the University of Liverpool, for providing the research facilities that made it possible for me to conduct this research project.

Last but not least, I am greatly indebted to my family and my fiancee, for their understanding, encouragement, patience and love throughout the period of my postgraduate life.

Abstract

This thesis mainly presents two intelligent approaches to the Asset Management (AM) of power substations, which include an Evidential Reasoning (ER)-based document ranking approach to an Ontology-based Document Search Engine (ODSE) for the Information Retrieval (IR) of power substations and an Association Rule Mining (ARM)-based Dissolved Gas Analysis (DGA) approach to the Fault Diagnosis (FD) of power transformers.

In an ODSE, an ontology model is used for expanding a submitted query with its relevant terms extracted from the ontology model. The ODSE then retrieves useful information from a document repository according to the expanded query. To develop the ER-based document ranking approach, a domain ontology model, used for Query Expansion (QE), and its connection with an ODSE are designed. A Multiple Attribute Decision Making (MADM) tree model is proposed to organise the terms of an expanded query for ranking purposes. An ER algorithm, based on the Dempster-Shafer (DS) theory, is used for evidence combination in the MADM tree model. In this thesis, the proposed approach is discussed in a generic frame for document ranking, in link with a query, namely fault diagnosis, as an example and evaluated using a number of queries commonly used in power substation document retrieval. The results show that the proposed approach provides a suitable solution to document ranking and the search accuracy of an ODSE has been significantly improved with the ER approach embedded, in comparison with a traditional keyword-matching search engine, and an ODSE without ER.

In the development of the ARM-based DGA approach, an attribute selection method and a continuous datum attribute discretisation method are used for choosing user-interested ARM attributes from a provided DGA data set, *i.e.*, the items that can be employed for the generation of association rules. The given DGA data set is composed of two parts, *i.e.*, training and test DGA data sets. An ARM algorithm namely Apriori-Total From Partial (TFP) is proposed for generating an Association Rule Set (ARS) from the training DGA data set. Afterwards, a rule set simplification method and a rule fitness evaluation method are utilised to select useful rules from the ARS and assign a fitness value to each of the useful rules, respectively. Based upon the useful association rules, a transformer FD system is developed, in which an optimal rule selection method is employed for selecting the most accurate rule from the system for diagnosing a given test DGA record. Test results demonstrate that, with the same training and test DGA data sets, a higher FD accuracy has been achieved with the association rule-based FD system, compared with that derived by a set of conventional FD techniques.

In order to efficiently reuse a useful ARS in different Rule-Based Expert Systems (RBESs) of transformer FD, a Semantic Web Rule Language (SWRL) is chosen for interpreting a useful ARS as a SWRL rule base, which is capable of being processed by various RBESs developed with different rule execution engines. With the purpose of verifying the accuracy of the derived SWRL rule base for transformer FD, Java expert system shell (Jess) is used for establishing a SWRL RBES (SRBES) with the SWRL rule base. With the same test DGA data set mentioned above, the same FD accuracy, compared with that of the association rule-based FD system, has been obtained by the SRBES, which illustrates the capability of the SWRL rule base for transformer FD.

An Agent-based AM System (AAMS) of power substations, with the ER-based ODSE and the association rule-based FD system embedded, then is developed. The system structure of AAMS is introduced. Meanwhile, the functions of the AAMS components as well as the AM services provided by AAMS are discussed in detail. The practical performance of AAMS is evaluated with a set of designed experiments. The results illustrate that AAMS can fulfill the requirements of substation AM for both the IR and FD aspects.

Contents

List of Figures

List of Tables

List of Abbreviations

AAMS	Agent-based Asset Management System
ACL	Agent Communication Language
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
AID	Agent Identification
AM	Asset Management
AMS	Agent Management System
ANN	Artificial Neural Network
API	Application Programming Interface
Apriori-T	Apriori-Total
Apriori-TFP	Apriori-Total From Partial
ARM	Association Rule Mining
ARS	Association Rule Set
BDI	Belief-Desire-Intention
C	Centigrade
CPU	Central Processing Unit
C_2H_2	Acetylene
C_2H_4	Ethylene
C_2H_6	Ethene
CH_4	Methane
CO	Carbon monoxide
CO_2	Carbon dioxide
DAA	Data Acquisition Agent
DAGSVM	Directed Acyclic Graph Support

	Vector Machine
DAML+OIL	DARPA Agent Markup Language+ Ontology Inference Layer
DF	Directory Facilitator
DGA	Dissolved Gas Analysis
DIA	Data Interpretation Agent
DN	Document Network
DS	Dempster-Shafer
DSA	Document Search Agent
EA	Evolutionary Algorithm
ECG	Electrocardiogram
ER	Evidential Reasoning
ES	Expert System
EU	European Union
EVSM	Extended Vector Space Model
FD	Fault Diagnosis
FDA	Fault Diagnosis Agent
FIPA specification	Foundation for Intelligent Physical Agent specification
FL	Fuzzy Logic
FN	False Negatives
FOL	First Order Logic
FP	False Positives
FST	Fuzzy Set Theory
GB	Gigabyte
GHz	Gigahertz
GUI	Graphical User Interface
H_2	Hydrogen
IA	Index Agent
ID3	Dichotomiser 3
IEEE	The Institute of Electrical and Electronics

	Engineers
IN	Inference Network
IR	Information Retrieval
JADE	Java Agent DEvelopment Framework
JDK	Java Development Kit
Jess	Java expert system shell
JRE	Java Runtime Environment
KNN	K-Nearest Neighbour
LAN	Local Area Network
LUCS-KDD	Liverpool University Computer Science-Knowledge Discovery in Data
MADM	Multiple Attribute Decision Making
MAS	Multi-Agent System
MB	Megabyte
MLP	Multi-Layer Perception
MS	Management Science
NG	National Grid, U.K.
NIST	National Institute of Standards and Technology
ODSE	Ontology-based Document Search Engine
OWL	Web Ontology Language
OWL DL	Web Ontology Language Description Logic
PD	Partial Discharge
ppm	parts per million
PSR	Power System Restoration
P-tree	Partial support tree
QA	Query Agent
QE	Query Expansion
QN	Query Network
RAM	Random-Access Memory
R1	Ratio 1=CH_4/H_2
R2	Ratio 2=C_2H_2/C_2H_4

R3	Ratio 3=C_2H_2/CH_4
R4	Ratio 4=C_2H_6/C_2H_2
R5	Ratio 5=C_2H_4/C_2H_6
RBA	Rule Base Agent
RBES	Rule-Based Expert System
RBF	Radial Basis Function
RDF	Resource Description Framework
RDF Schema	Resource Description Framework Schema
RS	Relevance Score
RuleML	Rule Markup Language
SONT	Substation ONTology
SRBES	Semantic web rule language Rule-Based Expert System
SVM	Support Vector Machine
SWRL	Semantic Web Rule Language
tf-idf	term frequency-inverse document frequency
TN	True Negatives
TP	True Positives
TREC	Text REtrieval Conference
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XML Schema	Extensible Markup Language Schema

Chapter 1

Introduction and Overview

1.1 Background of asset management for power systems

As defined in [3], Asset Management (AM) is the combination of management, finance, economy, engineering and other practices applied to physical assets, with the objective of providing the required level of service in the most cost-effective manner. In [4], a broader view of AM is given: whereby an asset is defined as any item or property owned by an individual or business which has monetary value. Three types of assets were identified: physical assets (*e.g.*, apparatuses), financial assets (*e.g.*, financial instruments and equity accounted investments) and intangible assets (*e.g.*, operating licences, knowledge and the skills of staff). That is to say, a complete AM system must concern all the three types of assets mentioned above. At the same time, other definitions of AM have also been denoted. Although different viewpoints of AM are stated by these definitions, certain concerns that are in common can be extracted and concluded as: an AM system must meet the goals of an organisation in both financial and physical parts, in order to improve the values of the organisation as well as the benefits of its shareholders.

Electrical apparatuses are usually capital intensive, robust, long-lived and not easily relocatable. Thus, in power transmission networks, apparatuses liv-

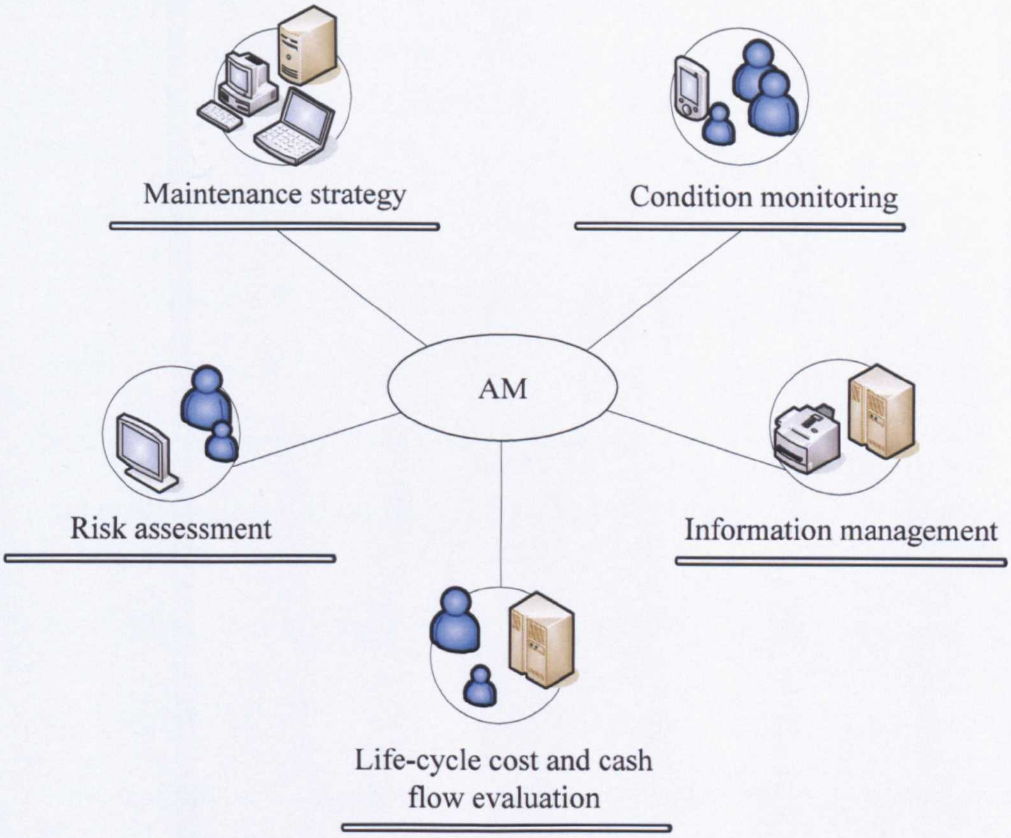


Figure 1.1: Asset management in a power system

ing to 40 and 50 years are not uncommon and some apparatuses even have been operated for up to 80 years, which have been causing high risks for the operations of the apparatuses. With the increasing pressure from both massive industrial growth and capital expenditure, power systems have recently been facing unprecedented strains on asset organisation and utilisation. In order to tackle the problems, AM now is always on the minds of many system operators in electric power industry and many efforts have been made to implement AM into power systems [5] [6] [7] [8]. In these AM projects regarding power systems, the main purpose of an AM programme is to manage physical assets and their associated performance optimally. The balance is achieved by integrating technical diagnoses and management decisions. According to the research work reported in these AM projects, the main elements of a typical power AM system can be summarised as in Figure 1.1. As can be seen from the figure, there are five main elements involved in a power AM system:

- Proactive maintenance with optimised repairing strategies.
- Condition monitoring and Fault Diagnosis (FD).
- Risk evaluation of system operations.
- Information management and knowledge representation.
- Life-cycle cost analysis and cash flow prediction.

1.2 Asset management aspects concerned in this thesis

With respect to the basic elements of a power AM system mentioned in Section 1.1, an Agent-based Asset Management System (AAMS) has been cooperatively developed for AM of power systems in our research group (the *e*-Automation laboratory at the University of Liverpool), based on the previous research of agent-based power system automation and condition monitoring [9].

In the development process of AAMS, the structure of AAMS was designed by the thesis author firstly. Then, two intelligent AM approaches were developed by the thesis author, which include an Evidential Reasoning (ER) [10] [11]-based document ranking approach to an Ontology-based Document Search Engine (ODSE) for the Information Retrieval (IR) of power substations and an Association Rule Mining (ARM) [12]-based Dissolved Gas Analysis (DGA) [1] approach to FD of power substations. Subsequently, according to the AAMS structure, the above two AM approaches were embedded into AAMS by the other researcher.

A power substation is one of the most vital apparatuses in a power system. Nowadays, a large number of digitally stored power substation documents, which are congregated in power company networks and databases, is causing considerable difficulties in IR and thus results in reduced efficiency of substation information reuse. In current power enterprises, a document search engine is usually recognised as an efficient tool for IR. However, several drawbacks still exist in such a system, as analysed in Section 1.4.1. Consequently, the accuracy of an IR process, *i.e.,* the proportion of user-interested documents out of all the documents discovered during the IR process, can be greatly reduced. Therefore in the first part of this thesis, an ER-based ODSE is introduced for accurately retrieving user-interested information of power substations.

Meanwhile, the aging components that are in service of a substation are faced with increasing unreliability, which raises potential risks of the substation. In a substation, a number of power apparatuses are involved, *e.g.,* power transformers, circuit breakers, high and low voltage switchgears and so on. Thus, FD of each apparatus can be treated as a type of substation FD work. Being served as the most expensive and the most important apparatus of a power substation, the working state of a power transformer is always treated as a vital factor that influences the safety of the substation. A failure of a transformer may cause a serious outage of a power substation, which can further greatly reduce the reliability and the final power quality of a power system. At the same time, the high cost of a system restoration task can greatly impact the budget of a power enterprise. Therefore, in the past few years, FD of power transformers has been widely studied for monitoring a substation's condition. Nevertheless, the deficiencies of traditional transformer FD methods, *e.g.,* Rogers, Dornenburg, Key Gas methods [1], still exist, as described in Section 1.4.2. As a result, the accuracy of a transformer FD process can be greatly reduced. Thus, in the second part of the thesis, a novel association rule-based FD system for DGA of power transformers is presented, with which the working state of a transformer can be diagnosed with a high accuracy.

Subsequently, the implementation process of the above two AM approaches in AAMS is illustrated. In the next subsection, a brief introduction to conventional AM techniques related to the research areas of the thesis, *i.e.,* IR of power substations and FD of power transformers, is provided.

1.3 Review of related areas in power substation asset management

1.3.1 Information retrieval of power substations

In a power enterprise, digitally stored information of power substations, *e.g.*, technical reports regarding substations, plenty of backup documents of substations and a large number of substation maintenance records and so on, are increasing rapidly in volume. As a result, the difficulty of retrieving useful documents regarding power substations has been growing. Therefore, one of the most pressing technological and commercial requirements of a power enterprise is an IR tool, which retrieves user-interested information of substations quickly and accurately.

Currently, although a manual database retrieval method is still under employment, the dominant paradigm for addressing this requirement is an automatic IR [13] solution, which is normally represented as a document search engine. The main objective of a search engine is to discover the information in relation to a user's query input. When a query is submitted, which is typically in the format of several keywords, a search engine retrieves documents using the query and then a list of matched documents is ranked based upon their assigned relevance scores to the query and returned to the user.

Normally, a well formatted query can explicitly illustrate the desired information of a user's interest and thus leads to a high search accuracy. However, practical users usually do not have any knowledge about search engines, and queries submitted are often presented unclearly and incompletely. As a result, the search accuracy of a search engine may be greatly restricted. In order to overcome this problem, Query Expansion (QE) has been suggested as a viable solution. Generally, QE presents a process of augmenting a query input with its related terms. With an expanded query, the documents which do not contain the same keywords of the original query but are correlative to the inferred terms, can be retrieved. Consequently, the search scale in a search process is suitably broadened and a more accurate result may be obtained by retrieving more relevant documents.

Previously, a number of QE techniques have been proposed, which are mainly developed upon mechanisms of relevance feedback [14] [15] and statistical term co-occurrence [16] [17] and so on. In the tests regarding these techniques, an improved search accuracy has been achieved with an expanded query in most cases, compared with that obtained with an unexpanded query. However, a significant drawback of the relevance feedback and statistical term co-occurrence-based QE approaches is that, the related terms of a query input are obtained by analysing the context of documents from a document repository. Thus, the relatedness between related terms and an original query cannot be ensured, if there are insufficient documents used for analysing before a search process.

Hence more recently, a new method which employs a corpus independent knowledge model, *i.e.*, an ontology model, for QE has been suggested to deal with this issue. An ontology model is "a classification, thesaurus or a set of concept clusters" [18]. Ontologies used for QE can be either a general-purpose ontology model, or a domain ontology model, developed regarding the context of a specific domain. The employment of a general purpose ontology model, *e.g.*, WordNet [19], for QE can be dated back to the early 1990s. More recently, the use of domain ontologies for QE has been widely studied. Compared with a general-purpose ontology model, the terminology in a domain ontology model is less ambiguous, therefore a query can be expanded with terms that are more relevant to itself [20]. As a result, a higher IR accuracy may be achieved during a search process with the expanded query, compared with that obtained using an expanded query derived with a general-purpose ontology model. The implementation results of recent ontology-based QE research, *e.g.*, [21] [22] [23] and [24] and so on, are illustrated in detail in Section 2.2.2.

1.3.2 Fault diagnosis of power transformers

Practically, major power transformers of a power substation are normally filled with mineral oil. When a transformer is working at normal conditions, there is usually a slow consumption of the filled mineral oil, which produces certain gases of low concentrations. However, the gas concentrations increase at a much more rapid rate when an electrical fault occurs within the transformer. Therefore, many efforts have been dedicated to the incipient fault detection of power transformers, out of which DGA is probably the most widely used technique. In the past three decades, DGA has been considered useful and reliable for FD of not only oil-filled transformers, but also some other oil-filled electrical apparatuses, *e.g.*, circuit breakers and switchgears and so on.

Conventionally, various criteria of DGA have been developed, such as Rogers, Dornenburg, Key Gas methods. In these methods, the FD criteria are mainly established by analysing the concentrations or the mutual comparison ratios of a set of key gases, *e.g.*, Hydrogen (H_2), Methane (CH_4), Ethene (C_2H_6), Ethylene (C_2H_4), Acetylene (C_2H_2), Carbon monoxide (CO) and Carbon dioxide (CO_2) and so on. In a transformer FD process, by applying these DGA interpretation techniques on the insulation oil samples of the transformer, firstly, the key dissolved gas concentrations or ratios are quantitatively determined. Then, transformer faults, mainly in the forms of thermal, arcing and Partial Discharge (PD), can be detected by analysing the obtained dissolved gas concentrations or ratios. Such an analysis process can indicate one or more possible fault states of a transformer and thus allows for the time to take necessary preventive measures.

On the other hand, in recent years, many attempts also have been made to employ Artificial Intelligence (AI) techniques for DGA, *e.g.*, Expert System (ES) [25], Fuzzy Logic (FL) [26] [27], Artificial Neural Network (ANN) [28] [29], Support Vector Machine (SVM) [30] [31] and K-Nearest Neighbour (KNN) [32] [33] and so on. These AI techniques are introduced detailedly in Section 4.2.2.

1.4 Motivation and objectives

1.4.1 Drawbacks of traditional ontology-based document search engines

As discussed in Section 1.3.1, the accuracy of a search engine may be enhanced with an ontology-based QE method. However, drawbacks still exist in ODSEs. Firstly, in most of the previous research, much attention has been focused on the algorithms of discovering the best expansion terms and the number of expansion terms on trying to achieve the best search accuracy, rather than proposing a more reasonable mechanism for the organisation of the terms existed in an expanded query. The related terms, *e.g.*, synonyms and hyponyms (subclasses), are added to an original query without considering their hierarchical relationships. In such a situation, although mutual weights are assigned differently to a pair of terms in an expanded query, the emphasis of the original query could be biased in a search process. Secondly, various methods are used to determine the relevance scores between the terms of an expanded query and a document. However, the relatedness between the expanded query and the document is normally calculated by the weighted sum of these generated relevance scores, based upon Vector Space Model (VSM) [34] [35]. In such cases, the relevance scores generated by the query terms are treated independently during the combination process, which may reduce the accuracy of a final search result.

1.4.2 Bottlenecks of conventional dissolved gas analysis methods

In a transformer FD process with a DGA criterion, various fault conditions and other interfering factors may be involved. Therefore, the relationship between a specific fault type and dissolved gas concentrations is difficult to be determined. Based upon empirical studies by power engineers, classification rules have been established in the conventional DGA methods, *e.g.*, the Rogers, Dornenburg and Key Gas methods. However, a vital limitation of the rules included in such a conventional DGA method is that in some cases, a set of measured gas concentrations or ratios may not fit within the predefined criteria in the conventional DGA method. As a result, faults that occur inside transformers are not identifiable. On the other hand, in practice, different DGA methods often produce different judgements when processing the same dissolved gas record [36]. Consequently, power engineers are forced to use several DGA methods and other related information about a transformer together, *e.g.*, the previous operation history of the transformer, the results of the latest inspection, the states of on-load tap changer [37] and so on, to assess the working state of the transformer, which obviously is not a convenient solution.

1.4.3 Objectives of this research

This thesis presents the development of two intelligent substation AM approaches for dealing with the problems mentioned above.

1. The original investigation of an ER-based document ranking approach is presented to cope with the problems that exist in an ODSE. ER is a decision making algorithm, which is specifically used to combine multiple probability assignments generated from a range of evidence considering nonlinearity and uncertainties. The basic idea of the proposed approach is that the expansion terms are considered as auxiliary evidence of an original query term. All the terms from an expanded query are organised as a Multiple Attribute Decision Making (MADM) tree model of multi-levels, regarding their hierarchical relationships defined in an employed ontology model. Moreover, different relative weights are assigned to the terms situated in the same level. The relevance scores generated between these query terms and a document are then combined with the proposed ER algorithm. Finally, a list of relevant documents is returned to users as a search result, in which the documents are ranked in the descending order of their relevance scores concerning an expanded query.

2. In order to deal with the problems that exist in conventional DGA methods, in this study, a novel ARM-based DGA approach, is presented for FD of power transformers. ARM is a data mining technique and was first introduced in [12]. In this study, it is the very first time that the ARM technique is proposed for FD of power transformers. The basic idea of implementing ARM in DGA is to generate association relationships between a set of key gas values and transformer working states, *i.e.*, various fault classes or no fault, from real DGA records. The derived association relationships are then interpreted as a set of raw association rules. Subsequently, a large number of useful association rules can be extracted from the rule set and used for FD of power transformers with a high accuracy.

The process of implementing the above two substation AM approaches in AAMS then is illustrated as well.

1.5 Thesis outline

The rest of the thesis is structured as follows: Chapter 2 presents a novel approach to document ranking in an ODSE using the ER algorithm. In Chapter 3, the experimental results of the ER-based document ranking approach are reported. An ARM-based DGA approach to FD of power transformers then is depicted in Chapter 4 and the corresponding evaluation results of the approach are described in Chapter 5. Chapter 6 introduces the development and evaluation processes of a Semantic Web Rule Language (SWRL) rule base, derived from a set of useful association rule generated in Chapter 5. Subsequently, the implementation processes of an ER-based ODSE and an association rule-based FD system in AAMS is illustrated in Chapter 7. Chapter 8 concludes this thesis by giving a summary of the results obtained in the research. In addition, related research work that can be investigated in future is suggested.

1.6 Major contributions of this research

The major contributions arising from this thesis are summarised as follows:

- In the first part of this thesis, the original work on the development of an ODSE using ER is described. In the study, it is the first time that the terms of an expanded query are organised into a MADM tree model during a search process. The ER algorithm, based on the Dempster-Shafer (DS) [38] theory, is then proposed for the combination of the relevance scores generated between the terms of the expanded query and a document. Test results show that the proposed ER-based approach is a suitable solution for document ranking in an ODSE and the search accuracy of an ODSE has been improved, compared with that of an ODSE without the proposed ER-based approach.

  In this approach, a domain ontology model, *i.e.*, Substation ONTology (SONT), has been constructed in the context of a power substation and used for QE. Known as one of the most significant advantages of the Ontology technique, a well-defined ontology model can be reused by other context related ontology-based research. Therefore, it is possible to employ SONT to develop other projects concerning power systems and thus reduces expenses for rebuilding a new knowledge base.

  Most significantly, the ER-based document ranking approach used in an ODSE aims to generate an overall relatedness between an expanded query and a document. Therefore, it can be implemented with other QE techniques, *e.g.*, relevance feedback-based techniques and statistical co-occurrence-based techniques, without considering the way of how an expanded query is generated.

- The second part of the thesis describes the initial investigations of the ARM-based DGA approach which is made for FD of power transformers. The idea of using a set of generated association rules, instead of the empirical rules defined in a conventional DGA method, *e.g.*, Dornenburg or Rogers, for transformer FD is introduced in the study. Furthermore, the implementation process of a developed association rule-based FD system is illustrated in detail. Experiment results illustrate that the proposed ARM-based DGA approach is capable of generating a number of meaningful association rules which can cover the empirical rules used in a conventional DGA method, *e.g.*, Dornenburg or Rogers. Moreover, a higher FD has been achieved with an association rule-based FD system, compared with that derived by a set of conventional DGA methods. As a conclusion, the ARM-based DGA approach is a very helpful transformer FD approach to power operators.

Additionally, in this part of this thesis, the original research of interpreting a set of association rules as a SWRL rule base is reported. A SWRL rule base can be effectively executed by a set of published rule engines. As a result, a SWRL rule base can be reused by different Rule-Based Expert Systems (RBESs) and thus cuts down expenses for developing a new rule base for transformer FD. Moreover, a SRBES development process is clearly illustrated, which shows an implementation strategy of the SWRL rule base.

- In the third part of the thesis, AAMS, cooperatively developed with the other researcher using the proposed substation AM approaches, is presented. The advantages of AAMS, which are not possibly achieved by using a traditional substation AM system are clearly stated. Final test results show that AAMS can fulfill the requirements of substation AM for both IR and FD aspects.

1.7 Auto-bibliography

List of the publications produced from this work:

Conference papers:

1. **Z. Yang**, C. Ma, J.Q. Feng, Q.H. Wu, S. Mann and J. Fitch. A multi-agent framework for power system automation. In *Proc. Conf. on Artificial Intelligence in Energy Systems and Power*, Hotel Royal Savoy, Island of Madeira, Portugal, February 2006.

2. **Z. Yang**, Z. Lu, C. Ma, Q.H. Wu and J. Fitch. Improving control ability of relay protection system with intelligent agents. In *Proc. IEEE Int. Conf. on Power System Technology*, Chongqing, China, October 2006.

3. **Z. Yang**, W.H. Tang, C. Ma and Q.H. Wu. Fault diagnosis of power transformer with association rule mining. Submitted to *IEEE Asia-Pacific Power and Energy Conf.*, Wuhan, China, March 2009.

4. C. Ma, J.Q. Feng, **Z. Yang** and Q.H. Wu. Agent-based personal article citation assistant. In *Proc. IEEE Int. Joint Conf. on Web Intelligence and Intelligent Agent Technology*, pages 702-705, Compiegne University of Technology, France, September 2005.

5. C. Ma, J.Q. Feng, **Z. Yang**, Q.H. Wu and J. Fitch. A broker agent for remote control of distributed power systems. In *Proc. IEEE Int. Conf. on Power System Technology*, Chongqing, China, October 2006.

Journal papers:

1. **Z. Yang**, C. Ma, J.Q. Feng, Q.H. Wu, S. Mann and J. Fitch. A multi-agent framework for power system automation. *Int. J. of Innovations in Energy Systems and Power*, 1(1):39-45, November 2006.

2. **Z. Yang**, W.H. Tang and Q.H. Wu. Document ranking in ontology-based document search engine using evidential reasoning. Submitted to *IEEE Trans. on Knowledge and Data Engineering*, April 2007.

3. **Z. Yang**, W.H. Tang, A. Shintemirov and Q.H. Wu. Association rule mining-based dissolved gas analysis for fault diagnosis of power trans-

formers. Submitted to *IEEE Trans. on System, Man and Cybernetics, Part C: Applications and Reviews*, March 2008.

4. C. Ma, W.H. Tang, **Z. Yang**, Q.H. Wu and J. Fitch. Asset managing the power dilemma. *IET Control and Automation Magazine*, pages 40-45, October 2007.

5. C. Ma, **Z. Yang**, W.H. Tang and Q.H. Wu. An Agent-based transformer fault diagnosis system using association rule mining-based dissolved gas analysis. Ready to submit to *IEEE Trans. on Power Systems.*

Chapter 2

Document Ranking in An Ontology-Based Document Search Engine Using Evidential Reasoning

2.1 Introduction

In order to cope with the drawbacks of a traditional ODSE, which are analysed in Section 1.4.1, an ER-based document ranking approach is presented in this chapter. In a document search process of an ODSE using the ER-based document ranking approach, a domain ontology model, *i.e.,* SONT, is employed to extend an original query. Then, the relevance scores, which are generated between the terms of an expanded query and a specific document, are defined as a set of evidence within a MADM tree model. Consequently, the final output of this MADM tree model, which represents the overall relatedness between the expanded query and the document, can be evaluated based upon the combination of the available evidence using ER. Finally, a list of relevant documents is returned to users as a search result, in which the documents are ranked in the descending order according to their overall relatedness concerning the query.

The rest of this chapter is organised as follows: In Section 2.2, the history of IR and a set of published document ranking techniques of IR are introduced, respectively. The architecture of an ODSE and the component functions of the ODSE are described in Section 2.3. Section 2.4 gives a brief introduction to the Ontology technique firstly. Then, the development process of SONT, used for QE in this study, is described. In Section 2.5, the ER algorithm and the DS theory are briefly reviewed. Section 2.6 explains the methodology of transferring an input query into a MADM tree model in detail. In Section 2.7, the evidence combination process of the MADAM tree model using the ER algorithm is demonstrated. This chapter is summarised in Section 2.8.

2.2 Literature review of information retrieval

2.2.1 History of information retrieval

The practice of archiving information can be dated back to around 3000 BC, when Sumerians stored clay tablets into different areas, described with specific cuneiform inscriptions. Then, the need of storing and retrieving information became increasingly significant, with the broadening use of paper media, which made the size of written information growing rapidly. A very popular method for IR at that time was a collection of selected words, *i.e.*, index, which indicated the association relationships among related documents. As soon as computers were invented and implemented for file storages, the idea of using computers for IR was proposed. In 1945, the idea of automatic searching computer-stored information was initially introduced by Bush [39]. Subsequently, the term "IR" was coined by Mooers in 1948 [40] and has been widely used since 1950. In the mid 1950s, a number of research works were undertaken for searching information with computers, out of which a key development was reported in [41]. In this research, the original ideas of using words as document indexing units and employing word overlapping frequencies as the relevance score between a query and a document were adopted.

Sequently, in the 1960s, a few of IR systems were developed, in which the most influential ones were the Cranfield evaluations [42] and the System for the Mechanical Analysis and Retrieval of Text (SMART) [43]. In [42], Cleverdon and his colleagues at Cranfield College of Aeronautics established a milestone in the evaluation of IR systems, *i.e.*, the method of recall and precision curve. Briefly, this method requires the test document repository containing a fixed number of documents. Then, the documents are marked as relevant documents according to a set of specific topics by human assessors. The effectiveness of an IR system then is computed as a recall and precision curve, which is described detailedly in Section 3.5 of this thesis. The SMART system was developed at Cornell University between 1961 and 1964 as the first IR system capable of indexing up to 500 Megabyte (MB) of documents. With the development of SMART, many important IR concepts were introduced, *e.g.*, VSM and relevance feedback and so on. The evaluation methodology presented in SMART, *i.e.*, single term measure, synonym recognition and so on, paved the way for many further critical developments in the field of IR.

Due to the lack of large document repositories, techniques used for IR of large document repositories have not been published until 1992. Since that year, with the help of Text REtrieval Conference (TREC) [44] data sets, a number of new and/or modified methods have been published for IR of large corpus. TREC is a series of workshops co-sponsored by the National Institute of Standards and Technology (NIST) and the Disruptive Technology Office of the U.S. Department of Defense. Briefly, TREC focuses on a number of IR research areas, *e.g.*, the retrieval of spoken information, non-English language retrieval, information filtering, user interactions with a retrieval system and so on. The purpose of TREC is to encourage the research of IR from large text collections and increase the speed of the lab-to-product transfer of novel IR technologies. Based upon the large document repositorys provided by TREC, many techniques were developed with the aim of effectively retrieving information over a large number of documents.

From 1996 to 1998, the idea of employing the IR techniques into Internet web retrieval was proposed. Meanwhile, corresponding IR systems were developed, *e.g.*, Google [45], Yahoo [46] and MSN [47] search engines and so on. In these web search engines, a set of web-based evidence is taken into the consideration for document ranking, *e.g.*, hyperlink recommendation and so on, which is out of the research scope of this study. Therefore, the document ranking techniques of web search engines are not discussed in the thesis. In the following subsection, a set of document ranking techniques of ODSEs are surveyed.

2.2.2 Mathematical models for the document ranking of an ontology-based document search engine

In the past few years, many IR models have been developed for document ranking in document search engines, in which VSM is the dominant one and widely utilised in ODSEs [21] [22] [23] and [24]. In this subsection, several document ranking models are briefly reviewed firstly. Then, VSM is described in detail, as the proposed ER-based document ranking approach is dedicated to overcoming the drawbacks of document ranking algorithms used in ODSEs, which in turn, are developed based upon VSM.

Boolean model

The Boolean model is the first operational mathematics model for IR [13]. In an IR system with a Boolean model, query terms are formed with Boolean operators, *i.e.*, AND, OR and NOT. Practically, Boolean model-based IR systems have several significant limitations. For example, there is not a degree of match for a document. In the other words, the document either satisfies a query or does not relate to the query completely. Although such a system can clearly illustrate the reason of why retrieving a specific document, it is too hard for common users to input a good query, fully stating their information needs. Meanwhile, in a search result, the identified relevant documents are normally ranked according to the features of the documents, *e.g.*, the date of document creation and so on. That is to say, document ranking with the relevance scores between a query and a set of relevant documents is often not critical in a Boolean model-based IR system. Therefore, the difficulty of users in navigating their interested information is increased.

Probabilistic model

The original idea of using a probabilistic model for IR was proposed by Maron and Kuhns [48] in 1960 and then updated by Robertson and Jones in [49]. Subsequently, a number of probabilistic models have been presented, each of which was established based on a different probability estimation technique. However, a general principle was adopted during the development processes of all these probabilistic models, *i.e.*, in an IR process, the documents from a document repository should be ranked in the descending order of their probabilistic relevance regarding an input query.

Currently, the most widely employed probabilistic ranking formulation in the field of IR is known as the Okapi BM25 algorithm [50]. In BM25, the relevance weight assigned to a document regarding a term of a query is computed with a set of parameters, such as the weight of the term in the query, the total number of terms in the query, the frequency of the term occurring in the document, the total number of documents in a document repository, the number of documents from the collection containing the term, the length of the document and the average length of all the documents from the document repository (both measured in Bytes). Consequently, the final relevance score between the document and the whole query is calculated as the sum of all the term weights assigned to the document.

Inference network model

The first Inference Network (IN) model was introduced by Turtle in 1990 [51]. Basically, IN is a Bayesian network [52], which is used to model the documents of a document repository and a query in an IR process, respectively. An IN model is composed of two sub-networks, *i.e.*, a Document Network (DN) produced during an indexing process and a Query Network (QN) produced from a query during a search process.

DN is used to represent a document repository as a set of document nodes. Also, all the terms extracted from the collection are described as a number of concept nodes, *i.e.*, document term nodes. Subsequently, a document assigns a certain strength to each of its terms. On the other hand, QN defines a query as a bag of query concept nodes, *i.e.*, query term nodes. With QN, statistical operators (*e.g.*, Sum) and Boolean operators (*e.g.*, AND, OR and NOT) are permitted to be submitted with a query.

In an inference process of IN, the relevance score between a document and a query is accumulated from the multiple terms of the query, with their strengths assigned by the document. From an operational perspective, the strength of a term concerning a document can be treated as the weight of the term regarding the document. Therefore, the relevance calculation function of this model becomes similar to that of the probabilistic model stated above and VSM introduced in the next subsection. There is one thing to be mentioned, the strength of a term for a document is not defined by the model, and thus can be computed by employing any suitable algorithms.

Vector space model

1. Introduction to VSM: The concept of VSM was firstly introduced by Salton [43] in 1971. Briefly, VSM is a mathematical model used for determining the relevance score between a query input and a specific document from an indexed document repository. In a document ranking process using VSM, firstly, a document d is conceptually represented as a vector. Each keyword extracted from the document is stored as a component of the vector. In the case that a keyword appears more than one time in a document, it is also defined as one component without considering its frequency. Meanwhile, a query input q is also defined as a vector containing the query terms, each of which is denoted as a vector component. Then, the weight of a query term in the document vector is calculated by a weight method, *e.g.*, term frequency-inverse document frequency (*tf-idf*) method. In *tf-idf*, such a term weight is computed with two factors: $tf_{d,t}$ and df_t. $tf_{d,t}$ is denoted as the frequency of a query term t occurring in q for d and df_t is defined as the number of documents in the indexed document repository containing t. Consequently, the term weight $w_{d,t}$ can be obtained with equation (2.2.1):

$$w_{d,t} = tf_{d,t} \times idf_t = tf_{d,t} \times log\frac{N}{df_t}, \tag{2.2.1}$$

where N is the total number of documents in the indexed document repository and idf_t is denoted as the inverse document frequency of t. As illustrated by (2.2.1), the value of $w_{d,t}$ increases when t appears more frequently in d or the number of documents identified from the indexing document repository, holding t, is reduced. On the other hand, the weight of t in q, *i.e.*, $w_{q,t}$ can be also calculated with *tf-idf*, as shown in equation (2.2.2):

$$w_{q,t} = tf_{q,t} \times idf_t. \tag{2.2.2}$$

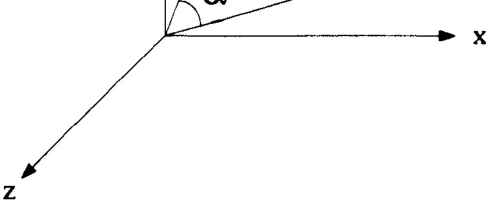


Figure 2.1: A classical VSM

Once $w_{d,t}$ and $w_{q,t}$ of each query term in q are obtained, the total relevance score between the query vector and the document vector is computed with a cosine measure. For instance, $\overrightarrow{V}(d)$ and $\overrightarrow{V}(q)$ are defined as the vectors of d and q respectively and shown as Figure 2.1. As indicated by the figure, the angle between the two vectors is α. With the cosine measure, the relevance score between $\overrightarrow{V}(d)$ and $\overrightarrow{V}(q)$, *i.e.*, $Sim_{d,q}$ is defined as [35]:

$$\begin{aligned} Sim_{d,q} = \cos(\alpha) &= \frac{\overrightarrow{V}(d) \cdot \overrightarrow{V}(q)}{\mid \overrightarrow{V}(d) \mid\mid \overrightarrow{V}(q) \mid} \\ &= \sum_{t=1}^{T} RS_t, \end{aligned} \tag{2.2.3}$$

where T is the total number of the query terms in q and RS_t ($t = 1, \ldots, T$) is the relevance score derived between t and d, and

$$RS_t = \frac{w_{d,t} \times w_{q,t}}{\sqrt{\sum_{t=1}^{T} w_{d,t}^2} \times \sqrt{\sum_{t=1}^{T} w_{q,t}^2}}. \tag{2.2.4}$$

As observed from equation (2.2.3), the relevance score between q and d is mathematically computed by the sum of RS_t ($t = 1, \ldots, T$), generated between the query terms of the query q and the document d.

In the following content of this subsection, a set of VSM-based ODSEs is reviewed. Then, the drawbacks of these ODSEs are analysed, followed by the explanation of the purpose of developing the proposed ER-based document ranking approach.

2. Review of ODSEs developed with VSM: In the past few years, VSM has been widely employed for document ranking in ontology-based IR projects.

In [21], Voorhees used WordNet as a knowledge model for QE, and a set of tests were carried out based upon the TREC document repository. In order to obtain the best performance with the proposed QE strategy, all the expansion terms regarding a query input were selected manually. In an expanded query, the included terms were kept separately using the Extended Vector Space Model (EVSM) [53]. Then, the relevance score between a document and the expanded query was computed by the sum of the relevance scores generated between the document and each term of the expanded query, with the consideration of assigned mutual weights. In experiments, little improvement of search accuracy was achieved for long queries. This was due to that, the user's information demands were already well described by a long query even without QE. But in contrast, the search accuracy of a less complete query was significantly improved with the manually selected expansion terms.

Fu, Jones and Abdelmoty [22] reported a spatial QE technique developed in an European Union (EU) semantic web project, namely SPIRIT. In the study, a set of factors was considered when expanding a spatial query, *i.e.,* types of spatial terms extracted from a geographical ontology model, types of non-spatial terms defined in a domain ontology model and the semantics of spatial relationships and their context of use and so on. In a search process with an expanded query, firstly, a geographical relevance was calculated by a geographical confidence and the emphasis degree of a place name in the document. Then, a textual relevance was computed using VSM of IR. Finally, the overall relevance score between the expanded spatial query and the document was achieved by a weighted sum of the geographical relevance and the textual relevance. Meanwhile, a textual only search was also performed in the study, once the spatial QE technique was turned off. In the tests, the search performance of a search engine was significantly improved by the proposed spatial QE technique.

In [23], a semantic IR system was introduced. In the system, a knowledge base was first constructed by annotating the documents of a document repository, using an ontology-based semiautomatic annotation mechanism. Subsequently, based upon the knowledge base, a query was expanded by semantic inferencing mechanisms. In a search process, a semantic IR model, developed with an adaptation of VSM, was implemented for document retrieval using the expanded query. However, the search performance of such an IR system could be greatly reduced due to the incompleteness of the knowledge base. Hence, a keyword-based search model of VSM was also employed in the system, with which a satisfactory search accuracy have been achieved even without the knowledge base. The relevance score between the expanded query and a document was then computed by the weighted sum of a semantic similarity value and a keyword-based similarity value, generated by the semantic IR model and the keyword-based search model respectively. The test results showed that the proposed system achieved significant improvements in IR, in comparison with a keyword-based IR system.

A semantic search engine, used for IR in web resources, was presented in [24]. Firstly, target web resources were semantically annotated based upon an open semantic platform. Then, in a search process, a query was suitably expanded with its relevant concepts by performing a concept navigation using a domain ontology model. This procedure ensured that a large number of relevant web resources could be discovered in the search process, compared with that achieved using the original query. Finally, VSM was employed to calculate the overall relevance score between the expanded query and a web resource. The experimental results demonstrated that the semantic search engine has improved the retrieval performance in terms of precision and recall, compared with a keyword-based IR system.

As illustrated in the above projects, with a well-defined ontology model, a high accuracy of an ODSE may be achieved in a search process. Nevertheless, drawbacks still exist in these ontology-based search engines, as explained in Section 1.4.1. In order to overcome these problems, in this study, the relevance scores of the query terms, derived from a SONT-based QE process, are combined with the proposed ER-based approach, as explained detailedly in the following sections.

2.3 A typical ontology-based document search engine

The architecture of an ODSE as well as the component functions of the ODSE are introduced in this section. As indicated in Figure 2.2, an ODSE is mainly composed of three function modules, *i.e.*, an indexer, a query processor and an interface.

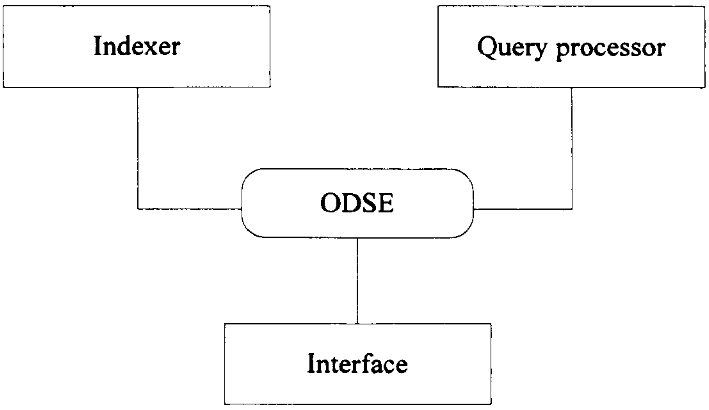


Figure 2.2: Components of an ODSE

- The indexer is a mathematical model to extract valuable information contained within documents. The generated information then is transferred into a query processor accessible format and stored in an index server. A typical indexing process may include splitting a document into constituent terms, calculating the frequencies of the terms within a document and a whole document repository and so on.

- When the index server of an ODSE is successfully built by the indexer, the ODSE is ready to process users' queries with the query processor. The main function of the query processor is to firstly expand a query with an employed ontology model; and secondly match and rank documents from the index server according to the expanded query.

- The interface of the ODSE is used to display search results and link the discovered relevant documents for users. In a search result, a list of relevant documents is ranked in the descending order of their relevance scores concerning an expanded query.

2.4 Introduction to ontology

In this section, the fundamentals of the Ontology technique are given firstly. Subsequently, a domain ontology development process is illustrated.

2.4.1 Fundamentals of ontology

Many definitions of the Ontology technique have been made since it became a popular research topic in AI from the beginning of the 1990s. In these definitions, the one which best characterises the essence of an ontology model is given as: an ontology model is a formal, explicit specification of a shared conceptualisation [54]. In practice, an ontology model is used to depict a set of concepts with their semantic meanings and mutual relationships [55].

The nature of an ontology model is designed for the integration of heterogeneous data extracted from data sources, with a machine-processable representation language. A well-defined ontology model provides a "clear and rigorous vocabulary" [56], *e.g.*, WordNet and Cyc [57], which is explicitly defined and separate from an implementation process. With an ontology model, a communication understanding can be supported across human and computer applications.

Ontology plays a key role in defining the semantics of information for intelligent systems [55]. As illustrated in Figure 2.3, previously, the Ontology technique has been investigated in a variety of areas, such as IR [21] [22] [23] [24], knowledge engineering [58] [59] and natural-language processing [60] and so on.

In an IR system, *e.g.*, a document search engine, one significant application of an ontology model is highlighted as QE. However, besides QE, ontologies also have been implemented in a number of other IR applications [20], *e.g.*, indexing, thematic summarisation, query formulation, IR cross multi-languages and so on.

In practice, the main objective of an ontological knowledge engineering process is to facilitate the construction of a domain ontology model. Briefly, a domain ontology model is an ontology model that describes the context of a

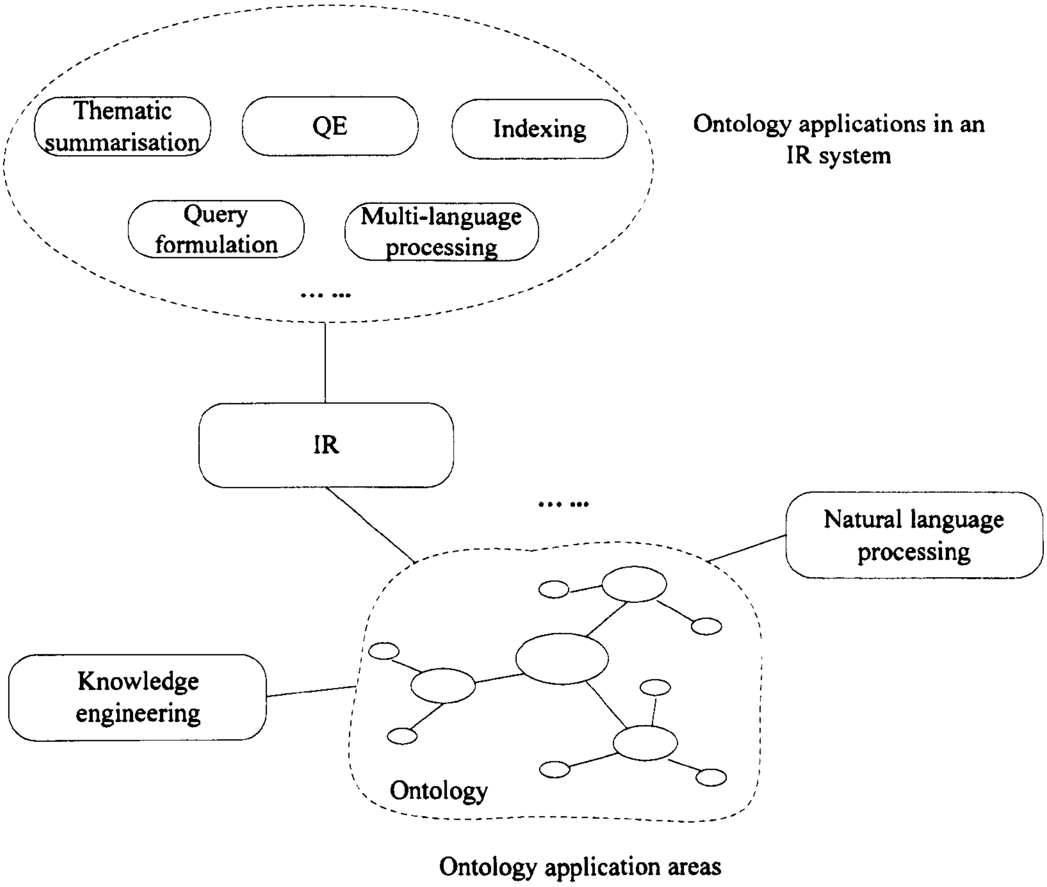

Thematic summarisation
QE
Indexing
Ontology applications in an IR system
Query formulation
Multi-language processing
... ...
IR
... ...
Natural language processing
Knowledge engineering
Ontology
Ontology application areas

Figure 2.3: Ontology applications

specific domain, *e.g.*, a power substation. With a domain ontology model, the heterogeneous information of a specific domain can be integrated and a common vocabulary for knowledge sharing is provided. In an engineering process of formalising a domain ontology model, many efforts have been taken to organise various concepts and the relationships among them, which are exhibited in the domain. In the efforts, the most useful method is to consult human experts related to the domain. With the knowledge of the experts, a domain ontology model then can be constructed by ontology developers using a toolkit. Subsequently, in a QE process using the domain ontology model, a query term matches to a unique ontology concept regarding a specific domain. As a result, the meaning of the term can be effectively disambiguated. The related terms of the query term then can be accurately selected from the domain ontology model for expanding the query term.

2.4.2 Ontology languages

In the development process of a domain ontology model, an ontology language is required for programming. In the past decade, several ontology languages have been published which involve Extensible Markup Language (XML) [61], Extensible Markup Language Schema (XML Schema) [62], Resource Description Framework (RDF) [63], and Resource Description Framework Schema (RDF Schema) [64], DARPA Agent Markup Language+Ontology Interence Layer (DAML+OIL) [65] and Web Ontology Language (OWL) [66] and so on. In 2004, OWL has been recommended as a World Wide Web Consortium (W3C) standard language for programming ontologies. Therefore, ontology development softwares have been exploited to support the construction of OWL ontologies, out of which Protege [67] is the most widely employed one.

In OWL, three sub-languges are defined, *i.e.*, OWL Lite, OWL Description Logic (DL) and OWL Full [68]. These three sub-languages are briefly introduced below:

- OWL Lite: OWL Lite is developed based upon the syntax of RDF Schema by adding some properties to express conception definitions and axioms. It contains a hierarchical classification method and only offers simple constraints for describing knowledge. OWL Lite is arranged as the base of the OWL languages and kept as a very simple description language.

- OWL DL: Briefly, OWL DL is developed based upon DL, which in turn is a reasoning mechanism derived from First Order Logic (FOL) [69]. In these three sub-languages, OWL DL is designed as the most computational sub-language of OWL.

- OWL Full: OWL Full interprets a more complete vocabulary for defining entities in an ontology model, compared with OWL Lite and OWL DL. Therefore, a more expressive power is provided. However, OWL Full offers no computational guarantees in logic, which shows the main limitation for implementing OWL Full ontologies.

2.4.3 Development process of a domain ontology model

The ontology development method of the Enterprise ontology model, reported by Ushold [70], has been followed by many later domain ontology developers in their domain ontology building processes: the identification of an ontology building purpose, building an ontology model, the evaluation of the ontology model and the documentation of the ontology model.

Furthermore, the ontology building step of the method is described as: an ontology model should be firstly developed as an informal ontology model, which is subsequently formalised with a formal ontology language, *e.g.*, OWL, and transferred as a formal ontology, *e.g.*, SONT.

As illustrated in Figure 2.4, the process for constructing an informal ontology model is mainly composed of four steps as well, which include the collection of ontology concepts, the clustering of the collected concepts, the refinement of the concept set by investigating what concepts are basic, what are generic, or specific and finally the relationship denotation among the obtained concepts. In this process, the names of the ontology concepts must be defined differently and each concept must have only one meaning with respect to a specific domain. However, the informal ontology model cannot be processed by computers unless it is represented with a formal and computer processable ontology language. Hence, in the next step, the informal ontology model is reorganised as a formal ontology model with a formal ontology language, *e.g.*, OWL.

In this study, a domain ontology model, *i.e.*, SONT, has been constructed and employed for QE in an ODSE. The development process of SONT followed the domain ontology construction method explained above. SONT was con-

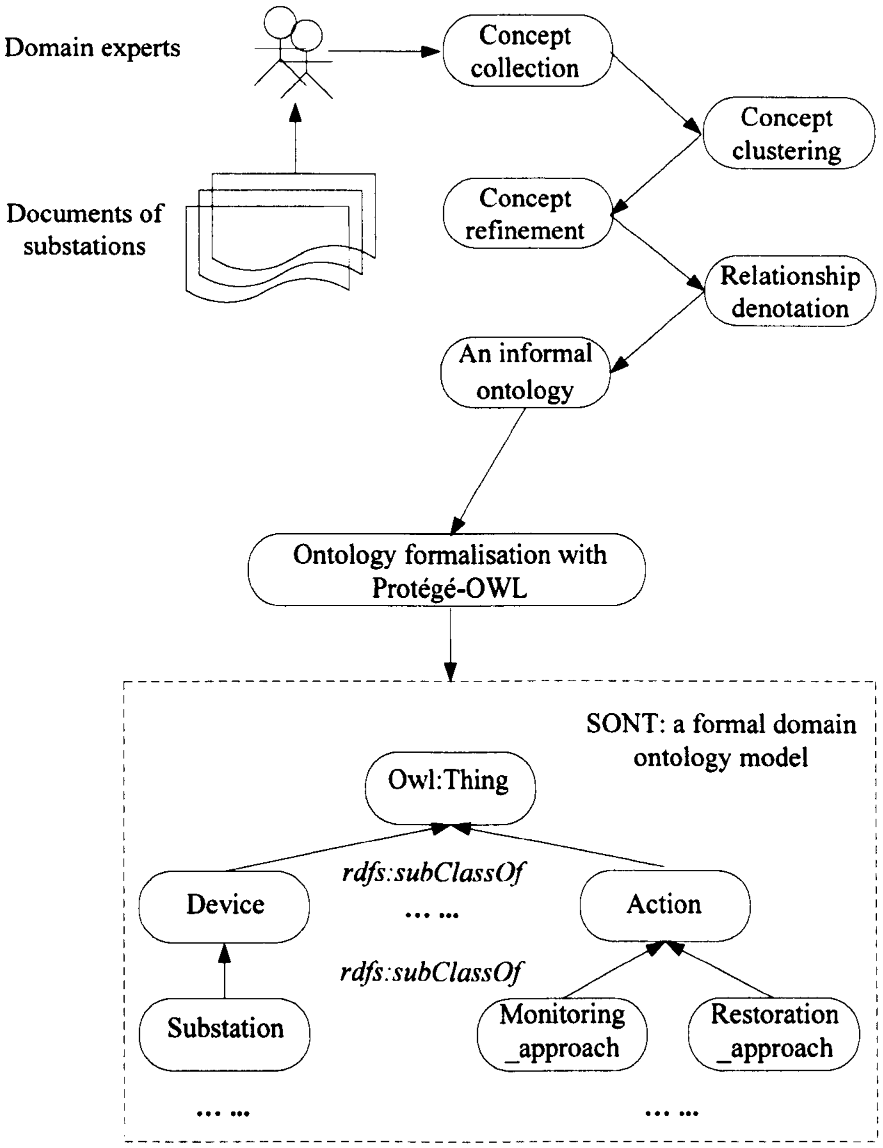


Figure 2.4: A domain ontology model development process

structed with the Protege ontology development software. A standard ontology langauge, *i.e.*, OWL DL, was employed for programming SONT.

Compared with other standard ontology languages recommended by the W3C organisation, *i.e.*, OWL Lite and OWL Full, OWL DL offers a greater capability for discovering latent relationships between the concepts of a domain ontology model, based on a logical reasoning method. Therefore, in a QE process using an OWL DL-based ontology model, more accurate terms may be automatically selected for expanding a given query, compared with that obtained by OWL Lite and/or OWL Full ontologies. As a result, an enhanced search accuracy of a document retrieval process may be achieved.

Significantly, the development of SONT is an initial investigation of defining the knowledge of a substation as a domain ontology model. The whole process for building SONT was based upon the knowledge extracted from WordNet. However, WordNet is not a lexical thesaurus particularly designed for depicting the context of a power substation. Therefore, the employed knowledge of WordNet was not sufficient and specific for the development of SONT. Hence, in the study, a knowledge expansion process of SONT was carried out by power system experts instead. Finally, SONT has been constructed with 413 classes (*i.e.*, concepts), 67 properties and 31,579 instances of the classes. With the above procedures, the meanings of the concepts and the relationships among the concepts of SONT are clearly defined. Consequently, a reliable QE process can be achieved with SONT. In the next section, the basics of ER are introduced.

2.5 Basics of evidential reasoning algorithm and dempster-shafer theory

In a decision making process, the usage of different evidence can derive different conclusions. In practice however, a lot of evidence may still be involved in such decision making problems with uncertainties, *e.g.*, ignorance and randomness. Therefore, an evidence combination approach is required to produce an ultimate conclusion based upon the evidence. Regarding this problem, an ER approach is introduced in the thesis, which is dedicated to solving decision making problems concerning a set of quantitative and/or qualitative evidence accompanying uncertainties.

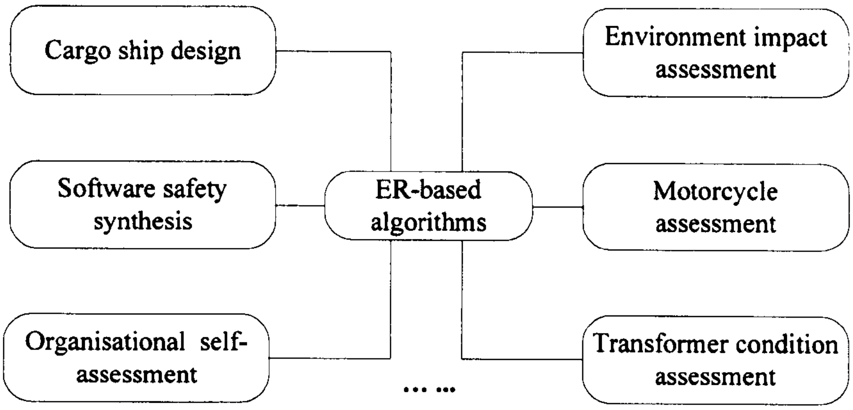

Cargo ship design
Software safety synthesis
Organisational self-assessment
ER-based algorithms
...
Environment impact assessment
Motorcycle assessment
Transformer condition assessment

Figure 2.5: ER applications

As can be seen from Figure 2.5, in recent years, the ER algorithm has been applied for dealing with decision analysis, assessment and evaluation problems in a number of projects, such as general cargo ship design [71], environmental impact assessment [72], motorcycle assessment [10], power transformer condition assessment [73], organisational self-assessment [74], software safety synthesis [75] and so on.

The ER algorithm is developed based upon a MADM tree model and the evidence combination rules of DS. Briefly, the DS theory is a mathematical method dedicated to combining evidence with uncertainty. The seminal work on this subject was set by Dempster in 1967 [76] and subsequently expanded by Shafer in 1976 [38]. Compared with other techniques used for evidence combination, *e.g.*, IN, the most attractive feature of the DS theory is highlighted as its specific design for handling uncertainty during an evidence combination process. In the ER algorithm, all evidence is organised into a MADM tree model and treated as a set of attributes. Furthermore, different weights can be assigned to the evidence. In order to assess the attributes, a few evaluation grades are defined, which may be quantified using certain scales. Then, each attribute is evaluated to one of the grades. With the combination rules of DS, an aggregation process is implemented using the available attributes and the overall output of the MADM tree is derived as the final conclusion. In the following sections, the ER-based approach to document ranking in an ODSE is described in detail.

2.6 Methodology for organising an expanded query

2.6.1 Query expansion with a domain ontology model

Using an ontology model for QE in an IR system was firstly reported by Voorhees [21]. In that research, an ontology model implementation method of QE was outlined, which has been widely adopted in the studies of ontology-based QE: firstly, the terms of an original query must be disambiguated, so that every term maps to a unique concept in an employed ontology model. Then, in the ontology model, terms related to the disambiguated query concepts are selected and added to the original query.

In an ontology-based QE task, specific kinds of related terms can be selected for expanding a query term, *i.e.*, "synonyms", "synonyms and hyponyms", "synonyms, hyponyms and hypernyms", "meronyms" and so on or a mixture of these relations [77]. However, this study focuses on the implementation of the proposed ER-based approach, designed for evidence combination in a document ranking process. Hence, not all the above QE methods were investigated. The idea stated in [78] was adopted in the study, *i.e.*, a QE process with an ontology model is normally achieved by extending provided query terms with synonyms and/or hyponyms. In other words, in this study, a query term was extended with its synonyms and hyponyms only. Nevertheless, the proposed ER-based approach can also be implemented in an ODSE if a query term is further expanded with its hypernyms and meronyms. In such cases, a method is required to suitably allocate the hypernyms and meronyms into a MADM tree model. Then, the relevance scores generated between the terms of the expanded query and a document can be combined with the ER algorithm.

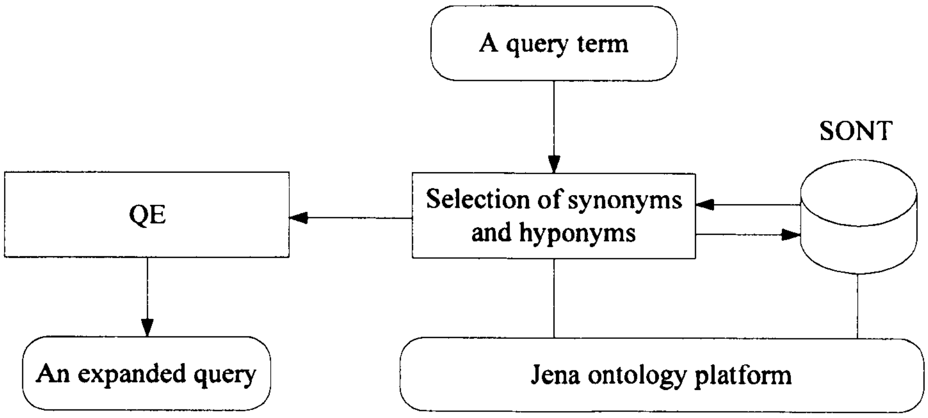


Figure 2.6: A QE process with SONT

As mentioned in Section 2.4.3, a domain ontology model called SONT, used for QE in an ODSE, was developed in the study. A QE process with SONT is illustrated in Figure 2.6. As can be seen from the figure, a Jena [79] ontology platform was employed for accessing SONT. In a QE process regarding the context of a power substation, the advantage of using SONT, compared with employing a textual thesaurus [80], is: a query term matches an ontology concept name of SONT concerning the domain of power substation. As a result, the meaning of the term can be restricted within the domain of substation and effectively disambiguated before a QE process, which is not achievable by published textual thesauri. Thus, a high accuracy of a QE process may be achieved by SONT.

As stated above, in this study, SONT was used for extending a provided query term with its synonyms and hyponyms (subclasses) only. For example, with SONT, a synonym set and a hyponym set were derived as (2.6.1) and (2.6.2) respectively, given a query as *Fault diagnosis.* As can be seen from (2.6.1) and (2.6.2), the singular and the plural of a query term were treated as different terms.

This is due to that the two forms of the term are practically considered as two separate words during a document search process with the employed document search engines of this study. Thus, in the case that a query term is not expanded with its plural, a number of documents, which do not contain the original query term, but are correlative to its plural, will not be retrieved during a subsequent document search process. Consequently, the search accuracy of a search engine can be seriously reduced.

For comparison purposes, a QE process of *Fault diagnosis* was also implemented with WordNet, in which none of synonyms or hyponyms were discovered for *Fault diagnosis*. Therefore, in a document search process of *Fault diagnosis*, more useful documents may be retrieved using the expanded query derived from SONT, compared with that of WordNet. Consequently, a higher search accuracy is possibly achieved with SONT than using WordNet.

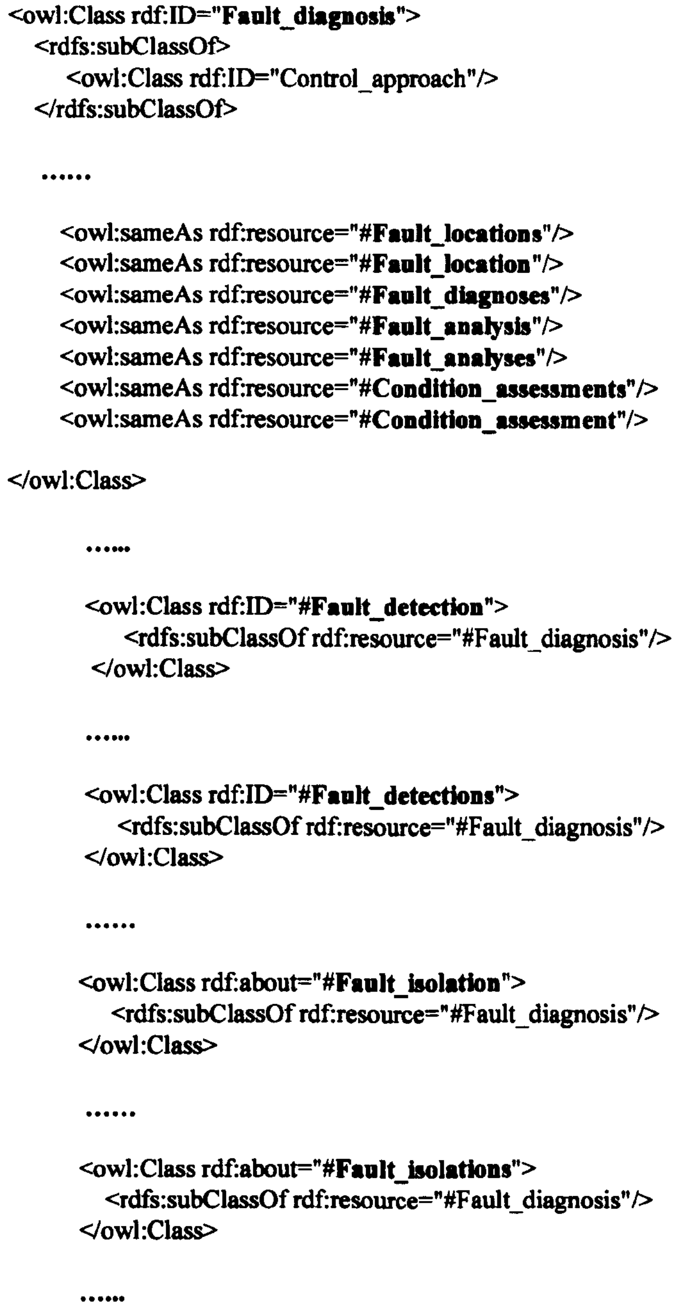
$$\begin{aligned}&\{\mathit{Fault\ diagnoses},\ \mathit{Condition\ assessment},\\&\mathit{Condition\ assessments},\ \mathit{Fault\ location},\\&\mathit{Fault\ locations},\ \mathit{Fault\ analysis},\ \mathit{Fault\ analyses}\}\end{aligned} \tag{2.6.1}$$

$$\begin{aligned}&\{\mathit{Fault\ detection},\ \mathit{Fault\ detections},\\&\mathit{Fault\ isolation},\ \mathit{Fault\ isolations}\}\end{aligned} \tag{2.6.2}$$

In SONT, the relationships among the above terms, can be described with an OWL ontology segment as shown in Figure 2.7. With respect to the hierarchical relationships among an original query, its synonyms and hyponyms, the methodology of organising an expanded query as a MADM tree model is introduced in the next subsection.

2.6.2 Organising an expanded query with a multiple attribute decision making tree model

In the context of a document search engine, a relevance score denotes the relatedness between a query term and a document. In the case of QE, the relevance score between an expanded query and a document is generated based



```
<owl:Class rdf:ID="Fault_diagnosis">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Control_approach"/>
  </rdfs:subClassOf>

  ......

    <owl:sameAs rdf:resource="#Fault_locations"/>
    <owl:sameAs rdf:resource="#Fault_location"/>
    <owl:sameAs rdf:resource="#Fault_diagnoses"/>
    <owl:sameAs rdf:resource="#Fault_analysis"/>
    <owl:sameAs rdf:resource="#Fault_analyses"/>
    <owl:sameAs rdf:resource="#Condition_assessments"/>
    <owl:sameAs rdf:resource="#Condition_assessment"/>

</owl:Class>

      ......

      <owl:Class rdf:ID="#Fault_detection">
         <rdfs:subClassOf rdf:resource="#Fault_diagnosis"/>
       </owl:Class>

      ......

      <owl:Class rdf:ID="#Fault_detections">
         <rdfs:subClassOf rdf:resource="#Fault_diagnosis"/>
      </owl:Class>

      ......

      <owl:Class rdf:about="#Fault_isolation">
         <rdfs:subClassOf rdf:resource="#Fault_diagnosis"/>
      </owl:Class>

      ......

      <owl:Class rdf:about="#Fault_isolations">
        <rdfs:subClassOf rdf:resource="#Fault_diagnosis"/>
      </owl:Class>

      ......
```

Figure 2.7: An ontology program segment of an expanded query

on a set of relevance scores, which are achieved between the query terms of the expanded query and the document. As mentioned in Section 1.4.1, an unreliable expansion method, *i.e.*, appending expansion terms to an original query with assigned weights only, may bias the focus of the original query and thus leads to an unappreciated search performance. This is due to the fact that, the query terms are organised without the consideration of their hierarchical relationships *i.e.*, a hyponym of the original query is located at the same level of the original query. Therefore, in this study, a MADM tree model was employed for the organisation of the query terms from an expanded query. The essential idea of this method is to organise an expanded query into a hierarchical tree model. In the tree model, the synonyms are located at the same level as the original query and the hyponyms are distributed at a lower level. The relevance scores generated by these expansion terms concerning a document then are designated as auxiliary evidence for evaluating the overall relatedness between the expanded query and the document.

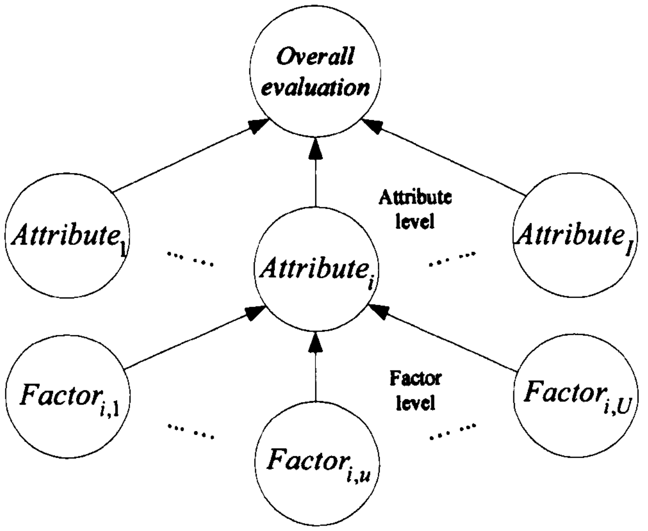


Figure 2.8: A MADM tree model

A generalised MADM tree model is illustrated as Figure 2.8, in which, a

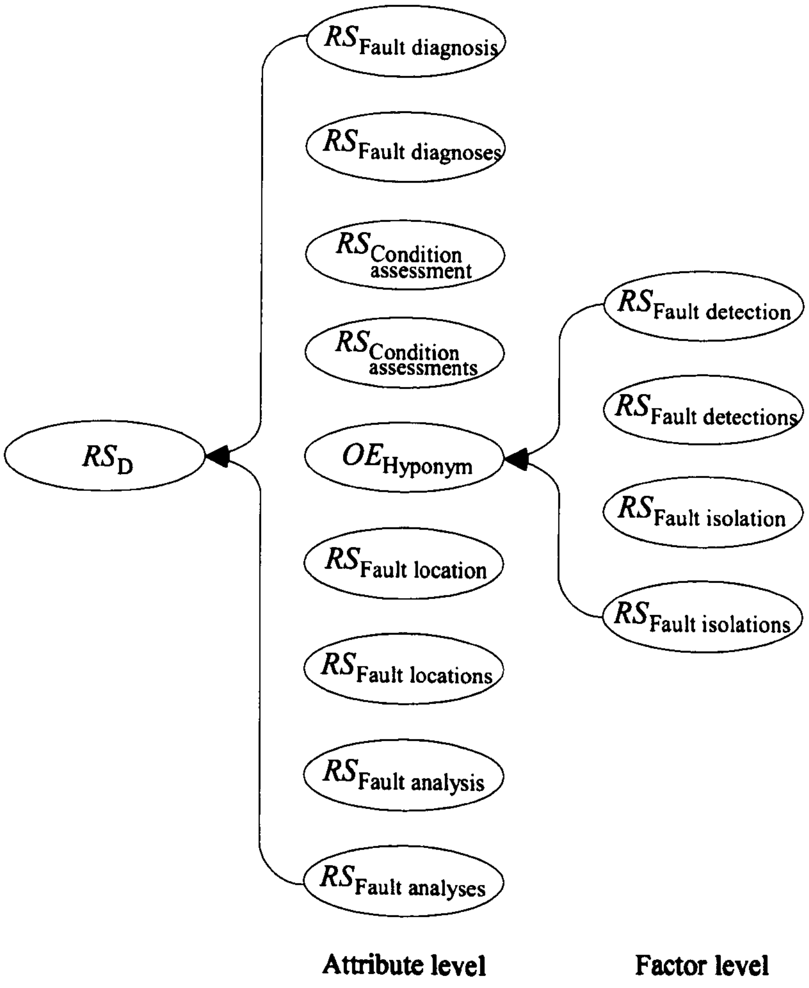

$RS_{\text{Fault diagnosis}}$
$RS_{\text{Fault diagnoses}}$
$RS_{\text{Condition assessment}}$
$RS_{\text{Condition assessments}}$
RS_{D}
OE_{Hyponym}
$RS_{\text{Fault location}}$
$RS_{\text{Fault locations}}$
$RS_{\text{Fault analysis}}$
$RS_{\text{Fault analyses}}$
$RS_{\text{Fault detection}}$
$RS_{\text{Fault detections}}$
$RS_{\text{Fault isolation}}$
$RS_{\text{Fault isolations}}$
Attribute level
Factor level

Figure 2.9: A tree model used for the combination of multiple relevance scores

set of nodes are organised with a hierarchical structure. Two different levels, namely a factor level and an attribute level, are involved. In an evidence combination process used for calculating the value of *Overall evaluation*, a set of attributes ($Attribute_i$ ($i = 1, \ldots, I$)) from the attribute level are evaluated. Meanwhile, the values of these attributes can either be obtained from users and/or mathematical algorithms, or generated by several factors located in the factor level. For instance, the value of $Attribute_i$ is determined by $Factor_{i,u}$, where $u = 1, \ldots, U$.

As introduced above, an expanded query derived from the query *Fault diagnosis* can be organised as a tree model by considering the hierarchical relationships among the involved terms. The relevance score (RS_{D}) between the query *Fault diagnosis* and a specific document is defined as the value of *Overall evaluation* node in Figure 2.8. The relevance scores concerning the document, which are determined by the query terms from the synonym set (2.6.1) and hyponym set (2.6.2), are regarded as auxiliary evidence for deciding RS_{D}.

Returning to the MADM tree model, a decision making process to generate RS_{D} then is depicted as shown in Figure 2.9. As can be seen from the figure, the four hyponyms of *Fault diagnosis* are located in the factor level. In the decision making process, their relevance scores regarding the document are utilised to generate OE_{Hyponym} with the ER algorithm, which denotes an overall relevance score generated by the hyponyms. In the attribute level of Figure 2.9, the relevance scores decided by *Fault diagnosis* itself and the terms from the synonym set (2.6.1) are included, as well as the value of OE_{Hyponym}. The overall output of this MADM tree model, defined as RS_{D}, is then calculated with the ER algorithm as well.

2.6.3 Decision making with evidential reasoning algorithm

A decision making process often comes with uncertainties. In order to obtain RS_{D} shown in Figure 2.9 with available evidence, the ER algorithm was adopted in the study. The utilisation of the ER algorithm aims at providing a mathematical framework for the combination of evidence in a document ranking process, which is from different levels with uncertainty.

In this study, all the search engines were developed based on an Apache Lucene search engine library [81]. In Lucene, a relevance score obtained between a query term and a document is determined with a Boolean model and VSM. In the task of generating relevance scores between a term from a query and a large number of documents, firstly, the Boolean model is utilised to filter the unrelated documents from the document set regarding the whole query, based upon on a Boolean logic. Then, the relevance scores between each term of the query and the documents that exist in the filtered document set are calculated by VSM, as explained in Section 2.2.2.

In the experiments of the research, all the relevance scores regarding a search process were generated with a Lucene-based search engine. Thus, it was ensured that the relevance scores generated in the study were with an equal accuracy. Consequently, the uncertainty in such a search process focused on the mutual weight between a pair of evidence only. The ER algorithm, which is based on the DS theory and used for evidence combination, is described detailedly in Section 2.7.

2.7 Evidence combining of a multiple attribute decision making tree model

2.7.1 Dempster-shafer rules used for evidence combination

The main interest of investigating DS is to explore its effect of combining evidence for a MADM tree model. The theory of combination is stated in this section and only the rules applied in the research are described below.

In DS, a frame of discernment is defined as Θ. A hypothesis, *i.e.*, a singleton, in Θ is denoted as Ψ ($\Psi \subseteq \Theta$). The hypotheses in Θ are assumed mutually exclusive and each of the hypotheses can be treated as a one-element subset, *i.e.*, a singleton. Thus, for Θ containing n different hypotheses, the size of all its possible subsets is 2^n, including ϕ and Θ itself. As a result, the set of the original n hypotheses of Θ, to which belief can be assigned, is enlarged to 2^n distinct hypotheses.

A basic probability assignment $m_A(\Psi)$ represents the belief of a piece of evidence A supporting the hypothesis Ψ, where the value of $m_A(\Psi)$ is assigned from a close interval [0, 1]. This portion of belief cannot be further subdivided among the subsets of Ψ and does not include portions of belief committed to subsets of Ψ. Briefly, a basic probability assignment is a generalisation of the traditional probability density function of the Bayesian theory, which assigns a probability with a number in the range [0, 1] to every singleton of Ψ and the sum of these probabilities is one. In the enlarged Ψ with 2^n hypotheses, each hypothesis is assigned with a basic probability assignment m, valued from the range [0, 1]. Furthermore, the sum of these basic probability assignments is one.

Consider that two pieces of evidence (A and B) are provided for evaluating the value of the hypothesis Ψ. As a result, two probability assignments $m_A(\Psi)$ and $m_B(\Psi)$ are confirmed. In the DS theory, the evidence A and B must be ensured as independent, *i.e.*, $A \cap B = \phi$. Then, $m_A(\Psi)$ and $m_B(\Psi)$ are combined using the Dempster's rule of combination. The evidence combination rule used for combining two probability assignments $m_A(\Psi)$ and $m_B(\Psi)$ is defined as follows:

$$\begin{aligned} m_{AB}(\phi) &= 0, \\ m_{AB}(\Psi) &= \sum_{h_1 \cap h_2 = \Psi} \frac{m_A(h_1) m_B(h_2)}{1-K}, \\ K &= \sum_{h_1 \cap h_2 = \phi} m_A(h_1) m_B(h_2), \end{aligned} \tag{2.7.1}$$

where h_1 and h_2 represent two evaluation elements selected from the sample space Θ in all possible ways, in the case that their intersection is Ψ. K is reflected by the conflicting situations where both $m_A(h_1)$ and $m_B(h_2)$ are not equal to 0, but the intersection $h_1 \cap h_2$ is empty (ϕ).

2.7.2 Integrating basic probability assignments with evidential reasoning algorithm

In a document ranking process, the relatedness between an expanded query and one of its relevant documents identified from a document repository is determined by the relevance scores, generated between the terms of the query and the document. According to the DS theory introduced in Section 2.7.1, such relatedness can be defined as a hypothesis Ψ. Then, all the relatedness values generated between the query and its relevant documents from the document repository are denoted as the sample space Θ. On the other hand, each term of the query can be seen as a piece of evidence for supporting the hypothesis Ψ, *i.e.*, relatedness, between the query and a document. Meanwhile, the relevance scores are used as the probability assignments of these evidence for calculating the a relatedness value, with the consideration of their relative weights.

Taking that into account, a document repository named D_{repo} is provided with N different and independent documents. For the query *Fault diagnosis*, totally W documents from D_{repo} are determined as its relevant documents by a Lucene-based ODSE, after it is expanded with (2.6.1) and (2.6.2). Moreover, the relevant document set containing the W documents is defined as D_{relc}. Consequently, each of the terms in (2.6.1) or (2.6.2) as well as the original query *Fault diagnosis* generate W different relevance scores regarding the W relevant documents from D_{relc}. A term *Condition assessment*, which is a synonym of *Fault diagnosis*, is selected for an illustration purpose. A set of relevance scores may be obtained with *Condition assessment* as:

$$\{RS_{\mathrm{ca},1}, \ldots, RS_{\mathrm{ca},w}, \ldots, RS_{\mathrm{ca},W}\}, \tag{2.7.2}$$

where $RS_{\mathrm{ca},w}$ is the relevance score between *Condition assessment* and a relevant document D_w extracted from D_{relc}. The value of $RS_{\mathrm{ca},w}$ is assigned to a close interval [0, 1].

Since the relatedness between *Fault diagnosis* and D_w is denoted as a hypothesis Ψ in DS, $RS_{\mathrm{ca},w}$ can be considered as a confidence degree of *Condition assessment* assigned to D_w. In addition, there are W relevant documents of *Fault diagnosis* that exist in D_{relc} and the upper limit of the total belief committed to these document by *Condition assessment* is one. Each relevance score $RS_{\mathrm{ca},w}$ then can be normalised as:

$$\overline{RS}_{\mathrm{ca},w} = \frac{RS_{\mathrm{ca},w}}{W}. \tag{2.7.3}$$

The evidence set from the attribute level of Figure 2.9 can be defined as e_i $(i = 1, \ldots, L_j)$, where L_j is defined as the number of the evidence e_i. A basic probability assignment, that the evidence e_i supports the overall relatedness between *Fault diagnosis* and the document D_w then, can be expressed as m_i^w. Also, a confidence degree confirmed by the evidence e_i, *e.g.*, $\overline{RS}_{\mathrm{ca},w}$, is denoted by $\beta_{D_w}(e_i)$. Therefore, the basic probability assignment m_i^w, *i.e.*, a belief, may be determined by the following equation:

$$m_i^w = \lambda_i \beta_{D_w}(e_i), \tag{2.7.4}$$

where $\lambda_i = [\lambda_1, \ldots, \lambda_{L_j}]^{\mathrm{T}}$ expresses the relative weight assigned to the evidence e_i in the evidence set. Obviously, if there is only one piece of evidence existed in the attribute level, m_i^w is equal to $\beta_{D_w}(e_i)$. In the next subsection, the algorithm employed for determining the relative weights λ_i is introduced.

2.7.3 Determining relative weights among attributes

In this section, Analytic Hierarchy Process (AHP) [82] is introduced for generating the relative weights λ_i of evidence mentioned in Section 2.7.2. AHP is a multi-criteria decision support methodology used in Management Science (MS) [83]. It has already been verified as a convincing approach to deciding the mutual importance between elements in evaluation models from [84] and [85]. In AHP, the importance values of evidence are assigned as a set of grades, which are judged based on the same benchmark. The fundamental grades utilised in this study are illustrated as Table 2.1.

In AHP, a pair-wise comparison method is applied to a pair of evidence. The grades designed for locating evidence from an AHP pair are confirmed by users. In the document search cases addressed in this study, they were designated by both search engine experts and power system engineers. Let $\mathbf{A_{m,r}}(m = 1, \ldots, L_j;\ r = 1, \ldots, L_j)$ denote a matrix, in which the element $\xi_{m,r}$ represents the ratio comparison of mutual importance values between two evidence in an AHP pair. Therefore, a typical ratio comparison matrix can be

Table 2.1: Grade of measurement for AHP

Values	Definition
1	Equally important or preferred
3	Slightly more important or preferred
5	Strongly more important or preferred
7	Very strongly more important or preferred
9	Extremely more important or preferred
2, 4, 6, 8	Intermediate values to reflect compromise

defined as below:

$$\mathbf{A_{m,r}} = \begin{bmatrix} \xi_{1,1} & \xi_{1,2} & \cdots & \xi_{1,L_j} \\ \xi_{2,1} & \xi_{2,2} & \cdots & \xi_{2,L_j} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{L_j,1} & \xi_{L_j,2} & \cdots & \xi_{L_j,L_j} \end{bmatrix}. \tag{2.7.5}$$

In order to normalise the elements of the above matrix, firstly, the sum of each column is calculated; then, each element is used to divide the sum value generated from its corresponding column. As a result, a normalised matrix is achieved as:

$$\overline{\mathbf{A}}_{\mathbf{m,r}} = \begin{bmatrix} \overline{\xi}_{1,1} & \overline{\xi}_{1,2} & \cdots & \overline{\xi}_{1,L_j} \\ \overline{\xi}_{2,1} & \overline{\xi}_{2,2} & \cdots & \overline{\xi}_{2,L_j} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\xi}_{L_j,1} & \overline{\xi}_{L_j,2} & \cdots & \overline{\xi}_{L_j,L_j} \end{bmatrix}. \tag{2.7.6}$$

Then the average value of each row of matrix (2.7.6), *i.e.*, from row one to row L_j, is generated and assigned as λ_i.

By consulting search engine experts and power system engineers, in the tests, the mutual importance values between the relevance scores generated by an original query term and a synonym, the synonym and OE_{Hyponym}, the original query and OE_{Hyponym} from the attribute level of Figure 2.9 were defined as 2:1, 2:1 and 3:1, respectively. Meanwhile, the weights were equally assigned to the elements from the factor level of Figure 2.9. This is because they are all the hyponyms of the original query and thus have equal influences for determining the overall relatedness between an expanded query derived from the original query and a document. An example of the implementation of the AHP technique is illustrated in Section 3.4.

2.7.4 Implementing dempster-shafer combination rules for document ranking

Implementing the DS combination rules in the ER algorithm aims to generate a set of values, each of which represents the overall relatedness between an expanded query and a specific document. For the evidence set e_i mentioned in Section 2.7.2, a combined probability assignment that indicates the relatedness between *Fault diagnosis* and a specific document from D_{rele} can be defined as $m^{\Psi}_{e_i}(\Psi \subseteq D_{\mathrm{rele}})$. Furthermore, the remaining belief that is not assigned after commitments to all the documents from D_{rele} is defined as $m_i^{D_{\mathrm{rele}}}$, which is $m_i^{D_{\mathrm{rele}}} = 1 - \sum_{w=1}^{W} m_i^w$. Hence, derived from (2.7.1), the recursive formulas used for determining the overall relevance score between *Fault diagnosis* and D_w is obtained as follows [10]:

$$\begin{aligned}\{D_w\} \ : \ m^w_{e_{i+1}} = & K_{e_{i+1}}(m^w_{e_i} m^w_{i+1} \\ & + m^w_{e_i} m^{D_{\mathrm{rele}}}_{i+1} \ + \ m^{D_{\mathrm{rele}}}_{e_i} m^w_{i+1}),\end{aligned}$$

$$\{D_{\mathrm{rele}}\} \ : \ m^{D_{\mathrm{rele}}}_{e_{i+1}} = K_{e_{i+1}} m^{D_{\mathrm{rele}}}_{e_i} m^{D_{\mathrm{rele}}}_{i+1},$$

where

$$w = 1, \ldots, W, \tag{2.7.7}$$

and

$$K_{e_{i+1}} = \left[1 - \sum_{\tau=1}^{W} \sum_{\rho=1,\ \rho\neq\tau}^{W} m^{\tau}_{e_i} m^{\rho}_{i+1} \right]^{-1},$$

where

$$i=1,\dots,L_j-1. \tag{2.7.8}$$

In the case that a query is composed of more than one keyword, the overall relevance score generated by each of the keywords can be defined as a piece of evidence for the query. Therefore, the final relatedness between such a query and a document can then be derived with the above formulas by combining all the available evidence, with assigned mutual weights.

2.8 Conclusion

Due to the lack of knowledge in document search engines, common users usually cannot state their information request clearly by a submitted query. As a result, the accuracy of a search process may be greatly reduced. Therefore, in the past few years, an ontology-based QE technique has been investigated for dealing with this problem. As illustrated from a set of previous work regarding ODSEs, the accuracy of a document search engine can be enhanced using this technique. However, some drawbacks still exist in ODSEs: firstly, the related terms are added to an original query without considering their hierarchical relationships; Secondly, the relevance scores generated by the terms of an expanded query are treated independently during a combination process. As discussed in Section 1.4.1, these problems can restrain further improvements concerning the search accuracy of a search engine.

Therefore, an ER-based document ranking approach to tackling the problems mentioned above has been presented in this chapter. In the chapter, the literature review of IR was given firstly. Then, the architecture of an ODSE as well as the functions of the ODSE components were introduced, respectively. Subsequently, the development process of SONT was described in detail. The methodology used for organising the terms of an expanded query into a MADM tree model was also presented. The ER algorithm, used for the evidence combination of MADM for generating the relatedness between the expanded query and a specific document, then was demonstrated.

In the next chapter, a number of tests, used for illustrating the practical search performance of the ER-based document ranking approach to an ODSE, are described.

Chapter 3

Implementation Results of The Evidential Reasoning-Based Document Ranking Approach

3.1 Introduction

The experimental work of the ER-based document ranking approach is reported in this chapter. In Section 3.2, three document search engines, which were developed with and without the proposed ER-based approach respectively, are introduced. In Section 3.3, the system configuration of an experiment platform, a document repository and query sets used in tests are illustrated, separately. A search scenario with an ER-based ODSE is demonstrated in Section 3.4. In Section 3.5, evaluation schemes, employed to reveal the practical search performance of the three search engines, are introduced. A number of tests and obtained test results are described in Section 3.6. This chapter is summarised in Section 3.7.

3.2 Three document search engines developed for comparison purposes

Implementing the ER-based approach to an ODSE aims at improving the search accuracy of a document retrieval process, compared with that achieved by an ODSE using the weighted sum algorithm of VSM. Hence, the test work of this study was carried out especially for the comparison of the search performance between two ODSEs with ER and with a weighted sum algorithm. Practically, an ODSE using other relevance combination methods out of only employing a weighted sum algorithm has been also investigated. With one of the methods, a better search accuracy may be obtained by an ODSE, in comparison with that achieved only using a weighted sum algorithm. However, these relevance combination methods are still developed based upon a weighted sum algorithm considering a number of extra parameters, *e.g.*, the number of query terms appearing in a document. In cases where these parameters are considered in both the proposed ER-based approach and a conventional relevance combination method using the weighted sum algorithm of VSM, the comparison work goes back to the difference of search accuracy between these two approaches only.

Table 3.1: Document search engines

	Definition
SE_1	A traditional keyword-matching document search engine
SE_2	An ODSE without the proposed ER-based approach
SE_3	An ODSE with the proposed ER-based approach

In order to illustrate the practical performance of the ER-based document ranking approach, a number of tests were carried out in the *e*-Automation laboratory at the University of Liverpool. As shown in Table 3.1, in total three distinct document search engines, named SE_1, SE_2 and SE_3 respectively, were developed based upon the Apache Lucene search engine library and implemented for the tests. Briefly, SE_1 represents a traditional keyword-matching document search engine with a weighted sum algorithm and does not employ a QE technique. SE_2 and SE_3 are two ODSEs with a weighted sum algorithm and with the implementation of the ER algorithm, respectively. In these three search engines, a relevance score between a query term and a document is automatically generated with Lucene. In other words, a relevance score is determined by the cosine-distance or dot-product between the document and query vectors of VSM [86].

In a Lucene-based document search engine, another main service offered besides document searching is indexing, as briefly mentioned in Section 2.3. The indexing service of Lucene is used to store all useful content of documents existed in a specific document repository. The purposes of building an index server in a document search engine are to reduce search speed and improve search accuracy in document search processes with a submitted query. In the case that an index server is not provided, a search engine needs to scan all the content of the documents stored in a document repository during a document retrieval process. In such a situation, considerable execution time and computing power are required, which should strongly be avoided.

The process of generating an index sever with Lucene, employed in SE_1, SE_2 and SE_3 respectively, is illustrated as Figure 3.1. As can be seen from the figure, such a process firstly employs several document parsers, *e.g.*, the HTML parser, the PDF parser, the Word parser and the Text parser and so on to extract text content from the different types of documents of a document repository. Then, the generated text content is reprocessed by the Lucene analyser and stored as a set of index files to the index server. In the process, the Lucene analyser changes all the text content to lowercase and removes common stop words, *e.g.*, "a", "an" and "in" and so on, from the text content. This is because these stop words are very common and exist in almost all documents. Thus, they are not useful for searching individual documents in a document search process.

With Lucene, documents are decomposed as one or more field objects. Each

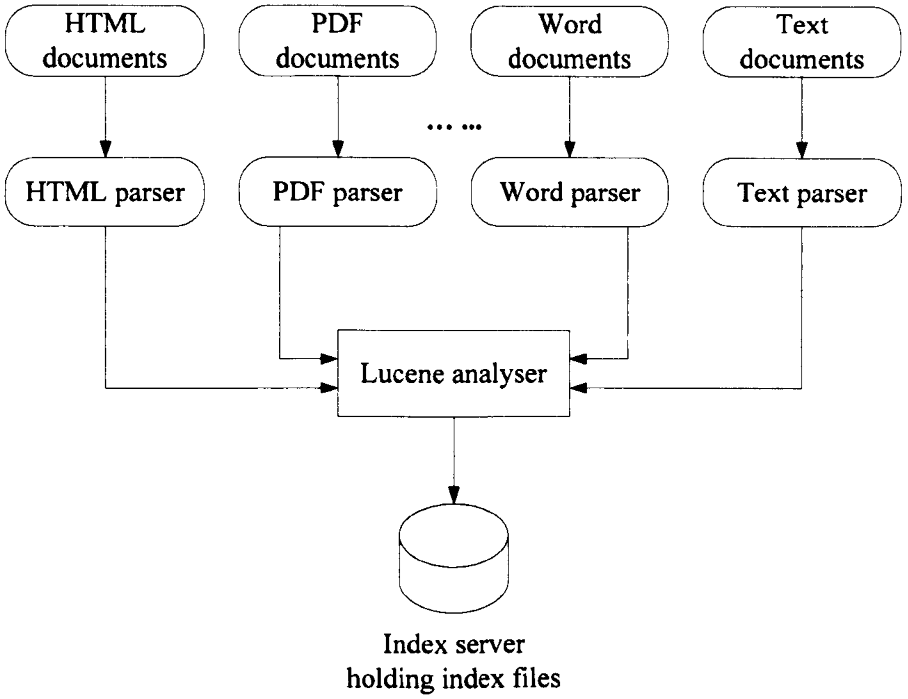


Figure 3.1: An index server generation process with Lucene

field is a name-value pair and used to store a piece of information, *e.g.*, "document name"-"power system automation". In Lucene, the fields of a document can be either indexed or not indexed, *i.e.*, can be either employed for searching documents and displaying a search result, or only displaying a search result. For instance, the name of a document is stored as a field and indexed by Lucene. Then, the document name can be utilised in a document search process and consequently displayed as a piece of information in a search result. Meanwhile, the unique document identifier of the document, used for distinguishing the document from the other documents of a document repository, can be also saved as a field. However, this field is not useful for searching and thus not indexed. In a document search process, the field is only used as a piece of information when displaying a search result.

The index server used in this study was generated with Lucene according to

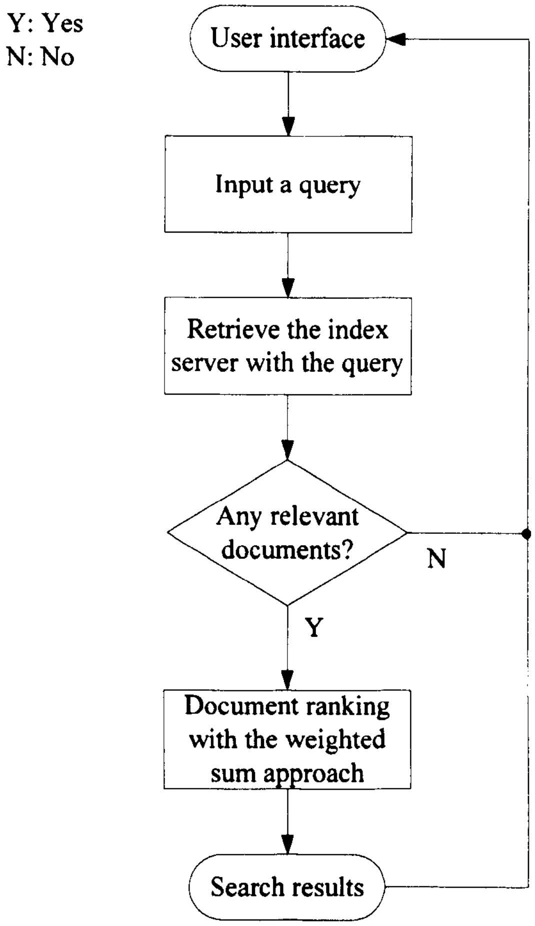

Y: Yes
N: No
User interface
Input a query
Retrieve the index server with the query
Any relevant documents?
N
Y
Document ranking with the weighted sum approach
Search results

Figure 3.2: A SE_1 working process

the above index server generation procedures. With the generated index server, the three search engines then can be implemented for IR of power substations. The working processes of SE_1, SE_2 and SE_3 are illustrated as Figure 3.2, Figure 3.3 and Figure 3.4 respectively and described as follows:

- SE_1: As shown in Figure 3.2, in the search process of SE_1, the relevant documents of a submitted query are firstly identified. Then, the relevance score between the query and a relevant document is computed by the weighted sum of the relevance scores, generated between the terms of the

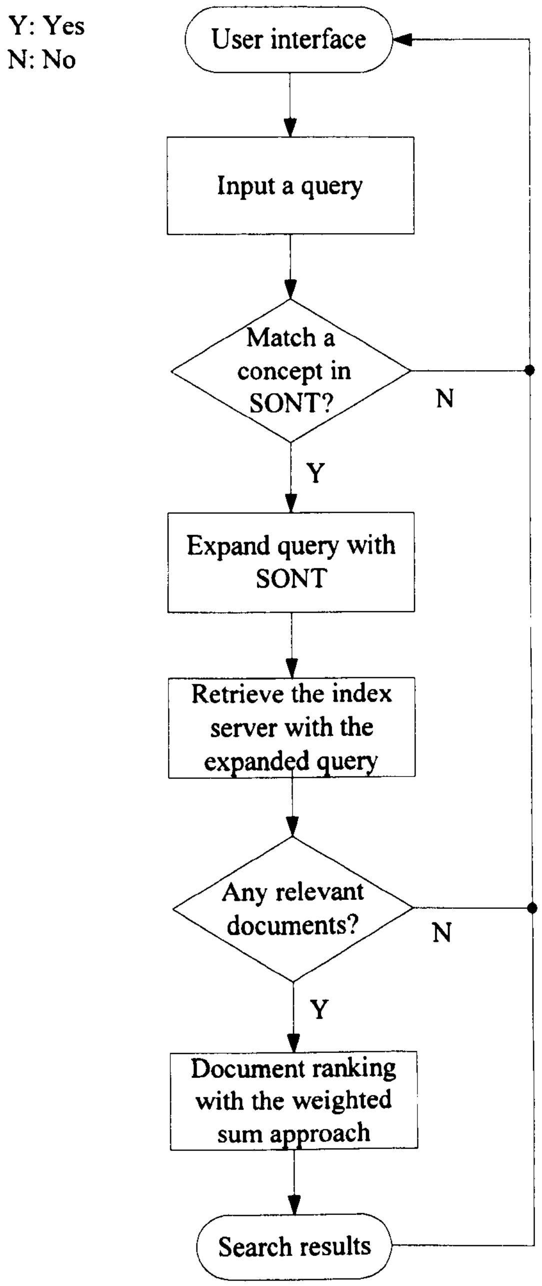


Figure 3.3: A SE_2 working process

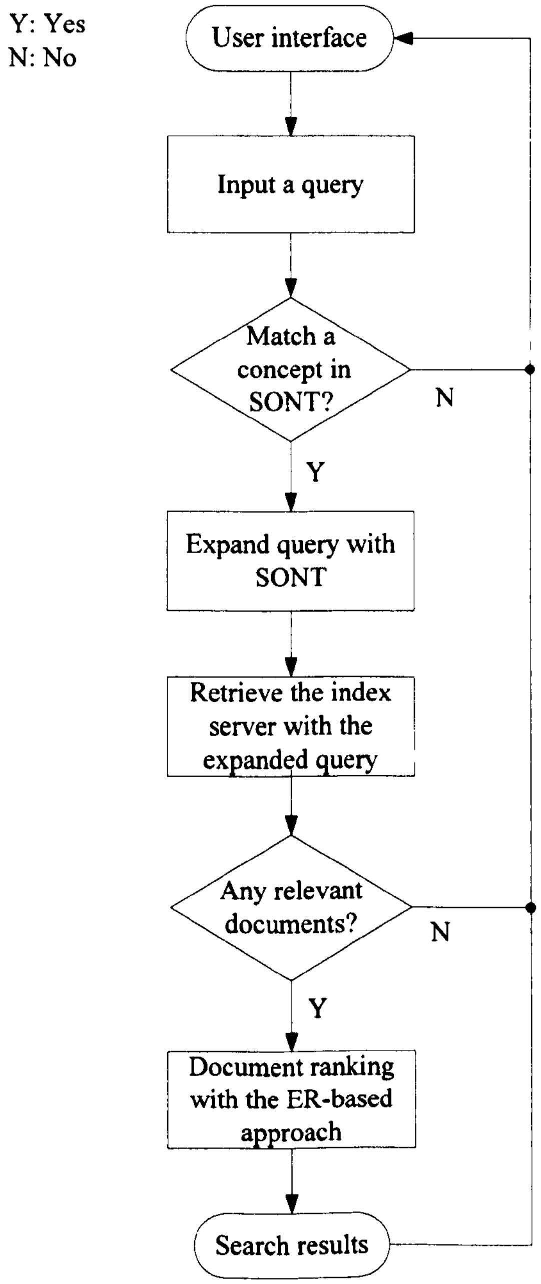


Figure 3.4: A SE_3 working process

query and the document. In this calculation process, mutual weights are equally assigned to the terms of the query, if multiple terms exist in the query.

- SE_2: The search process of SE_2 is illustrated as Figure 3.3. In the process, the query terms of a query are expanded with SONT firstly, if any of them matches a concept defined in SONT. Subsequently, the relevant documents of the expanded query are obtained by SE_2. Afterwards, the relevance score between the expanded query and a relevant document is computed by the weighted sum of the relevance scores, generated between the query terms of the expanded query and the document. The mutual importance values between the relevance scores generated by an original query term and a synonym, the synonym and a hyponym, the original query and the hyponym are defined as 2:1, 2:1 and 3:1, respectively. Then, the weights assigned to the generated relevance scores during a search process can be derived using AHP discussed in Section 2.7.3.

- SE_3: The process of generating an overall relevance score between a query input and a document with SE_3 is described as Figure 3.4, which can be summarised into following steps: first of all, the query terms of a query are expanded with its synonyms and hyponyms using SONT, if any of them matches a concept defined in SONT. In the next step, the relevant documents of the expanded query are achieved with SE_3. Finally, the ER algorithm is implemented to combine the relevance scores between the expanded query and the relevant documents, by considering the mutual weights assigned to a pair of evidence explained in Section 2.7.3. Also, as mentioned in Section 2.7.4, in the case that an original query is composed of more than one keyword, the overall relevance score obtained by each of the keywords can be defined as a piece of evidence for the original query. The final relatedness between the query and a document then is computed with the ER algorithm by combining all these evidence, with equally assigned mutual weights.

In a search process with any of the three search engines, all the relevant documents regarding a query are ranked from the largest relevance score to the smallest relevance score, when the corresponding relevance scores are generated. Then, the ranked document list is returned to users as a search result. The details of the tests are demonstrated in the following subsections.

3.3 Document repository and query sets

For each of the three search engines in Table 3.1, two query sets involving a unique-keyword query set and a combined-keyword query set were utilised for checking its practical search performance. Final results are reported in the following content of this chapter. All the experiments were undertaken using the same computer, with an Intel Pentium 4 2.80 Gigahertz (GHz) Central Processing Unit (CPU), a 1.00 Gigabyte (GB) Random-Access Memory (RAM), a 80 GB hard disk and a Windows XP Professional Sever Pack 2 (SP2) operating system.

Totally 136,735 documents (all in English) were involved throughout the tests, a part of which were collected concerning power substations, including published academic papers, emails of power companies, technical reports and massive maintenance records and so on.

Table 3.2: Statistics of the document sets utilised in the tests

Number of documents	136,735
Unique-keyword queries	10
Combined-keyword queries	10
Average number of documents per document pool	86.3
Average number of relevant documents per document pool	32.5

The basic statistics regarding the tests are listed in Table 3.2. Two sets of queries were selected by considering the widely used query terms related

Table 3.3: Queries utilised for the tests

Unique-keyword query set	Combined-keyword query set
1. Substation	1. Power system + Frequency response analysis
2. Transformer	2. Winding + Distortion analysis
3. Coolant	3. Relay + Fault location + Power system
4. Circuit breaker	4. Harmonics + Distortion + Power quality
5. Fault diagnosis	5. Transmission line + Protection + Relay
6. Dissolved gas analysis	6. Temperature + Overload + Thermal modelling
7. Relay	7. Transformer + Dissolved gas analysis
8. Switch	8. Fault analysis + Partial discharge
9. Thermal model	9. Transformer + Vibration + Mechanic
10. Voltage	10. Bus bar + Protection

to a power substation. As illustrated in Table 3.3, while the first set was composed of 10 unique-keyword queries, another 10 combined-keyword queries were included in the second set.

It should be noted that, all the documents used in the tests were named differently, so that each of them could be considered as a distinct individual. In the next section, a simple search process with the ER-based ODSE, *i.e.*, SE_3, is illustrated.

3.4 A simple search scenario with the proposed document search engine

In order to get a better understanding on how to implement the proposed ER-based approach in an ODSE, a document search scenario with SE_3, which was implemented in the tests, is presented in this section. Again, the query

Table 3.4: Relevance scores in the factor level

	Fault detection	*Fault detections*	*Fault isolation*	*Fault isolations*
D_a	0.0000	0.0000	0.0000	0.0000
D_b	0.0202	0.0000	0.0265	0.0000

Table 3.5: Relevance scores in the factor level after normalisation

	Fault detection	*Fault detections*	*Fault isolation*	*Fault isolations*
D_a	0.0000/24933	0.0000/24933	0.0000/24933	0.0000/24933
D_b	0.0202/24933	0.0000/24933	0.0265/24933	0.0000/24933

Fault diagnosis is utilised here for an illustration purpose. In this search process with SE_3, first of all, the submitted query *Fault diagnosis* was expanded with its synonyms and hyponyms, expressed as (2.6.1) and (2.6.2), respectively. Then, totally 24,933 documents from the test document set were identified as the relevant documents of the expanded query with the Boolean model of Lucene, as introduced in Section 2.6.3. Subsequently, a search result was generated according to the working procedures of SE_3, as introduced in Section 3.2.

The method used for ranking two different documents in this search process is described as follows. Supposing that, two relevant documents of the expanded query of *Fault diagnosis* were randomly selected from the test document set and marked as D_a and D_b, respectively. Their overall relevance scores regarding *Fault diagnosis* were defined as RS_a and RS_b, separately.

As stated in Section 2.6, the overall relevance score RS_D between the query *Fault diagnosis* and a document is determined by the combination of the relevance scores of itself, its synonyms and hyponyms. These relevance scores are treated as a set of evidence concerning the generation of the overall relevance score RS_D. Based on the MADM tree model illustrated in Figure 2.9, the outputs of the four hyponym branches can be treated as four pieces of evidence for computing the value of OE_{Hyponym}.

The relevance scores obtained by the four hyponyms of *Fault diagnosis* regarding D_{a} and D_{b} are shown in Table 3.4. Then, the normalised relevance scores, calculated with equation (2.7.3) are presented in Table 3.5. According to equation (2.7.4), in the search process, these relevance scores were further treated as the confidence degrees $\beta_{D_w}(e_i)$. The relative weights λ_i were assigned as the same value 1/4 to each of them, as mentioned in Section 2.7.3. Therefore, for example, the probability assignments m_i^w confirmed by *Fault detection* regarding D_{b} was generated as 0.0202/24933*1/4=0.0202/99732. When all the probability assignments were obtained for D_{a} and D_{b}, the values of OE_{Hyponym} were obtained as 0.0000 and 4.6825E-7 with equations (2.7.7) and (2.7.8) for D_{a} and D_{b} respectively. The output values of the evidence from the attribute level of Figure 2.9 are listed in Table 3.6, when normalised by equation (2.7.3).

In order to integrate these heterogeneous relevance scores of Table 3.6 into scaled inputs for generating RS_{a} and RS_{b}, the relative weights λ_i of them were derived using AHP discussed Section 2.7.3. As stated in Section 2.7.3, in the tests, the mutual importance values between an original query and a synonym, the synonym and OE_{Hyponym}, the original query and OE_{Hyponym} were defined as 2:1, 2:1 and 3:1, respectively. Therefore, the matrix (2.7.5) introduced in Section 2.7.3 can be expressed as matrix (3.4.1).

$$\begin{bmatrix} 1 & 2 & 2 & 2 & 3 & 2 & 2 & 2 & 2 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/3 & 1/2 & 1/2 & 1/2 & 1 & 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \end{bmatrix} . \tag{3.4.1}$$

With the AHP algorithm, the relative weight set of λ_i was obtained as [0.2052, 0.1057, 0.1057, 0.1057, 0.0548, 0.1057, 0.1057, 0.1057, 0.1057]$^{\text{T}}$. Finally, RS_{a} and RS_{b} were calculated as 3.1123E-6 and 4.3971E-6 with equations (2.7.7) and (2.7.8), respectively. Compared with the relevance score 3.1123E-6 of D_{a}, apparently, D_{b} has obtained a higher relatedness to the query *Fault diagnosis* with the relevance score 4.3971E-6. Consequently, D_{b} was returned with a higher ranking order in the final search result, *i.e.*, a relevant document list, compared with that of D_{a}. The document ranking scheme illustrated above can be implemented to rank more than two documents in a similar way with a provided query.

Table 3.6: Relevance scores in the attribute level

	Fault diagnosis	*Fault diagnoses*	*Condition assessment*	*Condition assessments*	OE_{Hyponym}
D_{a}	0.3782/24933	0.0000/24933	0.0000/24933	0.0000/24933	0.0000
D_{b}	0.5021/24933	0.0202/24933	0.0000/24933	0.0000/24933	4.6825E-7
	Fault location	*Fault locations*	*Fault analysis*	*Fault analyses*	
D_{a}	0.0000/24933	0.0000/24933	0.0000/24933	0.0000/24933	
D_{b}	0.0362/24933	0.0000/24933	0.0000/24933	0.0000/24933	

Practically, in the search processes with SE_1 and SE_2, the relevance scores generated between each term of an expanded query and a document are computed with the weighted sum of these relevance scores, as introduced in Section 3.2. In the next section, a recall and precision curve method, used for verifying the search performance of all the three search engines, *i.e.*, SE_1, SE_2 and SE_3, is described. A pooling method used in the study is later introduced as well.

3.5 Performance evaluation schemes

Typically, recall and precision are considered the most important performance indices employed for evaluating the effectiveness of a document search engine. Thus, the method of measuring recall and precision curve [42] [13] was selected in the tests for verifying the search performance of SE_1, SE_2 and SE_3. The recall and precision curve evaluation method is explained as below.

Given a query, all its relevant documents in a document repository are defined as a set R. A set of documents H is obtained by a document search engine after performing a search process. Therefore, the recall and precision can be defined as follows:

$$R_c = \frac{H \cap R}{R}, \tag{3.5.1}$$

where R_c is the recall value, and

$$P_r = \frac{H \cap R}{H}, \tag{3.5.2}$$

where P_r expresses the precision of a search process.

In order to obtain the set R for each of the queries in Table 3.3, a pooling method introduced in [87] was implemented to undertake a task such as this. Each of the 20 queries shown in Table 3.3, SE_1, SE_2 and SE_3 was utilised to implement a search process with the provided document repository respectively. The top 50 ranked documents of each search process were then pooled into a document set, defined as a document pool, for the corresponding query. Then, by eliminating the reduplicate items in each document pool, the average number of documents of the 20 document pools was derived as 86.3, as shown in Table 3.2.

For each of the 20 document pools, the documents were classified as *relevant* and *non-relevant* by power system engineers regarding a corresponding query. As a result, in the tests, R for a specific query was confirmed as the *relevant* document set. Moreover, the mean of relevant document numbers in all *relevant* document sets for the total 20 queries was obtained as 32.5, as illustrated in Table 3.2.

In practice, implementing such a pooling method can avoid evaluating all the documents in the document repository for a specific query. More significantly, with this method, R for a query is determined in a logical way and thus may lead to a more accurate test result. Since the method does not guarantee that all the relevant documents of a query can be found and located in the corresponding document pool, a recall value obtained in a search process is then defined as a *relative recall.*

With each of the two query sets from Table 3.3, the average precisions of SE_1, SE_2 and SE_3 were calculated at 10 different recall levels separately, from 10%, 20% to 100%. The test results are illustrated in Section 3.6 followed by a detailed discussion.

3.6 Test results and discussion

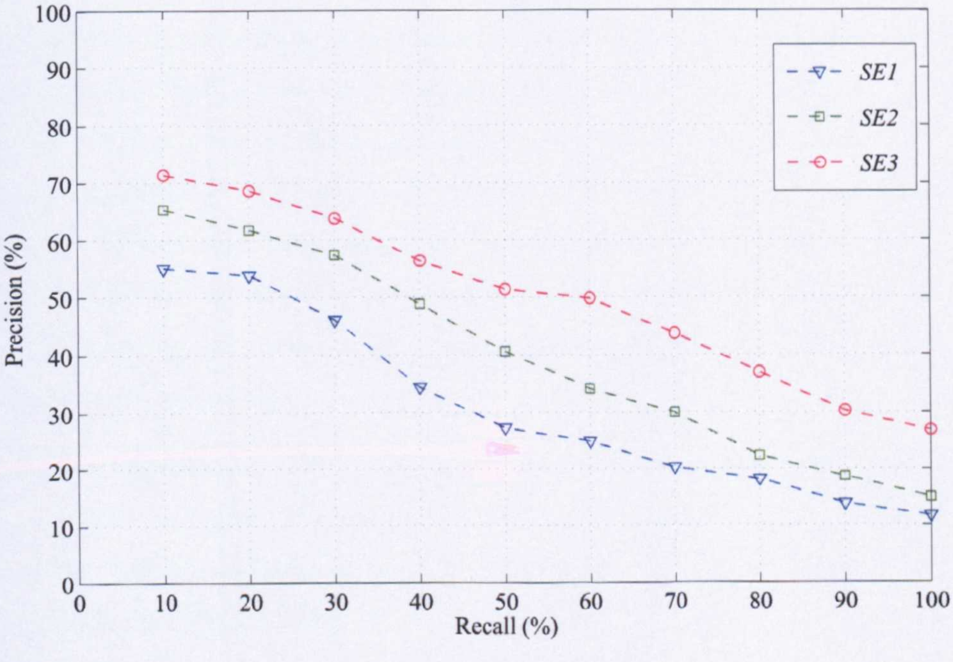


Figure 3.5: Average precision-recall curves for three search engines with 10 unique-keyword queries

Table 3.7: Average precisions (%) of 3 search engines at 10 recall levels obtained using 10 unique-keyword queries

Recall (%)	10	20	30	40	50
SE_1	55.0	53.8	46.1	34.4	27.3
SE_2	65.3	61.7	57.4	48.8	40.6
SE_3	**71.4**	68.6	63.7	56.3	51.3
Recall (%)	60	70	80	90	100
SE_1	24.9	20.2	18.1	13.8	11.6
SE_2	34.2	29.9	22.3	18.6	14.9
SE_3	49.8	43.7	36.9	30.2	26.8

In the study, a number of tests were undertaken with the test schemes introduced above. The results are illustrated in this section along with a discussion. Figure 3.5 shows the average precision generated with the 10 unique-keyword queries at different recall levels (from 10%, 20% to 100%). The corresponding precision values illustrated by these curves are then presented in Table 3.7. As shown in this table, SE_1 has obtained the lowest precision throughout the search processes compared with the other two search engines. This may be due to the following reasons:

1. The synonyms of the original queries were not considered within the search processes. Therefore, the documents, related to the user's interests but not containing the same words as the original queries in grapheme, could not be retrieved by SE_1;

2. The hyponyms of the original queries may have influenced the search performance as well, which means that the less consideration of the subclasses may reduce the search accuracy in many cases.

On the other hand, as can be viewed from Table 3.7, the average precision of SE_3 was higher than those of SE_2 at all the recall levels. This indicates that,

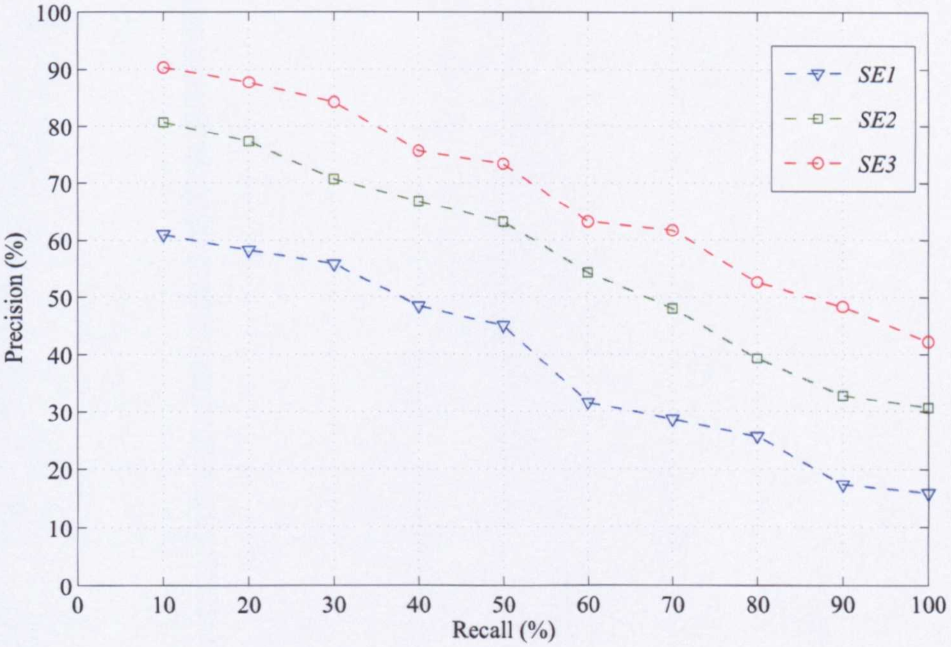


Figure 3.6: Average precision-recall curves for three search engines with 10 combined-keyword queries

although the accuracy of a keyword-matching search engine can be improved with a QE technique provided by a domain ontology model, a rough-and-tumble organisation of the relevance scores, generated by the expansion terms, can restrict the search precision. Therefore, the potential of the proposed ER-based approach to improving the search accuracy of an ODSE has been verified to some extent with the unique-keyword queries.

Figure 3.6 presents the average recall-precision curves for the 10 combined-keyword queries, which were generated by SE_1, SE_2 and SE_3, respectively. The precision values indicated by the curves are described in Table 3.8. Compared with the search accuracy of each search engine at the same recall level shown in Figure 3.5, the precision values of the three search engines displayed

Table 3.8: Average precisions (%) of 3 search engines at 10 recall levels obtained using 10 combined-keyword queries

Recall (%)	10	20	30	40	50
SE_1	61.0	58.2	55.8	48.5	45.0
SE_2	80.7	77.3	70.6	66.8	63.2
SE_3	**90.3**	87.7	84.2	75.6	73.3
Recall (%)	60	70	80	90	100
SE_1	31.5	28.6	25.7	17.3	15.9
SE_2	54.3	48.1	39.3	32.7	30.6
SE_3	63.2	61.7	52.6	48.3	42.2

in Figure 3.6 have been improved. This means that utilising multi-keyword in one search process can refine the search scale, and thus can improve the accuracy of document search engine outputs. Among the three search engines, SE_3 has delivered the best accuracy at every recall level again, while the accuracy achieved by SE_2 was much higher than that of SE_1.

Therefore, for both the unique-keyword query set and the combined-keyword query set, the ER-based document ranking approach has demonstrated its capability of improving the search accuracy of an ODSE.

Finally, the average recall-precision curves of all the 20 queries, generated by SE_1, SE_2 and SE_3 respectively, are shown in Figure 3.7 and the precision values of these curves are presented in Table 3.9. Conventionally, a high precision is the most important when it is generated at a low recall level. Therefore, as shown in Table 3.9, at the recall level of 10%, the highest precision 80.9% of the three search engines was obtained by SE_3. On the other hand, the precision values of the three search engines were derived as 78.2%, 69.5% and 56.0% respectively at the recall level of 20%. The above results clearly show that, in an ODSE, the ER algorithm has provided a suitable solution for combining multiple relevance scores of an expanded query. More significantly, the search

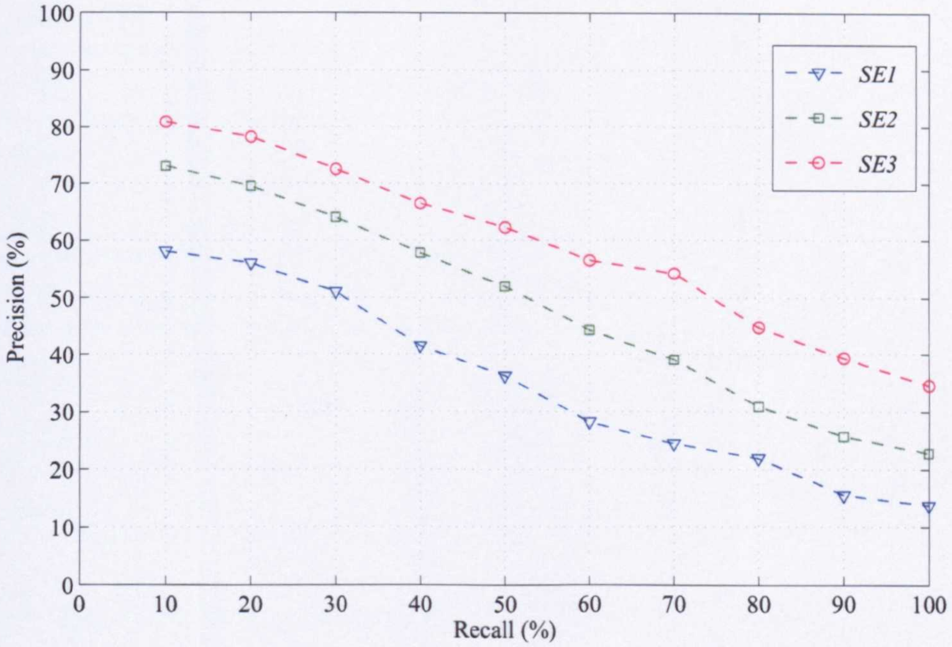


Figure 3.7: Average precision-recall curves for three search engines with all 20 queries

accuracy of an ODSE can be improved with the ER-based document ranking approach.

On the other hand, the average time consumptions of a search process with SE_1, SE_2 and SE_3 were recorded. The results show that, the time consumption of SE_3 is similar to that consumed by SE_1 and SE_2 in most cases.

3.7 Conclusion

The experimental work of the ER-based document ranking approach has been described in this chapter. The working flows of the three search engines

Table 3.9: Average precisions (%) of 3 search engines at 10 recall levels obtained using all 20 queries

Recall (%)	10	20	30	40	50
SE_1	58.0	56.0	51.0	41.5	36.2
SE_2	73.0	69.5	64.0	57.8	51.9
SE_3	**80.9**	**78.2**	72.5	66.5	62.3
Recall (%)	**60**	**70**	**80**	**90**	**100**
SE_1	28.2	24.4	21.9	15.6	13.8
SE_2	44.3	39.0	30.8	25.7	22.8
SE_3	56.5	54.2	44.8	39.3	34.5

were introduced firstly. Then, the test schemes were presented. In order to illustrate the search process of SE_3 clearly, a simple search scenario of SE_3, was depicted. Subsequently, a traditional keyword-matching search engine, two ODSEs with and without the ER algorithm were tested with 10 unique-keyword queries and 10 combined-keyword queries, respectively. The final results showed that the ER-based ODSE, *i.e.*, SE_3, has achieved the highest search accuracy. Therefore, the ER-based approach can be proposed as a viable solution for ranking documents in an ODSE.

Chapter 4

Association Rule Mining-Based Dissolved Gas Analysis to Fault Diagnosis of Power Transformers

4.1 Introduction

In order to deal with the limitation of the conventional DGA methods introduced in Section 1.3.2, various AI techniques have been implemented for FD of power transformers, *e.g.,* ES, FL, ANN, SVM and KNN and so on. In this chapter, an alternative AI approach to DGA namely ARM-based DGA approach, is presented for FD of power transformers, which can overcome the drawbacks of the conventional DGA methods and thus improve the accuracy of a transformer FD process.

Briefly, in an ARM process for DGA, a set of DGA records is selected for generating association rules, each of which is composed of several key gas concentrations and a working state derived by on-site inspections. In order to choose suitable gas concentrations for the ARM process, the gas concentrations employed in a conventional DGA method, *i.e.*, Dornenburg or Rogers, are selected, as these gases are chemical reaction products when transformer faults occur and thus have potential connections with the faults. In other words, each of the DGA records is converted into a new format, which employs the gas ratios used in a conventional DGA method to reorganise the key gas concentrations of the record. Afterwards, the gas concentrations of the DGA record, not employed in the reorganisation process, are eliminated. As a result, a new data set can be derived from the original DGA records. Subsequently, the new data set is categorised into two parts, *i.e.*, training and test data sets, used for association rule generation and the FD performance verification of a constructed association rule-based FD system, respectively. In the next step, with the training data set, association rules are generated with an ARM algorithm concerning user-specified attributes. In order to construct a transformer FD system with useful association rules, a rule set simplification method and a rule fitness evaluation method are used to eliminate the useless rules of the generated rule set and assign a fitness value to each of the remaining useful rules. Subsequently, the FD system is constructed based upon the useful association rules. Finally, the FD system is implemented for transformer FD with an optimal rule selection method, which chooses the most accurate rule for a diagnosis task from the useful association rules regarding their assigned fitness values.

The rest of this chapter is organised as follows: In Section 4.2, the literature review of DGA is given as the development basis of the ARM-based DGA approach, which involves a number of traditional DGA methods. Then, several conventional AI-based DGA classifiers are presented. Subsequently, the ARM technique is briefly introduced in Section 4.3. In Section 4.4, a practical ARM process for generating a set of useful association rules of DGA is illustrated. In the section, firstly, two methods, *i.e.*, an attribute selection method and a continuous datum attribute discretisation method, are presented for choosing the user-interested attributes used in an ARM process. Subsequently, an ARM algorithm namely Apriori-Total From Partial (TFP) [88] is introduced for generating an Association Rule Set (ARS) with the determined attributes. Finally, a rule set simplification method and a rule fitness evaluation method, used for selecting useful classification rules from the obtained ARS, are described. Section 4.5 presents an implementation strategy of a constructed association rule-based FD system. In that section, an optimal rule selection method is discussed. This chapter then is concluded in Section 4.6.

4.2 Traditional solutions of dissolved gas analysis

4.2.1 Fundamentals of dissolved gas analysis

As mentioned in Section 1.3.2, in the past few years, various criteria for DGA have been developed, such as the Dornenburg, Rogers, Key Gas methods. As explained previously, in order to choose suitable gas concentrations for an ARM process, a provided DGA record needs to be converted into a new format, which employs the gas ratios used in a conventional DGA method to reorganise the key gas concentrations of the record. In the tests of the ARM-based DGA approach, for comparison purposes, the gas ratios of Dornenburg and Rogers were used to reorganise the key gas concentrations of a DGA record, respectively. Hence, in this section, a detailed introduction to Dornenburg and Rogers is provided for making a clear understanding of the proposed ARM-based DGA research work. The review of the Key Gas method is out of the research scope of the thesis and thus not addressed. A detailed discussion of this method can be viewed from [1].

The working processes of Dornenburg and Rogers are briefly described in Figure 4.1. In a DGA process with Dornenburg or Rogers, the use of key gas ratios for diagnosing a possible transformer fault is mainly implemented with a set of diagnosis rules, which are developed based upon the experience of power engineers. As can be observed from this figure, in Dornenburg and Rogers, in total five key gas ratios are employed: Ratio 1 (Rl) = CH_4/ H_2, Ratio 2 (R2) = C_2H_2/C_2H_4, Ratio 3 (R3) = C_2H_2/CH_4, Ratio 4 (R4) = C_2H_6/C_2H_2, Ratio

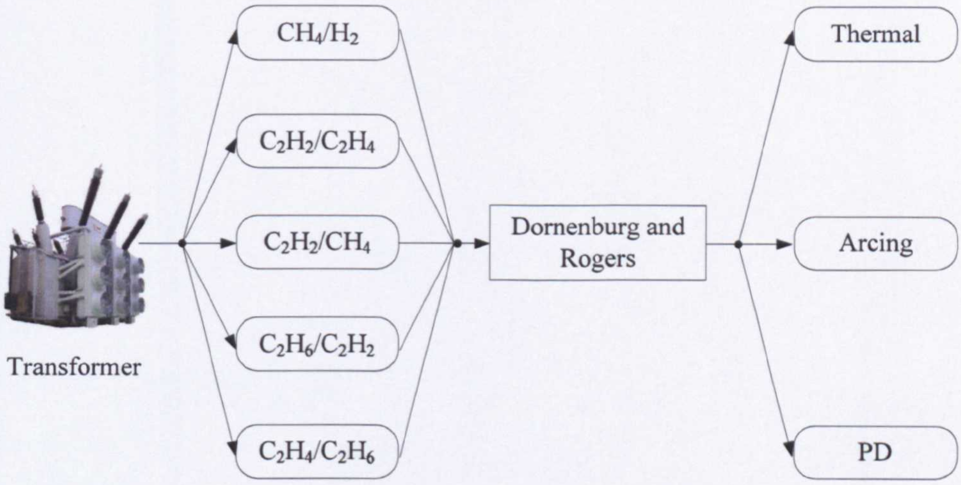


Figure 4.1: DGA processes of Dornenburg and Rogers

5 (R5) = C_2H_4/C_2H_6. In Dornenburg, the first four gas ratios, *i.e.*, R1, R2, R3 and R4, are utilised. On the other hand, three gas ratios, *i.e.*, R1, R2 and R5 are employed in a transformer FD process of Rogers. The possible fault types involved in these two DGA methods mainly are thermal, arcing and PD. In the following two subsections, the FD processes, using Dornenburg and Rogers respectively, are described in detail.

Dornenburg ratio method

The detailed work flow chart of the Dornenburg ratio method is illustrated in Figure 4.2. As shown in the figure, four possible transformer working states can be diagnosed by a DGA process with Dornenburg, including the three possible fault types mentioned above and no fault. In an FD process with a given test DGA record, each value of the six key gas concentrations, *i.e.*, H_2, CH_4, CO, C_2H_2, C_2H_4 and C_2H_6, is firstly compared with a predefined concentration value, *i.e.*, $L1$ in parts per million (ppm), as indicated in Table 4.1. This procedure aims to ascertain whether there is possibly an existing fault before taking a further FD process. As shown in Figure 4.2, when the conditions are satisfied, *i.e.*, the concentration of H_2 or CH_4 or C_2H_2 or C_2H_4>$2L1$ as

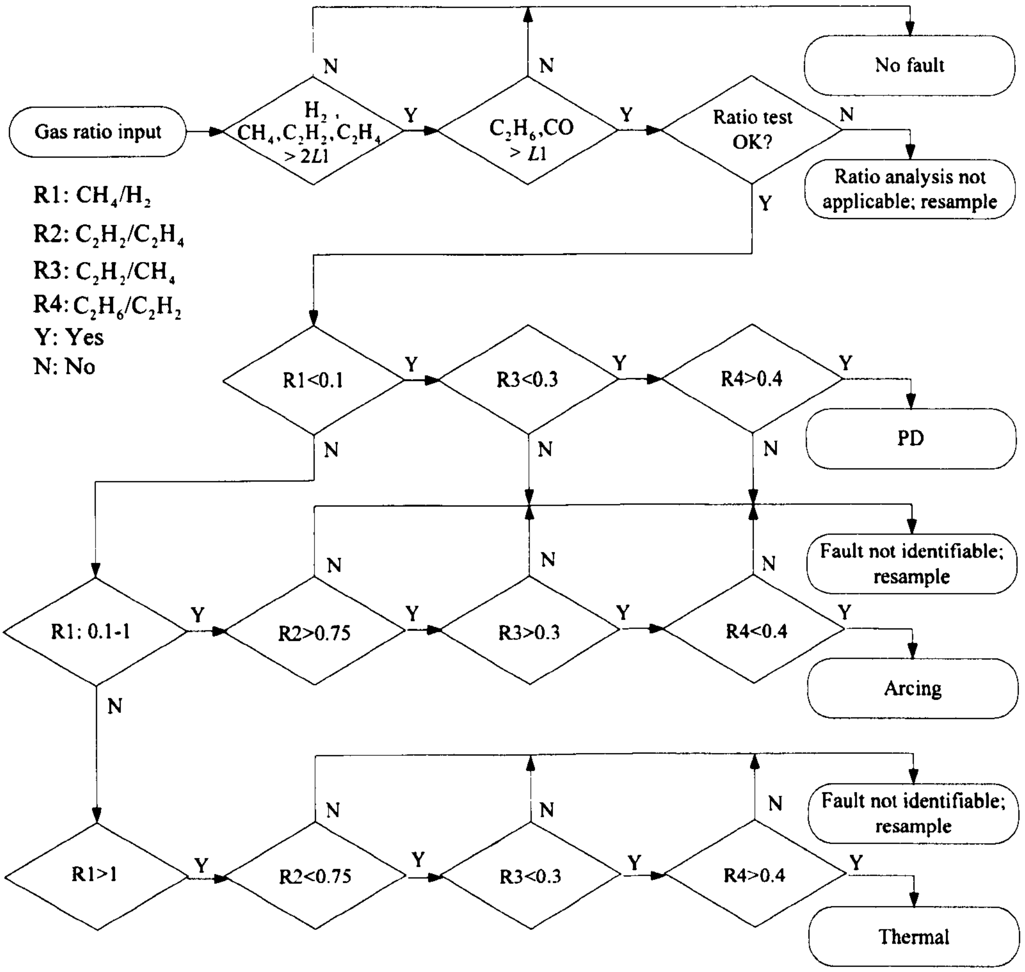


Figure 4.2: Dornenburg ratio method flow chart [1]

well as the concentration of C_2H_6 or CO>$L1$, a transformer fault might occur. Consequently, an FD process with Dornenburg is implemented.

Table 4.1: Dornenburg's $L1$ limits (reprinted from [1])

Gas	H_2	CH_4	CO	C_2H_2	C_2H_4	C_2H_6
L1 (ppm)	100	120	350	35	50	65

Table 4.2: Dornenburg ratio method (reprinted from [1])

		R1	R2	R3	R4
	Ratios of gases	< 0.1 (0) 0.1-1.0 (1) > 1.0 (2)	< 0.75 (0) > 0.75 (1)	< 0.3 (0) > 0.3 (1)	< 0.4 (0) > 0.4 (1)
Cases	Characteristic fault				
1	Thermal	2	0	0	1
2	PD	0	N/A	0	1
3	Arcing	1	1	1	0

The rules used in the FD processes of Figure 4.2 can be summarised as Table 4.2. In an FD process with Dornenburg, the key gas ratios, *i.e.*, R1, R2, R3 and R4, of a test DGA record are calculated based upon the key gas concentrations of the record firstly. Subsequently, the obtained four gas ratios are ranked in the order from R1 to R4. Then, each of the gas ratios is compared with the predefined value limits of the three rules from Table 4.2. If all the predefined value limits of a rule are satisfied by the gas ratios of the provided DGA record, the record is then assigned with the corresponding fault type of the rule.

Rogers ratio method

The Rogers ratio method of DGA is another DGA criterion that is used to identify a possible fault of a power transformer with the dissolved gas concentrations extracted from transformer oil. Being different from Dornenburg, the Rogers ratio method does not check the specific gas concentration values, *i.e.*, the $L1$ values defined in Table 4.1, of a transformer being diagnosed for ascertaining whether a subsequent diagnosis process is valid or not. However, it suggests that an FD result achieved by Rogers is more reliable when each of the concentration values of the employed key gases is greater than 10 times of a detection limit, compared with that obtained when any of the key gas concentration values is less than 10 times of the corresponding detection limit. Hence, a Rogers DGA process is normally taken when all the involved key gas concentrations are greater than 10 times of their detection limits, which are illustrated in Table 4.3. This checking procedure ensures that the influence of instrument inaccuracy on a diagnosis result can be minimised.

Table 4.3: Rogers' DGA detection limits (reprinted from [2])

Gas	C_2H_2	C_2H_4	CH_4	H_2	C_2H_6
Detection limit (ppm)	1	1	1	5	1

As stated above, only three key gas ratios are employed in an FD process with Rogers, *i.e.*, R1, R2 and R5. The detailed work flow chart of Rogers is shown in Figure 4.3. The rules employed in the FD processes of Figure 4.3 can be summarised as Table 4.4. In an FD process using Rogers, first of all, the values of R1, R2 and R5 are computed with the key gas concentrations of a given DGA record. Then, the rules defined in Table 4.4 are applied for diagnosing the record. Such a diagnosis process is similar to that of Dornenburg as introduced above. It should be noticed that the fault types shown in Table 4.4 are the most common ones of Rogers and derived by combining a number of possible fault types, originally defined by Rogers.

In the next section, a number of AI attempts of DGA are reviewed, *e.g.*,

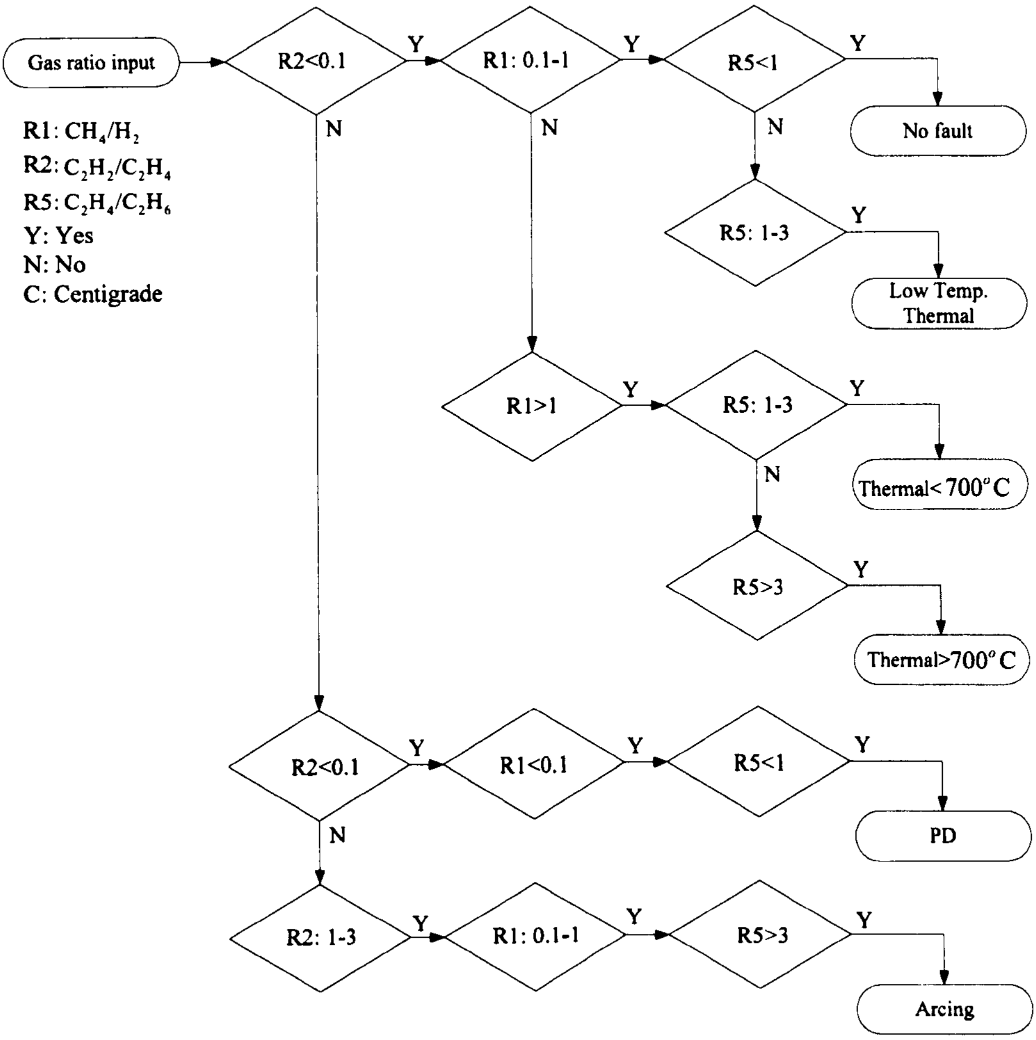


Figure 4.3: Rogers ratio method flow chart [1]

Table 4.4: Rogers ratio method (reprinted from [1])

		R1	R2	R5
	Ratios of gases			
	< 0.1	1	0	0
	0.1-1.0	0	1	0
	1.0-3.0	2	1	1
	> 3.0	2	2	2
Cases	Characteristic fault			
0	No fault	0	0	0
1	PD	1	0	0
2	Arcing	0	1	2
3	Low Temp. Thermal	0	0	1
4	Thermal< 700°C	2	0	1
5	Thermal> 700°C	2	0	2

ES, FL, ANN, SVM and *K*NN and so on. In the experiments of this study, ANN, SVM and *K*NN were tested with provided training and test data sets using ranges of classifier parameters. Then, the obtained FD accuracies were compared with that of the proposed ARM-based DGA approach.

4.2.2 Conventional artificial intelligence methods for dissolved gas analysis

The aforementioned DGA methods are computationally straightforward and usually used as the general guideline of transformer FD by power engineers. However, drawbacks still exist in these conventional DGA methods, as mentioned in Section 1.4.2. In order to deal with these problems, based upon the gas content extracted from transformer oil samples, various AI techniques have been developed. The involved methodologies are ES, FL, ANN, SVM and KNN and so on. In these methods, normally, the fault types of a set of DGA records are firstly classified by power engineers, based upon the integrated criteria of on-site inspections, total combustible gases analyses, the evaluation of gas generation rates and consulting the various gas methods. Then, the connections between the fault types and the key gas concentrations and/or ratios of the DGA records are discovered by utilising various mathematical algorithms. This is usually called a training process. Then, with the obtained criteria from the training process, a test DGA record can be diagnosed.

Expert systems in DGA

ES is a very popular tool used for DGA, which is designed to mirror the behaviours of DGA engineers, normally by representing their knowledge into a set of decision rules. In a DGA approach using ES, the conventional DGA criteria, *e.g.*, the Dornengburg, Rogers and Key Gas methods, and the experience of human expertise are taken into account to form a decision making system. Information regarding power transformers, such as manufacturers, the volume of oil, transformer sizes and previous diagnostic results may be also considered in the process.

ES is normally operated in a top-down sequence: new test DGA data, input from the top of the system, are processed into a form that can be analysed by the system firstly. Then, suitable IF-THEN decision rules are selected from the system for processing the provided data. Finally, obtained knowledge can be transferred back to the system for the corresponding updates of the system knowledge.

On the other hand, ES combined with other AI techniques, *e.g.*, FL and Evolutionary Algorithm (EA), also have been developed for DGA [27] [89]. FL is based on Fuzzy Set Theory (FST) which was formalised by Zadeh at the University of California in 1965 [90]. Similar as ES, FL takes the conventional DGA interpretation criteria and the experience of human expertise to form a decision making system with IF-THEN rules. However, "fuzziness" is allowed in FL to assign a credibility, in the interval [0,1], to a rule which measures the correctness confidence of the conclusion derived using the rule. EA is capable of taking an automatic knowledge acquiring process from a training data set concerning the previous diagnosis history of power transformers. With different training data, extracted knowledge may differ. In some cases, some FD knowledge which may still be unknown to power engineers can be discovered by EA.

Artificial neural network

The earliest work of ANN was reported by McCulloch and Pitts in the 1940's [28]. In an FD process with an ANN classifier, a "no decision" conclusion can be significantly avoided and replaced by providing a possible working state with an assigned probability value. This function thus makes ANN as one of the most widely used fault classifiers in DGA. Similar to a biological neuron system, ANN is a computational system with a large number of simultaneously functioning simple processes having many connections. ANN renders organisational principles similar to a human brain aiming to learn abilities. With regard to ANN, its learning (or training) process is basically the iterative adjustment of ANN architecture and weights. Analysing a set of training records, which are passed to ANN inputs, the adjustment is performed in order to obtain the outputs being closed to desired ones for the given training data. This self-training property makes ANN more attractive in comparison with other systems, which strongly conform to the predetermined operational rules, being formulated by experts.

It is recognised that one of the most widely used ANN structures for diagnosis problems is Multi-Layer Perception (MLP) with a backpropagation learning algorithm [29]. In this research, a simple three-layer MLP structure with input, hidden and output layers was employed as a classifier for transformer FD. Each neuron model of the hidden layer has a hyperbolic tangent activation function, whereas a logistic activation function is implemented for those of the output layer. The test result of ANN is reported in Section 5.7.2.

Support vector machine

SVM is recognised as one of the standard tools for machine learning and data mining, which was developed by Vapnik and his co-workers at AT&T Bell Labs in 1992 [30], based on advances in statistical learning theory. Being originally developed to solve binary classification problems, SVM determines a number of support vectors from training records and converts them into a feature space using various Kernel functions. The most commonly used kernels are the Gaussian Radial Basis Function (RBF), polynomial, MLP and so on [31]. Thus, by solving a quadratic optimisation problem, SVM defines the optimal separating hyperplane with a maximal margin between the two classes.

For the purpose of multi-category classification, various different binary classification methods have been implemented, such as "one-against-all", "one-against-one", Directed Acyclic Graph SVM (DAGSVM) and so on [91]. The SVM used in this research was a DAGSVM, which has been approved as one of the appropriate binary methods for multi-category classification [91]. In addition, a Gaussian RBF kernel defined by the following equation was employed:

$$Q(x,y) = \exp\left(-\frac{(x-y)^2}{2\varsigma^2}\right), \tag{4.2.1}$$

where x and y denote support vectors and ς is a RBF kernel parameter to be predetermined. In order to control the SVM generalisation capability, a misclassification parameter C also should be defined as cited in [92]. The transformer FD accuracies of SVM are presented in Section 5.7.2.

K-Nearest Neighbour

*K*NN is a supervised learning algorithm introduced by Davis and Rosenfeld in 1978 [32] and has been used in many applications in the fields of data classification, statistical pattern recognition, image processing and many others. It is based upon the assumption that the observations with closest location are members of the same category. The *K* closest neighbours are found from the training data set by calculating the Euclidean distance between the examined point and training records. The K closest data points are then analysed to determine which class label is the most common one among the set with the purpose of assigning it to the data point that is being examined. In general, the value K is selected not to be too small in order to minimise the noise effect in the training data. On the other hand, a large value of K essentially increases the computing time, therefore, in practice K is adjusted experimentally. The experimental results of KNN for transformer FD are illustrated in Section 5.7.2. In the next section, a brief introduction to ARM is provided.

4.3 Basics of association rule mining

ARM is a kind of data mining techniques. In practice, a data mining algorithm can be implemented with a variety of data formats, ranging from numerical measurements and text documents to more complex information such as spatial data, multimedia channels and hypertext documents [93]. Briefly, data mining techniques can be classified into two groups, namely descriptive data mining and predictive data mining. Methods involved in the former group are designed for exploiting general properties of a data set being currently mined. However, inference rules are not generated in such data mining processes. In contrast, data mining methods in the latter group are developed to generate inference rules based on a data set. Then, a classifier can be developed based on these rules and used for the classification tasks of new data sets.

ARM discussed in this thesis is a type of predictive data mining techniques. Previously, ARM has been widely investigated in a number of research areas. In [94], an ARM algorithm was used for Electrocardiograms (ECGs) data mining by Konias and Maglaveras. The mined association rules were used as complements to an ECG plot, and allowed a physician to test a set of hypotheses for discovering hidden heart diseases of a patient. In [95], Kocatas, Gursoy and Atalay employed two data mining techniques, ARM and Iterative Dichotomiser 3 (ID3) classification methods respectively, to solve the problem of predicting protein-protein interactions. With the two approaches, available interaction data and protein domain decomposition data were combined as a unique data repository, which was used to infer new interactions. In the tests, the ARM-based approach outperformed the ID3 method in the number of generated rules. Liu, Chen, Fan and Shen [96] applied ARM to discover the heuristic rules for Power System Restoration (PSR), which were further employed to guide a fast restoration process. In the tests, with the obtained association rules, the performance of some PSR cases were improved. Therefore, ARM was considered as a viable approach to PSR.

In ARM, one item extracted from a database record is defined as an attribute. The main task of an ARM process is to discover potential relationships and correlations among user-interested attributes. The generated association features are subsequently interpreted as a set of association rules. In this study, ARM was employed to generate association rules with a DGA training data set. Then, the generated association rules were utilised for FD of power transformers. Such an ARM implementation process is described detailedly in the following sections of this chapter.

4.4 An association rule mining process

A simple, but typical, knowledge discovery process should include three steps, *i.e.*, preprocessing data, applying a data mining algorithm and postprocessing obtained results. Therefore, as illustrated in Figure 4.4, with a set of DGA records, an ARM implementation process, dedicated to constructing an association rule-based FD system for power transformers, is mainly composed of three procedures: DGA record preprocessing, association rule generation and rule set postprocessing. In the DGA record preprocessing step, main tasks are attribute selection and continuous datum attribute discretisation. Then, in the second procedure, rule generation tasks are carried out using an ARM algorithm, *i.e.*, Apriori-TFP, with selected attributes. In the third step, two rule set postprocessing methods, involving rule set simplification and rule fit-

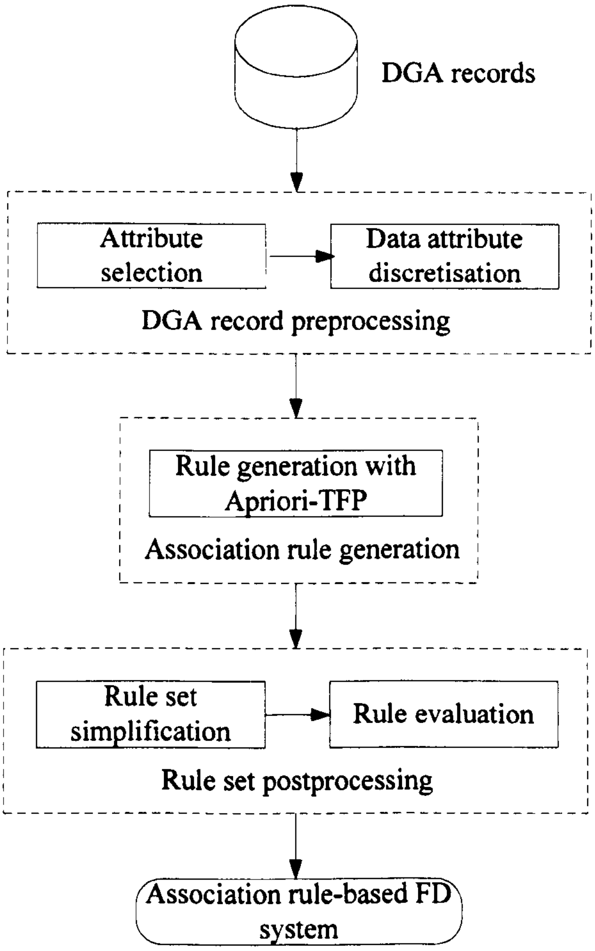

DGA records
Attribute selection
Data attribute discretisation
DGA record preprocessing
Rule generation with Apriori-TFP
Association rule generation
Rule set simplification
Rule evaluation
Rule set postprocessing
Association rule-based FD system

Figure 4.4: Development process of an association rule-based system for FD of power transformers

ness evaluation, are employed. Finally, an FD system is constructed with a set of obtained useful association rules. In the following subsections, the methods used in such an ARM process are presented in detail.

4.4.1 Dissolved gas analysis data preprocessing

Attribute selection

As mentioned in Section 4.3, in ARM, one item extracted from a database record is defined as an attribute. In an ARM process, useless attributes can somehow "confuse" a data mining algorithm, which leads to the discovery of inaccurate or valueless knowledge [97]. Therefore, as can be seen from Figure 4.5, the main motivation of an attribute selection task is to select user-interested attributes from the provided DGA data set for generating useful association rules.

Briefly, attribute selection methods can be summarised into two categories, *i.e.*, autonomous selecting methods and user-specified methods. Practically, an autonomous selecting method employs a mathematical algorithm to choose useful attributes for ARM. By contrast, a user-specified method represents a much simpler way of attribute selection compared with an autonomous method. In other words, with a user-specified method, useful ARM attributes can be directly determined by users. In this research, a user-specified method was employed for selecting ARM attributes, due to the fact that the attributes used the ARM-based DGA approach were clearly known by the domain experts of power systems and data mining. The main ideas of the employed user-specified attribute selection method are stated as below.

As analysed in Section 1.4.2, the classification rules used in the conventional DGA methods, *i.e.*, Dornenburg or Rogers, are established based upon the empirical studies of power engineers. However, a vital limitation of the rules included in such a conventional DGA method is that in some cases, a set of measured gas ratios may not fit within the predefined criteria. As a result, faults that occur inside transformers are not identifiable. In other words, with such classification rules, the main problem is recognised as that, in some FD cases, it is difficult to reveal the relationships between a set of gas ratios and a fault type.

Therefore, the attributes for an ARM process should be selected as the

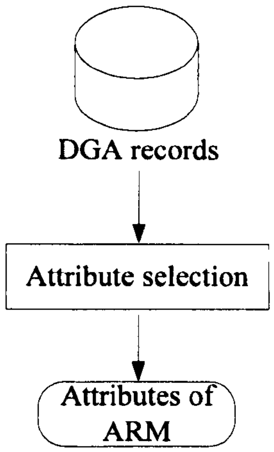


Figure 4.5: An attribute selection process

gas ratios used by the conventional DGA methods and the transformer working state labels of the provided DGA records. Subsequently, the potential relationships between the selected attributes, which are not defined in the conventional DGA methods, may be discovered during the ARM process. As a result, the limitation of the conventional DGA methods stated above, may be solved. A detailed process of attribute selection is illustrated in Section 5.3.

Discretisation of continuous data attribute

A datum attribute discretisation process is defined as a process of dividing a continuous datum attribute value into a finite set of value intervals with the minimal loss of information [98]. As observed from Figure 4.6, with a discretisation process, a continuous datum attribute can be transformed into a set of categorical attributes.

Since the ARM algorithm implemented in this study is capable to process binary valued attributes only, which is a special case of categorical attributes. Therefore, the continuous data attributes, *i.e.*, the gas ratio attributes selected in the last subsection, must be discretised and transformed into a finite set of categorical attributes. Subsequently in an ARM process, according to the con-

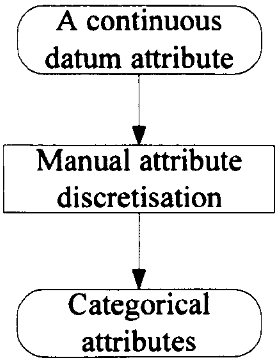


Figure 4.6: A datum attribute discretisation process

firmed attributes, *i.e.*, discretised gas ratio attributes and the working states of a power transformer, a training DGA record is converted into a set of binary valued data prior to an ARM process. When the ARM process has been finished, the generated association rules then are transformed back to the original formats and displayed to users.

Previously, a number of data discretisation techniques have been published, such as equal width, equal depth and entropy and so on [99]. In a discretisation process with one of these techniques, a set of boundary points used for discretising a continuous datum attribute value can be mathematically obtained using a discretisation algorithm. However, in this research, the boundary points used for discretising the value of a gas ratio should be selected with respect to the predefined boundary points of the corresponding DGA method, *i.e.*, Dornenburg or Rogers. This is because these boundary points are discovered based upon a number of empirical experiments and subsequently formalised as the standards of the Institute of Electrical and Electronics Engineers (IEEE) [1] for discretising the value of a gas ratio.

Therefore, a manual discretisation method suggested in [100] was implemented in this study. Normally, this method is employed when the boundary points are clearly known by human users. In the tests, with the manual discretisation method, all the boundary points regarding a continuous data attribute value were determined by domain experts of power system and data mining. This is a vital procedure of the ARM-based DGA approach, which can lead to a valueless knowledge process if boundary points are selected wrongly. A practical example of numeric attribute discretisation is explained in Section 5.3. In the next section, an association rule generation process, with a set of training DGA records, is illustrated.

4.4.2 Association rule generation with an association rule mining algorithm

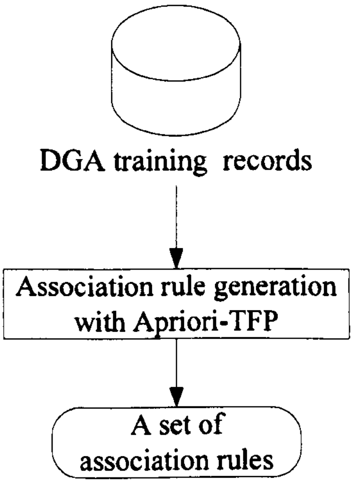

DGA training records
Association rule generation with Apriori-TFP
A set of association rules

Figure 4.7: An association rule generation process

The mining of association rules from a large DGA training data set is a computationally demanding task. Therefore, an accurate ARM algorithm with a fast association rule generation speed is highly required. In this study, the well-known Apriori-TFP algorithm was employed for generating association rules, which can fulfill such requirements.

The original idea of implementing TFP algorithm into ARM was introduced by Golbourne *et al.* [101] and then republished in [102]. In [88], Liverpool University Computer Science-Knowledge Discovery in Data (LUCS-KDD) research team presented an Apriori style ARM algorithm, namely Apriori-TFP. Apriori-TFP is developed based upon Apriori-Total (T) [103] and in turn, Apriori-T is derived from the Apriori [12] algorithm.

Apriori-TFP handles an ARM process in a similar way as Apriori-T. However, instead of processing raw data directly in Apriori-T, in an ARM process with Apriori-TFP, the raw data is firstly preprocessed and stored in a Partial support tree (P-tree). This preprocessing function can effectively reduce the execution time of an ARM process, when many duplicate records and/or duplicate attribute sets exist in raw data [88]. In the tests of this study, all the employed DGA records were collected from the same DGA database and thus might contain a number of duplicate records. In this case, Apriori-TFP is a suitable solution for ARM compared with Apriori-T.

Briefly, an Apriori-TFP ARM process can be described as follows: let $I=[i_1, i_2, \dots i_n]$ be a set of selected items, *i.e.*, attributes. Define T to be a set of training data. Each record R in T is composed of a set of items and assigned with a unique identifier. Also, $R \subseteq I$. Let A and B be two sets of items. A specific record R_t from T contains A if and only if $A \subseteq R_t$. An association rule then can be defined in the following form:

$$A \rightarrow B, \tag{4.4.1}$$

where $A \subset I$, $B \subset I$ and $A \cap B = \varnothing$. In (4.4.1), A and B are defined as the antecedent and consequent of the rule respectively. The expression (4.4.1) means that if A is presented in R_t, then B is likely presented in R_t as well.

With an ARM algorithm, *e.g.*, Apriori-TFP, an association rule is generated when its support and confidence values are greater than user-specified minimum support and confidence values, respectively. Support value of an association rule represents the percentage or number of records in T that contain $A \cup B$, *i.e.*, both A and B. This can be depicted as a probability $\mathrm{P}(A \cup B)$. The second parameter, *i.e.* the confidence value of the rule, is the percentage of records in T holding A, that also contain B. This can be expressed as a probability $\mathrm{P}(B|A)$. Therefore, conclusions are derived as $\mathrm{Support}(A \rightarrow B) = \mathrm{P}(A \cup B)$ and $\mathrm{Confidence}(A \rightarrow B) = \mathrm{P}(B|A)$.

Following an ARM process with Apriori-TFP, an ARS is generated. Subsequently, a set of useful rules can be extracted from the rule set and implemented for FD of power transformers. In the next subsection, the methods employed in such a useful rule generation process, *i.e.*, a rule set simplification method and a rule fitness evaluation method, are presented.

4.4.3 Rule set postprocessing

The use of ARM for FD is based upon the observation that a subset of associations rules, generated by an ARM process, can be utilised for the purpose of FD [104]. Thus, in order to obtain a set of useful FD rules from a rule set generated in Section 4.4.2, a rule set simplification method and a rule fitness evaluation method, used to prune the useless rules from the rule set and assign a fitness value to each of the remaining useful rules, were employed. Such a useful rule selection process is shown in Figure 4.8.

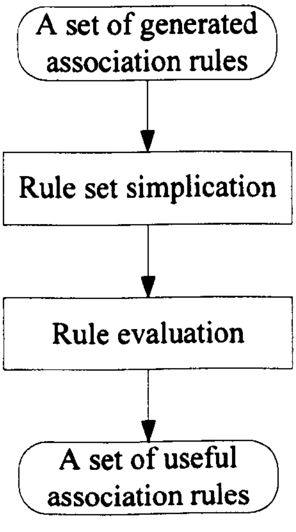

A set of generated association rules
Rule set simplication
Rule evaluation
A set of useful association rules

Figure 4.8: A process for eliminating useless rules

The rule simplification method is capable of pruning the useless rules from a generated ARS. In this study, the use of an association rule from a generated rule set aims to classify the working state of a power transformer with a given set of gas ratios. Hence, a rule whose consequence is not assigned as a unique transformer working state, *i.e.*, various fault classes or no fault state as mentioned above, cannot be used in such an FD task. Consequently, the rule is treated as a useless rule and then eliminated from the rule set.

Subsequently, in order to evaluate the correctness confidence of the remaining useful rules for an FD task, each of the rules needs to be evaluated by a fitness evaluation function and then assigned with a fitness value. Briefly, the fitness value of an association rule represents the correctness confidence of the rule on an FD task. Supposing that, two association rules are capable of diagnosing a test DGA record, a large fitness value means that the association rule may diagnose the record with a higher accuracy, compared with that obtained using the other rule with a small fitness value.

A rule evaluation process can be time-consuming, when a rule set contains a large number of rules. Thus, by applying the rule set simplification method stated above, the number of rules, included in the rule set can be reduced before the rule evaluation process. As a result, the time consumed for the rule evaluation process is decreased. In this study, the fitness value evaluation function published in [105] was employed for evaluating the fitness values of the useful rules, which were obtained with the rule set simplification process explained above.

For an association rule, the fitness value mainly depends on its predictive accuracy and comprehensibility. Therefore, returning to Section 4.4.2, let a rule be the form as (4.4.1). As mentioned in Section 4.4.2, the parameter for evaluating the predictive accuracy of an association rule is a confidence value. Practically, such a simple measure for evaluating the fitness value of a generated rule is unreliable in some cases. For example, if only one record existing in a database matches a generated rule, the confidence value of the rule could then be obtained as $P(B|A)=1/1=100\%$. Apparently, this rule most likely represents the idiosyncrasy of such a training data set. Therefore, the predictive accuracy of the rule cannot be guaranteed on other test data sets by using only a confidence value [105]. Therefore, in the employed fitness function, the fitness value of an association rule is evaluated based upon three individual parameters, namely confidence, completeness and simplicity. The mathematic algorithms for computing the values of these three parameters of an association rule are introduced as follows.

The values of the first two parameters, *i.e.*, confidence and completeness, are further determined by the following elements, *i.e.*, True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN), as illustrated in Table 4.5.

Table 4.5: Confusion matrix for evaluating an association rule

		Actual consequent	
		B	not B
Diagnosed consequent	B	TP	FP
	not B	FN	TN

As seen from Table 4.5, the meanings of TP, FP, FN and TN are defined as:

1. TP: Number of records from a training data sets satisfying A and B;
2. FP: Number of records from a training data sets satisfying A but not B;
3. FN: Number of records from a training data sets not satisfying A but satisfying B;
4. TN: Number of records from a training data sets satisfying neither A nor B.

Obviously, the fitness value of an association rule is proportional to the values of TP, and inversely proportional to the values of FP and FN. With the elements in the confusion matrix, the confidence value of an association rule, *i.e. Confidence*, can be defined as:

$$Confidence = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \qquad (4.4.2)$$

Also, the value of the parameter completeness (defined as *Completeness*), *i.e.*, the proportion of records containing the predicted consequent B that is actually covered by the rule antecedent A, can be denoted as:

$$Completeness = \frac{\text{TP}}{\text{TP+FN}}. \tag{4.4.3}$$

Hence, a fitness function for evaluating the fitness of an association rule is depicted as:

$$Fitness = Confidence * Completeness, \tag{4.4.4}$$

where *Fitness* is the fitness value of an association rule.

However, such a function does not include any evaluation of comprehensiveness of an association rule. Thus, the above equation is modified as below:

$$\begin{aligned} Fitness = {} & w_1 * (Confidence * Completeness) \\ & + w_2 * Simplicity, \end{aligned} \tag{4.4.5}$$

where w_1 and w_2 are relative weights defined by domain experts. In the tests of this study, w_1 and w_2 were assigned as 0.7 and 0.3, respectively. *Simplicity* is the simplicity degree of a rule and the value is assigned in the range [0, 1], and

$$Simplicity = \frac{1}{N}, \tag{4.4.6}$$

where N is the number of attributes existed in the antecedent of a rule.

When each of the useful rules is evaluated with the above fitness evaluation function and assigned with a fitness value, all the useful rules then are ranked from the highest value to the lowest value. Subsequently, an association rule-based FD system for power transformer FD is constructed, in which the evaluated rule set is served as an FD knowledge base. In the next section, the structure of such an FD system is introduced firstly. Then, the implementation method of the FD system is discussed.

4.5 Implementation method of an association rule-based transformer fault diagnosis system

The association rule-based FD system, constructed in Section 4.4.3, is illustrated in Figure 4.9. As observed from the figure, the system is mainly composed of an association rule base and an optimal rule selection module [105]. In a diagnosis process with this FD system, a rule which offers the highest probability for correctly diagnosing a specific test DGA record, can be selected from the FD system using an optimal rule selection method, according to the fitness values of all the rules involved in the FD system. Then, the DGA record can be diagnosed by the rule.

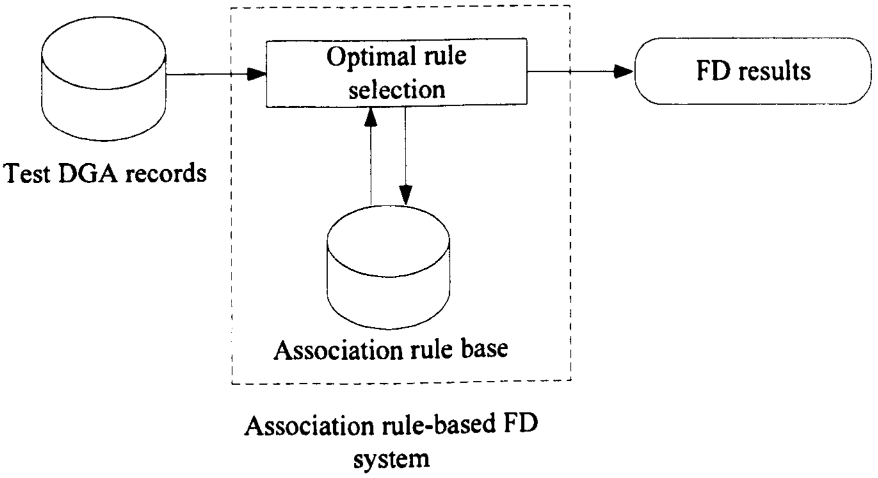


Figure 4.9: An FD process with an association rule-based FD system

The basic procedures of an optimal rule selection process for diagnosing a test DGA record are:

1. For the test DGA record being diagnosed, association rules whose antecedents are satisfied by the DGA record are selected firstly;

2. Then, according to the fitness value of each selected rule, the rule which owns the highest fitness value is treated as the optimal rule and utilised to diagnose the record.

A method for deciding whether the antecedent of an association rule is satisfied by a DGA record is described as follows. Let R_{te} be a record from a test data set and R_{tc} is denoted as a vector $(A_1, \ldots, A_k, \ldots, A_K)$, where A_k $(k = 1, \ldots, K)$ is the kth attribute of R_{te}. A literal p is an attribute-value pair with the form $(A_k,\ v)$, where v is a possible value of the attribute A_k. Define that R_{te} satisfies a literal p if and only if $R_k = v$, where R_k is the value of A_k in R_{tc}.

An association rule r is given as: $r = p_1 \wedge p_2 \wedge \ldots \wedge p_l \rightarrow c$. In r, the antecedent is the conjunction of literals p_1, p_2,. . ., p_l and the consequent is the class c. The record R_{te} satisfies the antecedent of r if and only if it satisfies every literal of the antecedent of r. In the case that R_{tc} satisfies r's antecedent, the working state of R_{te} then is classified as the class c. If a rule contains zero literal, its body is satisfied by any record.

In next chapter, tests used for verifying the FD performance of the proposed ARM-based DGA approach is demonstrated. The results from the tests then are compared with that achieved by a set of other fault classification techniques, *i.e.*, the Dornenburg and the Rogers ratio methods, as well as the ANN, SVM and KNN black-box fault classifiers.

4.6 Conclusion

The ARM-based DGA approach to FD of power transformers has been presented in this chapter. In order to generate an ARS with a set of DGA records, the data preprocessing and the rule discovery methods were introduced firstly. The use of the ARM-based DGA approach aims at extracting a set of useful association rules from the obtained rule set, which can be further used for FD of power transformers. Therefore, the rule set simplification method and the rule fitness evaluation method were presented for eliminating the useless rules from the obtained rule set and assigning a fitness value to each of the remaining useful rules. Subsequently, a transformer FD system was established with the useful association rules. With the purpose of implementing the FD system, the optimal rule selection method then was explained, with which the most accurate rule can be selected from the FD system for diagnosing a specific test DGA record.

Chapter 5

Performance Evaluation of The Association Rule Mining-Based Dissolved Gas Analysis Approach

5.1 Introduction

In the last chapter, an ARM-based DGA approach to transformer FD has been presented. In this chapter, the practical performance of the ARM-based DGA approach is evaluated and obtained results are reported subsequently. In Section 5.2, experiment data and experiment schemes are introduced, respectively. DGA data preprocessing results then are reported in Section 5.3. Section 5.4 presents the obtained results of association rule discovery processes using Apriori-TFP. An FD scenario with a constructed association rule-based FD system is demonstrated in Section 5.5. In Section 5.6, the overlapping frequencies generated between useful ARSs and the empirical rules used in two conventional DGA methods, *i.e.*, Dornenburg and Rogers, are illustrated and explained. The FD accuracies of the constructed association rule-based FD systems then are discussed in Section 5.7. Comparison results, which were generated with five conventional fault classification techniques, *i.e.*, the Dornenburg and the Rogers ratio methods, the ANN, SVM and KNN classifiers using the same training and test DGA records, are also reported. The chapter is concluded in Section 5.8.

5.2 Experiment data and schemes

In order to illustrate the practical performance of the ARM-based DGA approach, a number of tests were undertaken in the *e*-Automation laboratory at the University of Liverpool.

In general, an ARM process can extract more association features among user-interested attributes from a large training DGA data set, compared with that obtained with a small data set. Consequently, more useful association rules may be derived from these association features and a higher classification accuracy may be achieved with the rules for a classification task. On the other hand, in an FD process with an association rule-based FD system, a given test DGA record can be diagnosed as one of the various transformer working states. Thus, it is logical to deduce that the employed association rules are reliable in a classification task if generated using a training data set with equal number of records regarding different transformer working states.

Therefore, a large training data set and a test data set, consisting of 1091 and 181 DGA records respectively, were directly taken from a NG (National Grid, U.K.) DGA database and used for evaluating the FD capability of the proposed ARM-based DGA approach. The extracted DGA records contained not only the seven types of key gases as stated in Section 1.3.2, but also the diagnosed working states obtained from on-site inspections. The DGA records were evaluated using various engineering diagnostic tools by industry experts and obtained evaluation results were related to four classes, *i.e.*, thermal, PD, arcing and no fault. It should be noted that the DGA records from both the training and test data sets were almost with equal numbers according to various transformer working states.

The provided DGA data set for ARM, containing 1091 training and 181 test DGA records, was defined as D_{Ori}. Practically, due to the fact that the Dornenburg ratio method imposes limits with regard to the values of the gas concentrations to be processed [1] (Table 4.1), the training and test DGA records of D_{Ori} were reduced to 1016 training and 177 test records, respectively. Then, the DGA records were converted into the format as discussed in Section 4.1, *i.e.*, being organised with the gas ratios used in a conventional DGA method and a working state obtained from on-site inspections as stated above. In this study, for comparison purposes, the gas ratios of the Dornenburg and Rogers ratio methods were employed to reformat the DGA records, respectively. Consequently, two new data sets were derived from D_{Ori} and defined as D_{Dor} and D_{Rog}, both of which were composed of 1016 training and 177 test records, respectively.

For D_{Dor} and D_{Rog}, association rules were generated with the training sets at various user-specified support-confidence value points. In the tests, firstly, an overlapping frequency was calculated between a set of generated association rules and the empirical rules defined in the corresponding DGA method, *e.g.*, between a rule set generated based upon the training set of D_{Dor} and the rules of the Dornenburg ratio method. An overlapping frequency illustrates the number of the conventional Dornenburg rules that can be discovered by an ARM process with the training set of D_{Dor}. This test was designed to illustrate the capability of discovering the empirical rules employed in the Dornenburg ratio method and the Rogers ratio method using the proposed approach.

Then, a set of FD processes were implemented with the generated ARSs, by processing the test data sets of D_{Dor} and D_{Rog}, separately. Consequently, the FD accuracies of the proposed ARM-based DGA approach were obtained. For comparison purposes, the FD accuracies of the Dornenburg and Rogers ratio methods were also verified with the test data sets of D_{Dor} and D_{Rog}, respectively. Moreover, the FD performance of the ANN, SVM and KNN black-box classifiers were assessed with the training and test data sets of D_{Ori}, separately. At the end of the chapter, the diagnosis results obtained by the

Table 5.1: ARM attributes selected for D_{Dor} and D_{Rog} before discretisation

	Selected attributes
D_{Dor}	CH_4/H_2, C_2H_2/C_2H_4, C_2H_2/CH_4, C_2H_6/C_2H_2, no fault, PD, arcing, thermal
D_{Rog}	CH_4/H_2, C_2H_2/C_2H_4, C_2H_4/C_2H_6, no fault, PD, arcing, thermal

above methods are compared and discussed.

5.3 Data preprocessing results

For D_{Dor} and D_{Rog}, the attributes employed in an ARM process were only related to the gas ratios of the DGA criteria, *i.e.*, Dornenburg or Rogers, and the available transformer working states from on-site inspections, *i.e.*, thermal, PD, arcing and no fault. Hence the user-specifying method introduced in Section 4.4.1 was implemented for attribute selection tasks, demonstrated as follows.

Firstly, the four gas ratios employed in the Dornenburg ratio method, shown in Table 4.2, were selected as a part of the attributes for ARM regarding D_{Dor}. On the other hand, the three gas ratios, utilised for FD in the Rogers ratio method from Table 4.4, were chosen as a part of the attributes for ARM of D_{Rog}. Meanwhile, the transformer working states of the involved DGA records were thermal, PD, arcing and no fault, as mentioned above. Thus, these four transformer fault classes were chosen as another part of the ARM attributes for the data sets of D_{Dor} and D_{Rog}. As a result, before an datum attribute discretisation procedure, in total eight (CH_4/H_2, C_2H_2/C_2H_4, C_2H_2/CH_4, C_2H_6/C_2H_2, no fault, PD, arcing and thermal) and seven attributes (CH_4/H_2, C_2H_2/C_2H_4, C_2H_4/C_2H_6, no fault, PD, arcing and thermal) were confirmed for D_{Dor} and D_{Rog} respectively, as shown in Table 5.1.

Next, the discretisation method of continuous data attributes explained in Section 4.4.1 was implemented for converting the continuous attribute values, *i.e.*, gas ratio values, into sets of discretised values. Here, the attribute CH_4/H_2 of D_{Dor} is provided as an example. According to Table 4.2, the continuous value of CH_4/H_2 was discretised and replaced by three intervals, *i.e.*, "<0.1", ">0.1 and <1.0", ">1.0". As stated in Section 4.4.1, the reason of selecting these boundary points is due to that they are discovered based upon a number of empirical experiments and subsequently formalised as IEEE standards for discretising the value of a gas ratio. The same method then was employed for the value discretisation of all the other numeric attributes of D_{Dor} and D_{Rog}.

Subsequently, with respect to the selected ARM attributes, the DGA records of the training sets of D_{Dor} and D_{Rog} were converted into binary valued data prior to an ARM process. When the ARM process has been finished, the generated association rules then were transformed back to the original formats and displayed to users. However, this procedure did not influence the results of the ARM process. Thus, it is not discussed detailedly in this thesis.

With the obtained attributes, tests were implemented to exploit the capability of the ARM-based DGA approach with regards to transformer FD, as illustrated in the following subsections.

5.4 Association rule mining results with rule set postprocessing

Table 5.2: ARM results generated based upon the Dornenburg ratio method

	Sup.=1			
	Conf.=1%	Conf.=15%	Conf.=35%	Conf.=70%
Rules generated	105	81	63	46
Overlapping frequency	3	3	3	2

Continued on next page...

Table 5.2 – continued from previous page

Correctly diagnosed	156	156	156	94
Wrongly diagnosed	17	17	17	12
Not processable	4	4	4	71
Diagnosis accuracy (%)	**88.14**	**88.14**	**88.14**	53.11
	Sup.=8			
	Conf.=1%	Conf.=15%	Conf.=35%	Conf.=70%
Rules generated	84	67	49	32
Overlapping frequency	3	3	3	2
Correctly diagnosed	156	156	136	74
Wrongly diagnosed	17	17	17	12
Not processable	4	4	24	91
Diagnosis accuracy (%)	**88.14**	**88.14**	76.84	41.81
	Sup.=15			
	Conf.=1%	Conf.=15%	Conf.=35%	Conf.=70%
Rules generated	67	60	46	28
Overlapping frequency	3	3	3	2
Correctly diagnosed	136	136	136	74
Wrongly diagnosed	17	17	17	12
Not processable	24	24	24	91
Diagnosis accuracy (%)	76.84	76.84	76.84	41.81
	Sup.=22			

Continued on next page...

Table 5.2 – continued from previous page

	Conf.=1%	Conf.=15%	Conf.=35%	Conf.=70%
Rules generated	35	35	25	14
Overlapping frequency	3	3	3	2
Correctly diagnosed	136	136	136	69
Wrongly diagnosed	17	17	17	7
Not processable	24	24	24	101
Diagnosis accuracy (%)	76.84	76.84	76.84	38.98

For each of D_{Dor} and D_{Rog}, in total 1016 training records, were utilised to generate association rules using the Apriori-TFP algorithm, which is discussed in Section 4.4.2. Generally, in ARM, as the support and confidence values increase, the number of rules found by an ARM process reduces. Therefore, some useful rules may be eliminated during the ARM process and the accuracy of an association rule-based FD system constructed subsequently can decrease. In order to obtain the highest diagnosis accuracy with the proposed approach, association rules were generated with various sets of support and confidence values, specified by the domain experts of data mining. As can be seen from Table 5.2, association rules were obtained with the training set of D_{Dor} at four different support values (Sup.) separately, *i.e.*, 1, 8, 15 and 22. At each of these support values, there were four different confidence values (Conf.) assigned as well, *i.e.*, 1%, 15%, 35% and 70%. Therefore, in total 16 sets of support and confidence values were used to generate association rules with the training data set of D_{Dor}.

Table 5.3: ARM results generated based upon the Rogers ratio method

	Sup.=1			
	Conf.=1%	Conf.=15%	Conf.=35%	Conf.=70%
Rules generated	103	85	63	33
Overlapping frequency	5	5	5	5
Correctly diagnosed	162	162	162	133
Wrongly diagnosed	15	15	15	14
Not processable	0	0	0	30
Diagnosis accuracy (%)	**91.53**	**91.53**	**91.53**	75.14
	Sup.=8			
	Conf.=1%	Conf.=15%	Conf.=35%	Conf.=70%
Rules generated	88	78	58	30
Overlapping frequency	5	5	5	5
Correctly diagnosed	162	162	162	133
Wrongly diagnosed	15	15	15	14
Not processable	0	0	0	30
Diagnosis accuracy (%)	**91.53**	**91.53**	**91.53**	75.14
	Sup.=15			
	Conf.=1%	Conf.=15%	Conf.=35%	Conf.=70%
Rules generated	63	55	48	25
Overlapping frequency	5	5	5	5

Continued on next page...

Table 5.3 – continued from previous page

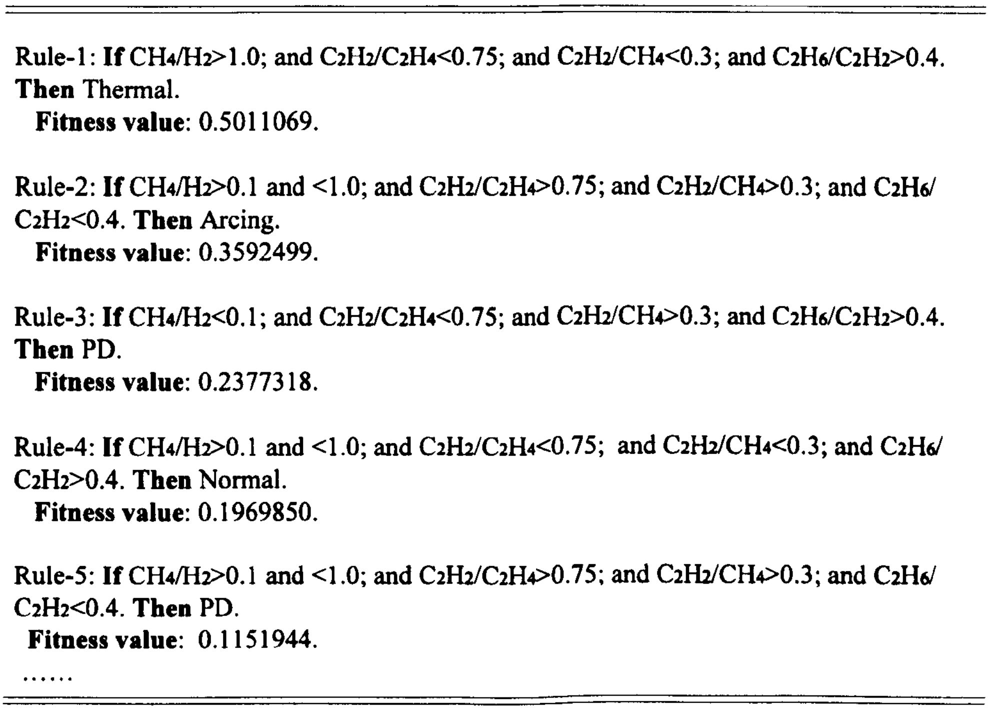
Correctly diagnosed	162	162	162	133
Wrongly diagnosed	15	15	15	14
Not processable	0	0	0	30
Diagnosis accuracy (%)	**91.53**	**91.53**	**91.53**	75.14
	Sup.=22			
	Conf.=1%	Conf.=15%	Conf.=35%	Conf.=70%
Rules generated	40	35	35	23
Overlapping frequency	5	5	5	5
Correctly diagnosed	146	146	146	128
Wrongly diagnosed	14	14	14	5
Not processable	17	17	17	44
Diagnosis accuracy (%)	82.49	82.49	82.49	72.32

When an ARS was generated, the rule set simplification and rule fitness evaluation methods introduced in Section 4.4.3 were employed to prune useless rules from the rule set. Moreover, each of the remaining useful rules from the rule set was assigned with a fitness value.

The rule numbers generated with the training data set of D_{Dor} at all the 16 support-confidence value points are described in Table 5.2. As illustrated in the table, at the same support value, the number of the useful rules obtained decreases with the increment of the confidence values. Also, at the same confidence value, the number of the useful rules reduces when the support value rises. As a result, after the rule set simplification, the largest rule set with 105 useful association rules was discovered at the support-confidence value point (1, 1%). In contrast, the smallest rule set with only 14 useful rules was derived at the support-confidence value point (22, 70%).

On the other hand, the same tests were implemented on the training data set of D_{Rog}. As shown in Table 5.3, again, the number of useful rules decreases when either support or confidence value increases. The largest rule set with 103 useful association rules was discovered at the support-confidence value point (1, 1%), and the smallest rule set was obtained at the support-confidence value point (22, 70%) with 23 useful rules, after being pruned.

5.5 A fault diagnosis scenario with an association rule-based fault diagnosis system



Rule-1: **If** $CH_4/H_2>1.0$; and $C_2H_2/C_2H_4<0.75$; and $C_2H_2/CH_4<0.3$; and $C_2H_6/C_2H_2>0.4$. **Then** Thermal.
Fitness value: 0.5011069.

Rule-2: **If** $CH_4/H_2>0.1$ and <1.0; and $C_2H_2/C_2H_4>0.75$; and $C_2H_2/CH_4>0.3$; and $C_2H_6/C_2H_2<0.4$. **Then** Arcing.
Fitness value: 0.3592499.

Rule-3: **If** $CH_4/H_2<0.1$; and $C_2H_2/C_2H_4<0.75$; and $C_2H_2/CH_4>0.3$; and $C_2H_6/C_2H_2>0.4$. **Then** PD.
Fitness value: 0.2377318.

Rule-4: **If** $CH_4/H_2>0.1$ and <1.0; and $C_2H_2/C_2H_4<0.75$; and $C_2H_2/CH_4<0.3$; and $C_2H_6/C_2H_2>0.4$. **Then** Normal.
Fitness value: 0.1969850.

Rule-5: **If** $CH_4/H_2>0.1$ and <1.0; and $C_2H_2/C_2H_4>0.75$; and $C_2H_2/CH_4>0.3$; and $C_2H_6/C_2H_2<0.4$. **Then** PD.
Fitness value: 0.1151944.

……

Figure 5.1: A segment of a set of generated association rules

In order to obtain an in-depth understanding of implementing the proposed ARM-based DGA approach to FD of power transformers, a simple FD scenario is presented in this section before the discussion of the obtained FD results. An example showing a segment of a set of association rules, which are in the form (4.4.1), is illustrated in Figure 5.1. This rule set was obtained with the training set of D_{Dor}, at the support-confidence value point (1, 1%). All the rules were ordered with their fitness values in a decreasing sequence. Supposing that, a DGA record from the test data set was provided as follows:

$$\mathrm{CH_4/H_2} = 0.33; \qquad \mathrm{C_2H_2/C_2H_4} = 0.81;$$
$$\mathrm{C_2H_2/CH_4} = 0.54; \qquad \mathrm{C_2H_6/C_2H_2} = 0.11.$$

With the optimal rule selection method introduced in Section 4.5, firstly, two rules, *i.e.,* Rule-2 and Rule-5, were selected, since the given DGA record satisfied the antecedents of Rule-2 and Rule-5, respectively. Compared with the fitness value 0.1151944 of Rule-5, it was obvious that Rule-2 had the priority for diagnosing the provided record with the fitness value 0.3592499. Therefore, Rule-5 was discarded and Rule-2 was chosen as the optimal rule and used for FD of the DGA record. The diagnosis result was then derived as arcing, which was the same as the on-site diagnosis result.

5.6 Overlapping frequencies

This section presents the overlapping frequencies generated between the obtained useful rule sets and the rules defined in the corresponding conventional DGA methods. Firstly, the overlapping frequencies generated between every rule set from Table 5.2 and the empirical rules of the Dornenburg ratio method are reported, which indicate the number of the Dornenburg rules that can be discovered by an ARM process with the training set of D_{Dor}. Then, the overlapping frequencies obtained between the rule sets in Table 5.3 and the empirical rules of the Rogers ratio method are also presented in the section.

As shown in Table 4.2 and Table 4.4, three and six empirical rules are included in the Dornenburg and the Rogers ratio methods respectively, regarding the various working states of a power transformer. However, in the tests, the working states of a power transformer obtained from on-site inspections were only related to thermal, PD, arcing and no fault. In order to calculate the overlapping frequencies with the six rules defined in Table 4.4, the involved working states were unified according to the above four classes. In other words, the three empirical rules defined in Table 4.4, *i.e.*, the rules regarding the fault types low temp. thermal, thermal< 700°C and thermal> 700°C, were all treated as the rules for classifying a thermal fault.

For illustrative purposes, here, the rule set from Table 4.2 and the rule set generated at the support-confidence value point (1, 1%) in Table 5.2 is used to demonstrate the method for calculating an overlapping frequency. In this process, the three empirical rules from Table 4.2 were treated as three test DGA records. With the optimal rule selection method introduced in Section 4.5, all the three records were diagnosed by the ARS. For the three records, the working states were classified as thermal, PD and arcing respectively, which are the same as that defined in Table 4.2. Hence, an overlapping frequency was obtained as three, as illustrated in Table 5.2. It should be noted that, as shown in Figure 5.1, the three empirical rules were listed in the top three positions in the derived ARS. The overlapping frequencies were also obtained between the other rule sets in Table 5.2 and the rule set from Table 4.2. The derived results are shown in Table 5.2. As shown in Table 5.2, with all the ARSs except for those that were generated at support-confidence value points (1, 70%), (8, 70%), (15, 70%), (22, 70%), the overlapping frequencies were obtained as three. Furthermore, the three empirical rules were listed in the top three positions in each of these ARSs. For each of the ARSs generated at support-confidence value points (1, 70%), (8, 70%), (15, 70%), (22, 70%), the overlapping frequency was derived as two. This result indicates that useful rules may be eliminated when an ARS is generated with a large support value or confidence value. However, in each of these four ARSs, the discovered empirical rules of the Dornenburg ratio method were all listed as the top two association rules.

On the other hand, the same tests were undertaken between the rule sets generated based upon the training set of D_{Rog} and the six empirical rules from

Table 4.4. The obtained results are described in Table 5.3. As can be seen from Table 5.3, with each of the generated ARSs, the overlapping frequency was obtained as five. Moreover, the five empirical rules were listed in the top 10 positions of all the ARSs. In the tests, with the inspections of domain experts, the empirical rule that was not discovered in all the ARSs was "**If** $CH_4/H_2>1.0$; $C_2H_2/C_2H_4<1.0$; $1.0\leq C_2H_4/C_2H_6\leq 3.0$. **Then** Thermal". In this case, the fault type thermal represented the fault type thermal< 700°C, as defined in Table 4.4. By analysing the training set of D_{Rog}, it was discovered that the cause of this situation was due to the lack of records regarding the antecedents and consequents of the above empirical rule in the training set. As a result, the rule could not be discovered with an ARM process. However, as discussed above, the capability of the proposed ARM-based DGA approach to discovering the empirical rules of the Dornenburg and the Rogers ratio methods has been clearly demonstrated.

With each of the obtained useful rule sets from Table 5.2 and Table 5.3, an FD system was constructed, as explained in Section 4.4.3. Then, the FD system was used to diagnose the DGA records from the corresponding test data sets of D_{Dor} and D_{Rog}, respectively. As discussed in Section 4.5, a test DGA record may satisfy the antecedents of several rules in an association rule-based FD system. However, only one optimal rule can be selected and used for FD for the record. In order to choose such an optimal rule, the optimal rule selection method introduced in Section 4.5 was implemented. The obtained FD results with these FD systems are discussed in the next section.

5.7 Fault diagnosis results

5.7.1 Diagnosis results compared with that of traditional gas ratio methods

In this subsection, the FD accuracies of all the constructed FD systems, derived with the 177 test DGA records of D_{Dor} and D_{Rog} respectively, are reported.

The FD accuracy values concerning D_{Dor} are illustrated in Table 5.2. As indicated by Table 5.2, the highest FD accuracy was generated as 88.14% at 5 different support-confidence value points, *i.e.*, (1, 1%); (1, 15%); (1, 35%); (8, 1%) and (8, 15%). Meanwhile, the lowest accuracy 38.98% was obtained at the support-confidence value point (22, 70%). On the other hand, with D_{Rog}, the highest diagnosis accuracy was higher than that of D_{Dor} with a value 91.53%, as shown in Table 5.3. This accuracy was achieved at nine different support-confidence value points, *i.e.*, (1, 1%); (1, 15%); (1, 35%); (8, 1%); (8, 15%); (8, 35%); (22, 1%); (22, 15%) and (22, 35%). Meanwhile, the lowest accuracy was obtained as 72.32% at the support-confidence value point (22, 70%).

Table 5.4: FD accuracies (%) of the Dornenburg and Rogers ratio methods

	Dornenburg ratio method	Rogers ratio method
Correctly diagnosed	83	48
Wrongly diagnosed	7	24
Not processable	87	105
Accuracy	**46.89**	**27.19**

The FD accuracies obtained by using the Dornenburg and Rogers ratio methods, with the 177 test DGA records of D_{Dor} and D_{Rog} respectively, are shown in Table 5.4. As illustrated by Table 5.4, with the Dornenburg ratio method, the accuracy was calculated as 46.89%. Compared with the highest accuracy 88.14% in Table 5.2, it is apparent that the proposed ARM-based approach to DGA has produced a better performance for transformer FD. Also, compared with the accuracy 27.19%, which was generated using the Rogers ratio method, the best result 91.53% from Table 5.3 has shown a much improved FD accuracy.

The obtained results illustrate that, potential rules which are not included in the empirical rules of the Dornenburg and Rogers ratio methods could be discovered with the proposed ARM-based DGA approach, using a set of real DGA records. Then, compared with the empirical rules, a set of obtained association rules offers a greater capability for dealing with various transformer FD cases. Therefore, the FD accuracy of the proposed ARM-based DGA approach has been proven to be much higher than that of the corresponding empirical DGA methods.

5.7.2 Diagnosis results compared with that of three conventional black-box fault classifiers

Table 5.5: FD accuracy (%) of ANN with different neuron number applied

	3 Neurons	4 Neurons	5 Neurons	6 Neurons	8 Neurons
mean	22.10	38.28	38.89	36.69	48.51
st.dv	14.22	16.91	11.12	9.54	11.41
best	46.41	59.91	49.72	50.83	60.77
	10 Neurons	12 Neurons	15 Neurons	20 Neurons	
mean	39.00	50.05	45.97	46.19	
st.dv	6.07	9.51	8.63	8.44	
best	45.30	**62.43**	59.67	59.67	

The training and test data sets of D_{Ori}, with 1016 and 177 DGA records respectively, were applied to the three black-box classifiers introduced in Section 4.2.2, *i.e.*, ANN, SVM and KNN. The FD results obtained by the three classifiers are listed in Table 5.5, 5.6 and 5.7, respectively.

As shown in Table 5.5, ANN was tested with nine different neuron values, ranging from 3 to 20. The best diagnosis accuracy was achieved as 62.43% with 12 neurons. In Table 5.6, the test results obtained by SVM at 48 different $\varsigma - C$ points are illustrated. The diagnosis accuracies were derived from 28.83%

Table 5.6: FD accuracy (%) of SVM with different parameters applied

ς	$C = 0.25$	$C = 2.5$	$C = 25$	$C = 250$	$C = 2500$	$C = 25000$
0.0001	37.66	53.46	56.85	51.23	65.6	71.5
0.0005	57.00	54.64	45.69	66.33	73.91	49.19
0.001	51.99	56.46	43.65	69.29	56.71	46.72
0.005	45.91	41.96	73.70	49.95	39.03	41.25
0.01	44.23	47.88	59.87	40.87	43.09	43.09
0.1	75.70	**82.10**	75.11	75.11	75.11	75.11
1	44.81	50.36	50.36	50.36	50.36	50.36
10	28.83	28.83	28.83	28.83	28.83	28.83

Table 5.7: FD accuracy (%) of KNN with different neighbour number applied

$K = 5$	$K = 10$	$K = 15$	$K = 20$	$K = 25$	$K = 30$	$K = 35$
53.04	58.56	**66.85**	62.98	51.93	58.56	56.35
$K = 40$	$K = 45$	$K = 50$	$K = 55$	$K = 60$	$K = 65$	$K = 70$
59.67	59.12	59.12	59.12	62.43	60.77	57.46

to 82.10%, where the best value 82.10% was generated at the $\varsigma - C$ point (0.1, 2.5). Table 5.7 illustrates the test results of KNN, calculated at 14 different K levels. The optimal FD accuracy of 65.85% was reached when the number of neighbours K was set as 15.

Compared with the highest accuracies obtained by the three conventional conventional classifiers, *i.e.*, 62.43%, 82.10% and 65.85%, it is apparent that the ARM-based approach presented in this study has demonstrated a better performance for FD of power transformers.

5.8 Conclusion

In this chapter, the test results of the proposed ARM-based DGA approach has been presented. For comparison purposes, in the experiments, several methods for FD of power transformers were implemented with the same training and test data sets respectively, including the ARM-based DGA approach, the conventional Dornenburg and Rogers ratio methods, the ANN, SVM and KNN classifiers. The final results demonstrated that the novel ARM-based DGA approach has achieved the highest FD accuracies, compared with that obtained by the other methods. In addition, the capability of the ARM-based DGA approach to discovering the empirical rules of the Dornenburg and the Rogers ratio methods has been verified as well. As a conclusion, the proposed ARM-based DGA approach can be proposed as a viable solution for FD of power transformers.

Chapter 6

Semantic Rule-Based Transformer Fault Diagnosis Expert System

6.1 Introduction

Currently, RBESs are widely employed for decision making in various areas. However, the inter-operability among these RBESs is greatly restricted, due to the lack of the development of a standardisation rule language for sharing rule bases among the different RBESs. Recently, SWRL is recognised as an important step for defining such a rule language. In order to efficiently reuse a generated useful ARS in different RBESs of transformer FD, in this chapter, SWRL is presented for interpreting a useful ARS as a SWRL rule base. Then, Java [106] expert system shell (Jess) [107] is introduced, in which an efficient rule inference engine for executing the SWRL rule base is provided. Subsequently, the process of establishing a SRBES with the SWRL rule base and Jess, which is used for transformer FD, is illustrated. At the end of this chapter, tests implemented for evaluating the transformer FD performance of the SRBES are illustrated.

The rest of this chapter is organised as follows: In Section 6.2, firstly, a brief introduction to SWRL is given. Meanwhile, several advantages of using a SWRL rule base in a RBES, which are not achievable by employing an ARS, are analysed in detail. RBESs and Jess are described in Section 6.3. In Section 6.4, the system framework of a SRBES is illustrated. In addition, the functions of the SRBES modules are presented in detail. A SRBES development process is demonstrated in Section 6.5, along with a suggested implementation strategy. In Section 6.6, the FD accuracy of a SRBES is evaluated with a number of test DGA records. The chapter is concluded in Section 6.7.

6.2 Semantic web rule language

As mentioned in Section 6.1, currently, RBESs are widely employed for decision making in various areas. However, the inter-operability among these RBESs is greatly restricted, due to the lack of the development of a standardisation rule language for sharing rule bases among the different RBESs. Recently, SWRL is recognised as an important step for defining such a rule language. As explained in the SWRL specification, SWRL is not bundled with any specialised rule inference engine. However, a SWRL rule engine bridge is provided by SWRL, with which a SWRL rule base can be effectively processed by a variety of published rule engines, *e.g.*, Jess, Hoolet [108], Algernon [109] and SweetRules [110] and so on. This mechanism thus provides a convenient starting point for sharing a SWRL rule base among different RBESs, developed with different rule engines.

Previously, besides SWRL, a number of rule languages have been developed for the goal of sharing a rule base among different RBESs, *e.g.*, Rule Markup Language (RuleML) [111], Metalog [112] and ISO Prolog [113] and so on. Compared with these rule languages, SWRL offers a great capability for reasoning OWL-based knowledge. Therefore, a SWRL rule base can be seamlessly integrated into an agent-based system, due to the fact that OWL is the most widely used programming language for defining knowledge models in agent-based systems. In order to implement a SWRL rule base into AAMS for transformer FD in the near future, in this study, SWRL was selected for interpreting a generated useful ARS as a SWRL rule base.

Briefly, SWRL is developed based upon the combination of OWL DL, OWL Lite and Unary/Binary Datalog RuleML sub-languages. With SWRL, it is allowed to write Horn-like rules [114] with OWL elements, *i.e.*, classes, properties and individuals. Then, the obtained SWRL rules are organised within an OWL ontology model. In recent years, SWRL has been successfully implemented into a number of ontology-based decision support systems, *e.g.*, [115] [116] and [117].

A SWRL rule is represented with an IF-THEN format, which is the same as the form of an association rule illustrated in (4.4.1). In the SWRL terminology, the antecedent and the consequent of an association rule are named as the rule body and the rule head of a SWRL rule, respectively. Moreover, the head or the body of a SWRL rule is consisted of one or more atoms, which are defined as attributes in an association rule. Compared with an ARS, a SWRL rule base offers several advantages in a RBES, which are shown as Figure 6.1 and listed as follows:

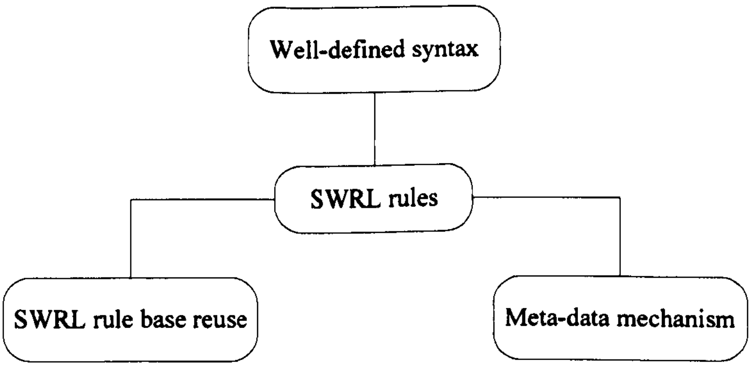


Figure 6.1: Features of a SWRL rule base

- Being as a subset of FOL, SWRL offers a logical representation format for rule descriptions. Thus, the syntax consistency of SWRL rules can be guaranteed;

- In SWRL, a meta-data mechanism is provided, with which the concepts defined in a SWRL rule set can be described with its related information, *i.e.*, the definition of a fault type, reasons causing an on-site transformer fault and the typical examples of a fault type and so on. The meta-data then can be used for explaining a detected fault to power engineers;
- Most significantly, a SWRL rule base can be effectively processed by a variety of published rule engines, which enables the SWRL rule base to be shared among different RBESs. As a result, for RBESs, the average expense on rule base rebuilding can be greatly reduced.

However, as mentioned above, SWRL is not bundled with any inference engine. In other words, a SWRL rule base cannot be used in a RBES directly. In order to evaluate the practical performance of a generated SWRL rule base for transformer FD, in this study, Jess was employed to develop a SRBES of transformer FD with a SWRL rule base. In the next section, a detailed introduction to RBES and Jess is provided.

6.3 Rule-based expert system and Java expert system shell

6.3.1 Rule-based expert system

Conventionally, a computer program is normally designed to solve a specific problem using a decision making logic. In such tasks, the employed knowledge is embedded as a part of the program code, and because of this, the program needs to be recoded once the knowledge is replaced or updated. Therefore, in recent years, a knowledge-based ES has been recommended as a suitable solution to tackle the above problem. Briefly, a knowledge-based ES is a system that can solve real-world problems using the knowledge extracted from the real-world. In this kind of systems, the small fragments of human knowledge are collected and stored in a knowledge base. Then, an ES is employed to reason through a decision making process based upon the established knowledge base. Compared with conventional decision making methods, one significant advantage of a knowledge-based ES is that more than one problem can be solved by the same ES if they are related to the domain of the knowledge base, employed in the ES. As a result, without the knowledge base having to be rebuilt for different problems, the average cost for solving a single decision making problem can be significantly reduced. Furthermore, an ES can explain the reasoning procedures of a decision making process detailedly and handle different levels of confidence and uncertainty [118].

As concluded in [119], some other important features of a knowledge-based ES have been summarised as Figure 6.2 and are introduced as follows:

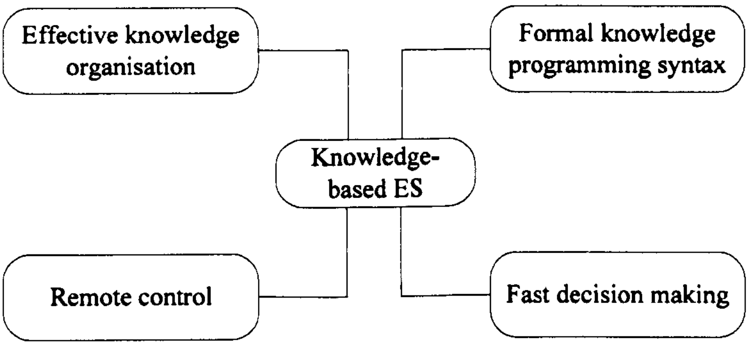

Effective knowledge organisation
Formal knowledge programming syntax
Knowledge-based ES
Remote control
Fast decision making

Figure 6.2: Features of a knowledge-based ES

- Human experience can be captured and preserved as knowledge in an ES. In a decision making process, a mathematical algorithm, *e.g.*, Rete network algorithm [120], is employed to automatically make use of the knowledge in an optimal way. This feature allows human experts to only focus on the method for organising the conditions of a decision making process as the suitable inputs of the ES and thus saves the expense for solving a specific problem;

- The knowledge stored in the knowledge base of an ES is represented with a computer readable language, in which, the syntax of the language is well defined. Compared with the raw knowledge of human experts, the knowledge base of the ES can be easily employed by a computer through a decision making process. Moreover, with the consistent knowledge representation syntax, the knowledge base can be easily maintained and updated;

- An ES can be embedded into a hardware, which is capable of being installed in a number of harmful locations and thus protects the health of human expertise;
- With the predefined knowledge and a computer, a decision making solution can be developed faster than that obtained by human expertise.

In recent years, the knowledge used in a knowledge-based ES has been normally represented as a set of decision making rules. Such an ES is thus called a RBES. The basic structure of a RBES is illustrated in Figure 6.3. As can be seen from the figure, there are four main modules involved in a RBES, *i.e.*, a working memory, a rule base, an inference engine and an execution engine. The functions of these modules are described as follows:

- The working memory, *i.e.*, fact base, is used to store the facts about the world, *e.g.*, a test DGA record being diagnosed;
- The rules used in a RBES are organised within the rule base;
- The inference engine provides a "black box" service, which performs reasonings over the rules from the rule base and the facts stored in the working memory. In the inference engine, the pattern matcher module is dedicated to selecting the rules that are applicable for diagnosing a fact set and then place the rules on the agenda module;
- The execution engine, sequently, is used to fire the rules located on the agenda module with a particular order, which is decided by the execution engine using a mathematical algorithm.

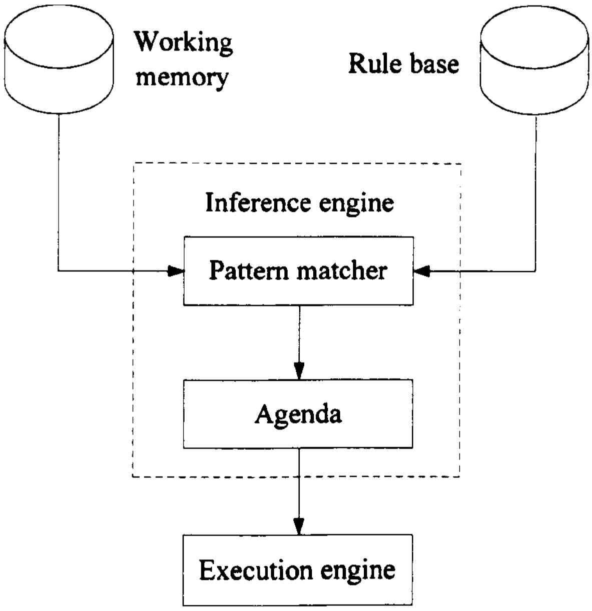


Figure 6.3: Architecture of a RBES

As introduced in Section 6.1, Jess has been employed to develop a transformer FD ES with a SWRL rule base, *i.e.* SRBES, which in turn was derived from a set of useful association rules generated in Chapter 5. Therefore, in the next subsection, Jess is briefly introduced in order to obtain a clear understanding of the SRBES development process.

6.3.2 Java expert system shell

Briefly, Jess is both a rule execution engine and a rule programming environment fully written in Java. Previously, a set of Jess-based research work has been taken, in which Jess was employed to execute SWRL rules [121] [122] [123] for solving decision making problems. In recent years, Jess has become a widely used tool for developing RBESs. The development of Jess started with the infrastructure of another ES shell, namely CLIPS [124], and it has finally established itself as a distinct ES software tool. Therefore, the syntax of Jess language is very similar to that used in CLIPS.

Briefly, Jess provides a Lisp-like [125] syntax and interpreter, with forward-chaining rules using the Rete network algorithm. The main features of Jess are illustrated as Figure 6.4 and summarised as follows:

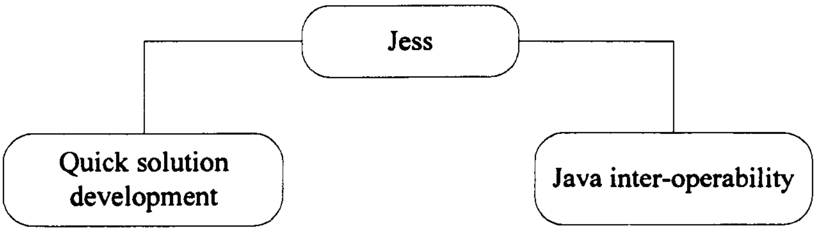


Figure 6.4: Features of Jess

- In a rule base, a large number of rules may exist, the size of which thus can bring a difficulty of suitably selecting rules for a specific decision making process. In such a situation, Jess can be proposed as a viable solution. Jess is a rule engine using a declarative paradigm, which can automatically select rules for a decision making process and then continuously apply them to the facts stored in a RBES fact base with a pattern matching function. Consequently, the obtained results are delivered back to the fact base and used for updating the fact base accordingly;

- On the other hand, Jess is also a general purpose programming language. With a well-defined Application Programming Interface (API), all Java libraries, Java codes and Java objects are processable in Jess. That is to say, Jess can be employed by other Java programs for dealing with decision making problems, and in turn, new functions written in Java can be easily added to Jess-based applications.

In a conclusion, a high flexible and high compatible rule execution platform can be provided by Jess. In this study, Jess was used to construct a SRBES with a SWRL rule base. Such an implementation process is described in the next subsections.

6.4 System architecture of a semantic web rule language rule-based expert system

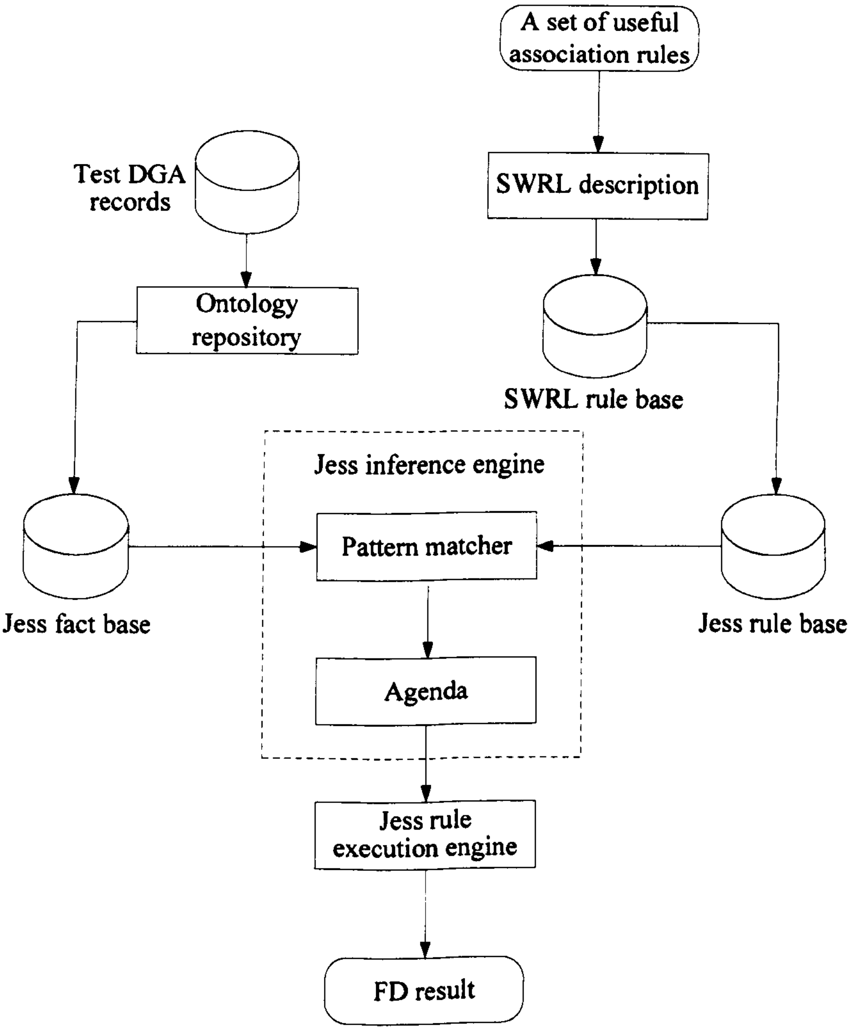


Figure 6.5: System architecture of a SRBES

As stated in Section 6.3.1, in a RBES, four modules are included, *i.e.*, a working memory (fact base), a rule base, an inference engine and an execution engine. Therefore, a SRBES was developed based upon these four modules and further extended with several other function modules. The basic structure of a

SRBES is shown as Figure 6.5. As can be seen from the figure, seven function modules in total, *i.e.*, a SWRL description module, the ontology repository of test DGA records, a SWRL rule base, a Jess fact base, a Jess inference engine, a Jess rule base and a Jess rule execution engine, are included in a SRBES. The services provided by the seven function modules are described as follows:

- SWRL description module: The SWRL description module is designed to interpret an input ARS as a SWRL rule base;

- Ontology repository of test DGA records: a test DGA record is composed of several gas ratios. In a SRBES, these gas ratios are firstly defined as a set of ontology classes. Then, a set of test DGA records is organised into a domain ontology model, *i.e.*, the ontology repository of test DGA records, and each of the records is represented as a set of OWL individuals of the ontology classes;

- SWRL rule base: this module is developed for the storage of SWRL rules derived from an ARS using the SWRL description module. In this rule base, each involved SWRL rule is represented as a set of OWL individuals of the ontology classes denoted in the SWRL rule base.

- Jess rule base and Jess fact base: as stated in Section 6.2, SWRL is not bundled with an inference engine. Thus, the derived SWRL rules cannot be directly utilised by a RBES in an FD process. In a SRBES, the Jess inference engine is employed to apply the SWRL rules existed in the SWRL rule base to the given test DGA records. However, the Jess inference engine can only match rules and test DGA records, which are represented with Jess processable formats. Therefore, in a SRBES, the SWRL rules from the SWRL rule base and the test DGA records stored in the ontology repository are represented as Jess rules and Jess facts respectively, before a transformer FD process. Then, the Jess rules and the Jess facts are temporarily stored into the Jess rule base and the Jess fact base for further operations, as indicated below.

- Jess inference engine: as can be seen from Figure 6.5, two modules, *i.e.*, the pattern matcher and the agenda, are involved in the Jess inference engine. In a transformer FD process with a SRBES, the patten matcher, with the Rete algorithm, is used to select Jess rules that are applicable for diagnosing a specific test DGA record of the Jess fact base. Then, the selected rules are placed on the agenda module.
- Jess rule execution engine: finally, the selected rules are fired by the Jess rule execution engine for diagnosing the test DGA record. The derived result then is represented as OWL knowledge and transferred back to the ontology repository of test DGA records for updating the ontology repository.

In the next subsection, the development process of a SRBES is described in detail.

6.5 Development of a semantic web rule language rule-based expert system for transformer fault diagnosis

In this section, the development process of a SRBES is illustrated in detail, regarding the architecture of a SRBES presented in Section 6.4.

6.5.1 Defining ontology elements for developing a semantic web rule language rule base

A detailed introduction to the Ontology technique is provided in Section 2.4. Generally, three kinds of elements are involved in creating an OWL ontology model, *i.e.*, class, property and individual. For a domain ontology model, a class represents a group of objects employed in the domain. In a class, a set of individuals, which are instances or members of the class, is included. Furthermore, subclasses may be contained in the class as well. A property of an ontology model is denoted as a named relationship assigned between a pair of classes or from a class to a data value, which also can be inherited by the individuals involved in the classes.

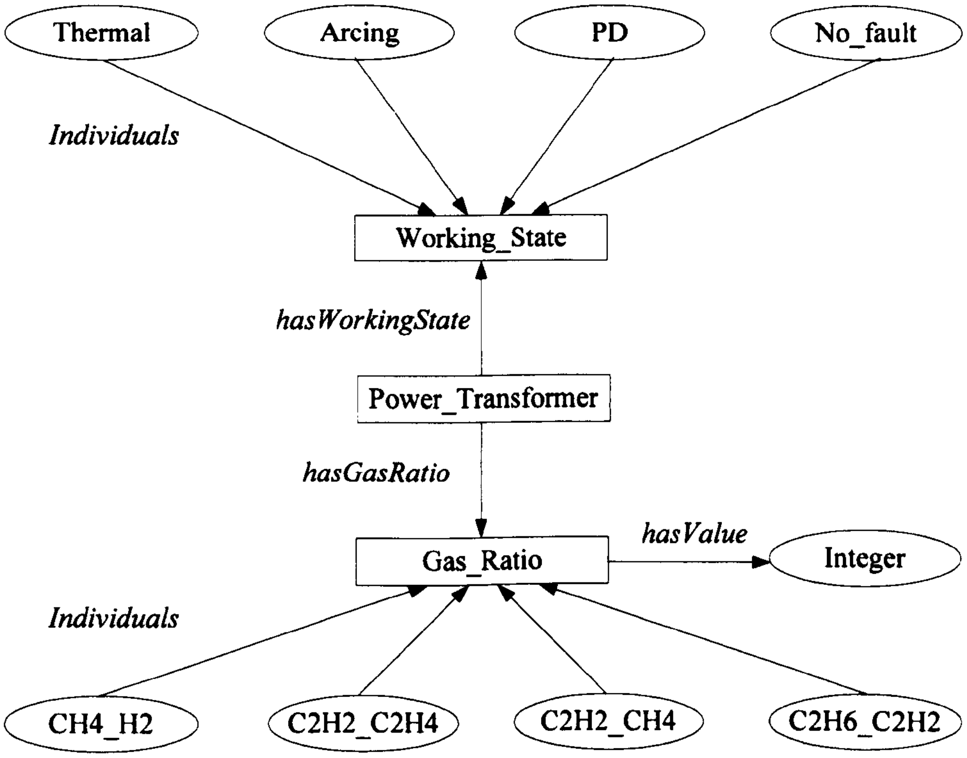


Figure 6.6: Ontology elements used for interpreting an association rule generated using the training data set of D_{Dor} as a SWRL rule set

In order to interpret a set of association rules as SWRL rules, which was generated from the training set of D_{Dor} or D_{Rog} in Chapter 5, a set of ontology classes, individuals and properties were defined in this study. The ontology elements, used to describe the rules of an ARS generated from the training set of D_{Dor}, are illustrated as Figure 6.6. As can be seen from the figure, in total three classes are involved, *i.e.*, Working_State, Power_Transformer, Gas_Ratio. In Working_State, four individuals, namely Thermal, Arcing, Partial_Diacharge and No_Fault were denoted. Meanwhile, four individuals were defined in Gas_Ratio, *i.e.*, CH_4_H_2, C_2H_2_C_2H_4, C_2H_2_CH_4 and C_2H_6_C_2H_2.

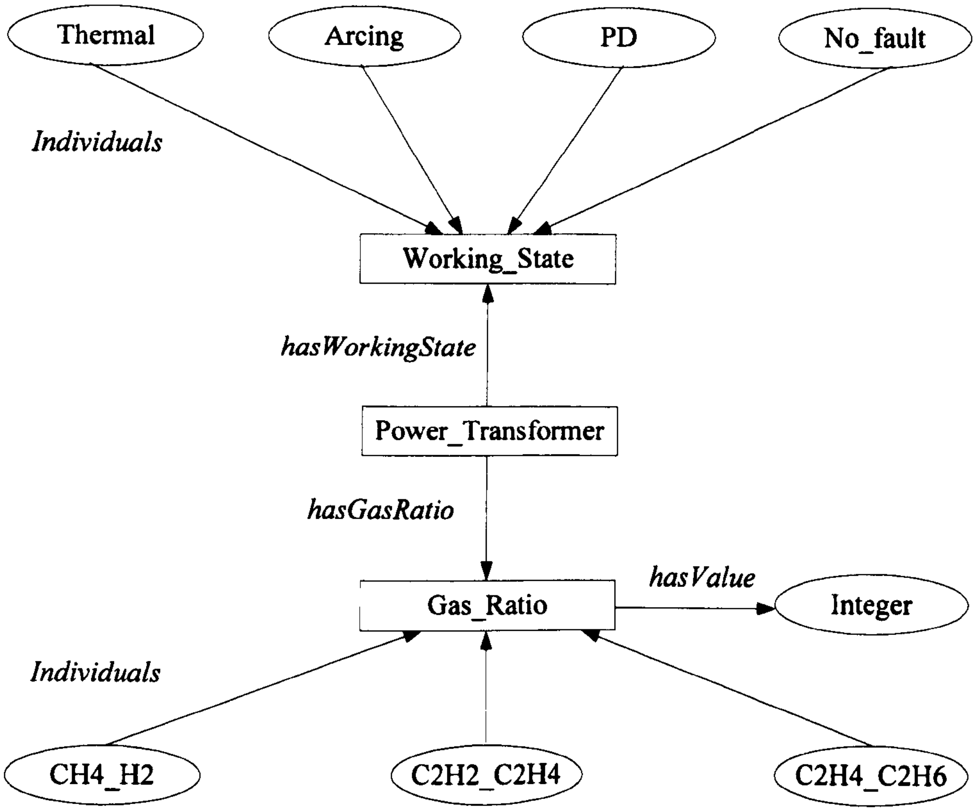


Figure 6.7: Ontology elements used for interpreting an association rule generated using the training data set of D_{Rog} as a SWRL rule set

Moreover, a number of properties, *i.e.*, *hasWorkingState*, *hasGasRatio*, *hasValue* and *Individuals*, were also coined for semantically denoting the relationships among the defined OWL classes and individuals.

On the other hand, the ontology elements, used for interpreting a rule set generated from the training set of D_{Rog} into a SWRL rule base, are shown in Figure 6.7. As shown in the figure, all the elements defined are the same as that illustrated in Figure 6.6, except that only three individuals were defined for Gas_Ratio, *i.e.*, $CH_4\_H_2$, $C_2H_2\_C_2H_4$ and $C_2H_4\_C_2H_6$.

With all the ontology elements introduced above, a set of association rules derived from Chapter 5 then can be interpreted as a SWRL rule base using SWRLTab [126], which is described detailedly in the next subsection.

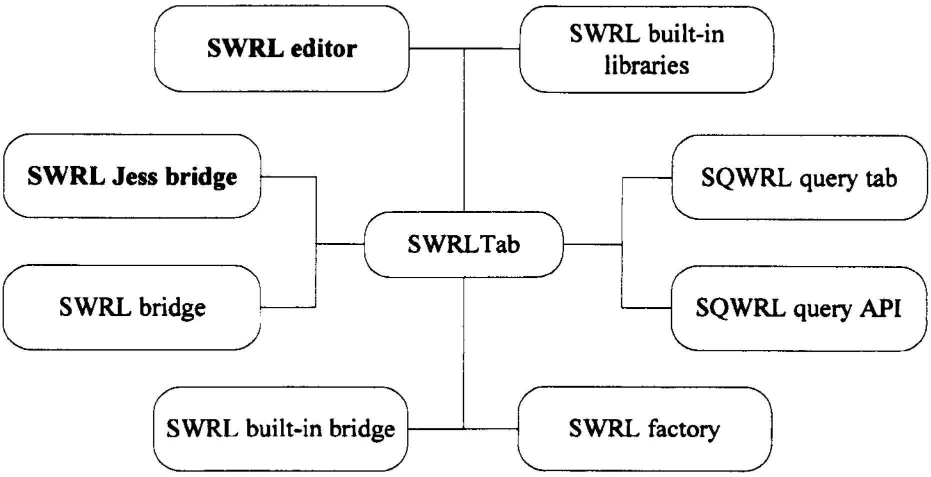


Figure 6.8: SWRLTab components

6.5.2 Introduction to SWRLTab

In this study, SWRLTab, developed by Connor, was employed for creating a SWRL rule base with a set of useful association rules. SWRLTab is a Protege plug-in and offers a SWRL development environment in Protege-OWL [67]. With SWRLTab, the editing of a SWRL rule is supported. Meanwhile, a rule engine interaction interface is provided by SWRLTab, with which SWRLTab can interact with various rule engines and thus allows SWRL rules to be executed by the rule engines. Moreover, with SWRLTab, user-defined methods can also be employed when creating a SWRL rule base.

As illustrated in Figure 6.8, in SWRLTab, several components are included, *e.g.*, a SWRL editor, a SWRL Jess bridge, a SWRL bridge, a SWRL built-in bridge, SWRL built-in libraries, a SQWRL query tab, a SQWRL query API and a SWRL factory. In this study, two of the components were employed for the construction of a SRBES, *i.e.*, the SWRL editor and the SWRL Jess bridge. The functions of these two modules are described as follows:

- SWRL editor: this is a Graphical User Interface (GUI) of SWRLTab. The SWRL editor allows users to interactively create, read and edit SWRL rules. A SWRL rule base can be developed using the SWRL editor and saved as a domain ontology model, in which each involved SWRL rule is described as a set of OWL individuals.

- SWRL Jess bridge: this module provides a bridge between an OWL ontology model containing SWRL rules and a Jess rule engine. With the bridge, the SWRL rules and the test DGA records stored in the test record ontology repository of a SRBES can be represented as Jess rules and sets of Jess facts, respectively. Sequently, the Jess facts can be diagnosed by the derived Jess rules via the Jess rule execution engine.

6.5.3 System construction

In this subsection, the construction process of a SRBES according to the architecture of a SRBES illustrated in Section 6.4 is presented. For comparison reasons, two ARSs, generated using the training sets of D_{Dor} and D_{Rog} in Chapter 5 respectively, were firstly employed to generate two different SWRL rule bases. Then, two different SRBESs were developed based upon the SWRL rule bases with Jess.

As illustrated in Table 5.2 concerning D_{Dor}, in total 16 different sets of association rules were generated. The highest FD accuracy, *i.e.*, 88.14%, was achieved by the rule sets generated at five different support-confidence value points, *i.e.*, (1, 1%); (1, 15%); (1, 35%); (8, 1%) and (8, 15%). The rule set, generated at the support-confidence value point (1, 1%), was named as ARS_{Dor} and selected from these five rule sets for developing a SRBES, which was defined as $SRBES_{\text{Dor}}$. This is due to that compared with the other four ARSs, this rule set contains the largest number of rules. Therefore, more kinds of transformer data sets may be correctly diagnosed by $SRBES_{\text{Dor}}$ established based upon this rule set, in comparison with that obtained using the other rule sets. As a result, a higher FD accuracy may be obtained. For the same reasons, the rule set obtained at the support-confidence value point (1, 1%) in Table 5.3, was denoted as ARS_{Rog} and chosen to develop the other SRBES, namely $SRBES_{\mathrm{Rog}}$.

For illustrative purposes, the construction process of $SRBES_{\mathrm{Dor}}$ using ARS_{Dor} is demonstrated as below. As illustrated in Section 5.5, in an FD process using ARS_{Dor}, more than one rule may be suitable for diagnosing a given test DGA record. Thus, an optimal rule selection method was implemented to select the most accurate rule for diagnosing the test DGA record. However, this kind of rule selection process is time-consuming, which may reduce the system performance of $SRBES_{\mathrm{Dor}}$ if employed. Therefore, in the study, ARS_{Dor} was pruned again using the optimal rule selection method before it was used to generate a SWRL rule base. In the rule set elimination process, the rules with the same antecedent were treated as similar rules and saved in a similar rule set. Then, the rule with the highest fitness value was treated as a useful rule and delivered back to ARS_{Dor}. The other rules from the similar rule set were then considered to be useless rules and thus pruned. The pruned ARS_{Dor} was composed of 24 association rules, with which a SWRL rule set then was generated and delivered to the SWRL rule base of $SRBES_{\mathrm{Dor}}$.

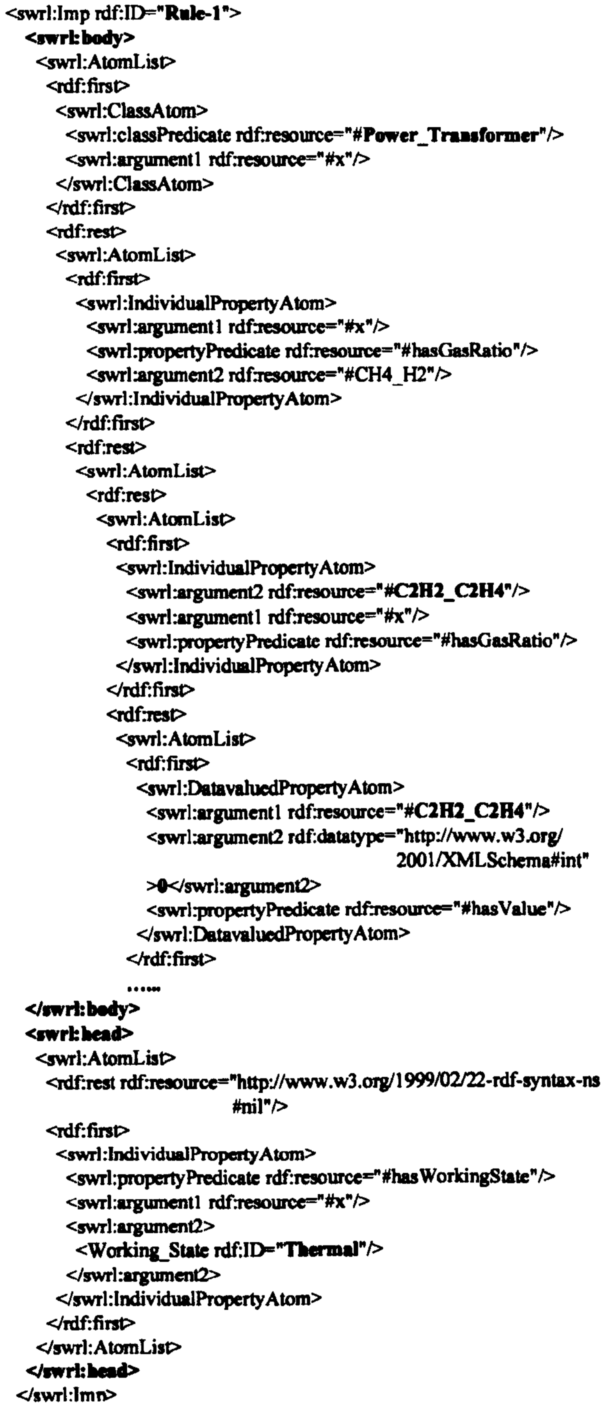
In order to obtain an in-depth understanding of a SWRL rule generation process, a rule namely Rule-1, selected from ARS_{Dor}, is given to demonstrate such a SWRL rule generation process. Supposing that, Rule-1 was provided as (6.5.1), in which the gas values were presented as binary values as defined in Table 4.2. With SWRLTab, Rule-1 then was interpreted as a SWRL rule, shown as (6.5.2).

Rule-1: **If:** CH4_H2 = 2;
and C2H2_C2H4 = 0;
and C2H2_CH4 = 0;
and C2H6_C2H2 = 1.
Then: Thermal. (6.5.1)

$$
\begin{aligned}
\textbf{Rule-1}:\ & \text{Power\_Transformer}(?\text{x}) \\
\wedge\ & \textit{hasGasRatio}(?\text{x}, \text{CH4\_H2}) \\
\wedge\ & \textit{hasValue}(\text{CH4\_H2}, 2) \\
\wedge\ & \textit{hasGasRatio}(?\text{x}, \text{C2H2\_C2H4}) \\
\wedge\ & \textit{hasValue}(\text{C2H2\_C2H4}, 0) \\
\wedge\ & \textit{hasGasRatio}(?\text{x}, \text{C2H2\_CH4}) \\
\wedge\ & \textit{hasValue}(\text{C2H2\_CH4}, 0) \\
\wedge\ & \textit{hasGasRatio}(?\text{x}, \text{C2H6\_C2H2}) \\
\wedge\ & \textit{hasValue}(\text{C2H6\_C2H2}, 1) \\
\rightarrow\ & \textit{hasWorkingState}(?\text{x}, \text{Thermal.})
\end{aligned}
\qquad (6.5.2)
$$

As introduced above, a SWRL rule is saved as a set of OWL individuals of a SWRL domain ontology model. Thus, in the SWRL rule base, the above SWRL rule, *i.e.*, Rule-1, was described with an OWL ontology segment, as shown in Figure 6.9.

Subsequently, in order to implement this SWRL rule into a transformer FD process, the SWRL Jess bridge was employed to transform Rule-1 into a Jess rule firstly, represented as (6.5.3). Then, the obtained Jess rule was stored in the Jess rule base of $SRBES_{\text{Dor}}$ and further utilised to diagnose a given test



```
<swrl:Imp rdf:ID="Rule-1">
 <swrl:body>
  <swrl:AtomList>
   <rdf:first>
    <swrl:ClassAtom>
     <swrl:classPredicate rdf:resource="#Power_Transformer"/>
     <swrl:argument1 rdf:resource="#x"/>
    </swrl:ClassAtom>
   </rdf:first>
   <rdf:rest>
    <swrl:AtomList>
     <rdf:first>
      <swrl:IndividualPropertyAtom>
       <swrl:argument1 rdf:resource="#x"/>
       <swrl:propertyPredicate rdf:resource="#hasGasRatio"/>
       <swrl:argument2 rdf:resource="#CH4_H2"/>
      </swrl:IndividualPropertyAtom>
     </rdf:first>
     <rdf:rest>
      <swrl:AtomList>
       <rdf:rest>
        <swrl:AtomList>
         <rdf:first>
          <swrl:IndividualPropertyAtom>
           <swrl:argument2 rdf:resource="#C2H2_C2H4"/>
           <swrl:argument1 rdf:resource="#x"/>
           <swrl:propertyPredicate rdf:resource="#hasGasRatio"/>
          </swrl:IndividualPropertyAtom>
         </rdf:first>
         <rdf:rest>
          <swrl:AtomList>
           <rdf:first>
            <swrl:DatavaluedPropertyAtom>
             <swrl:argument1 rdf:resource="#C2H2_C2H4"/>
             <swrl:argument2 rdf:datatype="http://www.w3.org/
                                              2001/XMLSchema#int"
             >0</swrl:argument2>
             <swrl:propertyPredicate rdf:resource="#hasValue"/>
            </swrl:DatavaluedPropertyAtom>
           </rdf:first>

           ......
 </swrl:body>
 <swrl:head>
  <swrl:AtomList>
   <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns
                          #nil"/>
   <rdf:first>
    <swrl:IndividualPropertyAtom>
     <swrl:propertyPredicate rdf:resource="#hasWorkingState"/>
     <swrl:argument1 rdf:resource="#x"/>
     <swrl:argument2>
      <Working_State rdf:ID="Thermal"/>
     </swrl:argument2>
    </swrl:IndividualPropertyAtom>
   </rdf:first>
  </swrl:AtomList>
 </swrl:head>
</swrl:Imn>
```

Figure 6.9: An ontology program segment of a SWRL rule "Rule-1"

DGA record.

$$
\begin{aligned}
&\text{(defrule } \textbf{Rule-1} \\
&Power\_Transformer \text{ (name ?x))} \\
&(hasGasRatio \text{ ?x CH4\_H2)} \\
&(hasValue \text{ CH4\_H2 2)} \\
&(hasGasRatio \text{ ?x C2H2\_C2H4)} \\
&(hasValue \text{ C2H2\_C2H4 0)} \\
&(hasGasRatio \text{ ?x C2H2\_CH4)} \\
&(hasValue \text{ C2H2\_CH4 0)} \\
&(hasGasRatio \text{ ?x C2H6\_C2H2)} \\
&(hasValue \text{ C2H6\_C2H2 1)} \\
&\Rightarrow \text{(assert } (hasWorkingState \text{ ?x Thermal))}
\end{aligned}
\qquad (6.5.3)
$$

With the rule interpretation method described above, the other association rules from the pruned ARS_{Dor} were also transformed into Jess rules and stored in the Jess rule base of $SRBES_{\mathrm{Dor}}$. Then, a transformer FD task can be carried out using $SRBES_{\mathrm{Dor}}$, with a given test DGA record. With the same method, $SRBES_{\mathrm{Rog}}$ was derived based upon ARS_{Rog} as well. In the next section, tests used for verifying the FD performance of these two SRBESs are demonstrated. Obtained results then are presented in the section.

6.6 Experiments

6.6.1 Experiment schemes

In this subsection, the tests implemented for evaluating the FD accuracies of $SRBES_{\mathrm{Dor}}$ and $SRBES_{\mathrm{Rog}}$ are presented, respectively. The test data set of D_{Dor} and D_{Rog}, reported in Section 5.2, were selected for evaluating the FD accuracies of $SRBES_{\mathrm{Dor}}$ and $SRBES_{\mathrm{Rog}}$, respectively. As explained in Section 5.2, for each of the test data sets, in total 177 test DGA records were included.

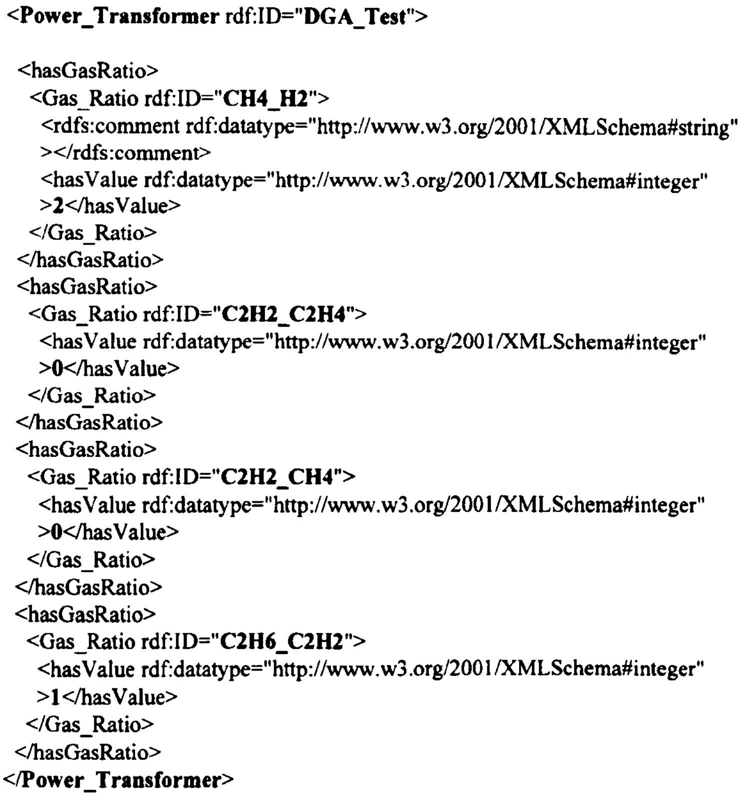
In the following subsections, firstly, in order to provide an in-depth understanding of a transformer diagnosing process with a SRBES, a practical FD example using a SRBES is demonstrated firstly. Subsequently, the test results of $SRBES_{\mathrm{Dor}}$ and $SRBES_{\mathrm{Rog}}$, achieved by diagnosing the test data sets of D_{Dor} and D_{Rog} respectively, are illustrated. Finally, the generated results are compared with that obtained by using ARS_{Dor} and ARS_{Rog}, separately.

6.6.2 A fault diagnosis scenario with a semantic web rule language rule-based expert system

For illustration purposes, a practical FD scenario using $SRBES_{\mathrm{Dor}}$ is demonstrated in this subsection. Supposing that, a test DGA record, namely DGA_Test was provided as below:

$$\mathrm{CH_4/H_2} = 2; \qquad \mathrm{C_2H_2/C_2H_4} = 0;$$
$$\mathrm{C_2H_2/CH_4} = 0; \qquad \mathrm{C_2H_6/C_2H_2} = 1.$$

Firstly, DGA_Test was described as a set of OWL ontology individuals, shown as Figure 6.10, and stored in the ontology repository of test DGA records, as illustrated in Figure 6.5. Then, with the SWRL Jess bridge module introduced in Section 6.5.2, DGA_Test was transformed into a set of Jess facts and saved in the Jess fact base of $SRBES_{\mathrm{Dor}}$. Subsequently, the Jess rule that was applicable for diagnosing this set of Jess facts, *i.e.*, the rule (6.5.3), was selected from the Jess rule base of $SRBES_{\mathrm{Dor}}$ by the patten matcher module of the Jess inference engine and then placed on the agenda module, as shown in Figure 6.5. Subsequently, the rule (6.5.3) was fired by the Jess rule execution engine for diagnosing the derived Jess fact set. The diagnosis result was obtained as thermal, which was the same as the on-site diagnosis result. Finally, the result was represented as OWL ontology knowledge and transferred back to the ontology repository of test DGA records for further operations.



```
<Power_Transformer rdf:ID="DGA_Test">

 <hasGasRatio>
  <Gas_Ratio rdf:ID="CH4_H2">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   ></rdfs:comment>
   <hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"
   >2</hasValue>
  </Gas_Ratio>
 </hasGasRatio>
 <hasGasRatio>
  <Gas_Ratio rdf:ID="C2H2_C2H4">
   <hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"
   >0</hasValue>
  </Gas_Ratio>
 </hasGasRatio>
 <hasGasRatio>
  <Gas_Ratio rdf:ID="C2H2_CH4">
   <hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"
   >0</hasValue>
  </Gas_Ratio>
 </hasGasRatio>
 <hasGasRatio>
  <Gas_Ratio rdf:ID="C2H6_C2H2">
   <hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"
   >1</hasValue>
  </Gas_Ratio>
 </hasGasRatio>
</Power_Transformer>
```

Figure 6.10: An ontology program segment of a test DGA record

Table 6.1: FD accuracies (%) of $SRBES_{\text{Dor}}$, $SRBES_{\text{Rog}}$, ARS_{Dor} and ARS_{Rog}

	$SRBES_{\text{Dor}}$	$SRBES_{\text{Rog}}$	ARS_{Dor}	ARS_{Rog}
Correctly diagnosed	156	162	156	162
Wrongly diagnosed	17	15	17	15
Not processable	4	0	4	0
Accuracy	**88.14**	**91.53**	**88.14**	**91.53**

6.6.3 Test results and discussion

The FD accuracies of $SRBES_{\text{Dor}}$, $SRBES_{\text{Rog}}$, evaluated using 177 test DGA records of D_{Dor} and D_{Rog} respectively, are illustrated in Table 6.1. Mean-

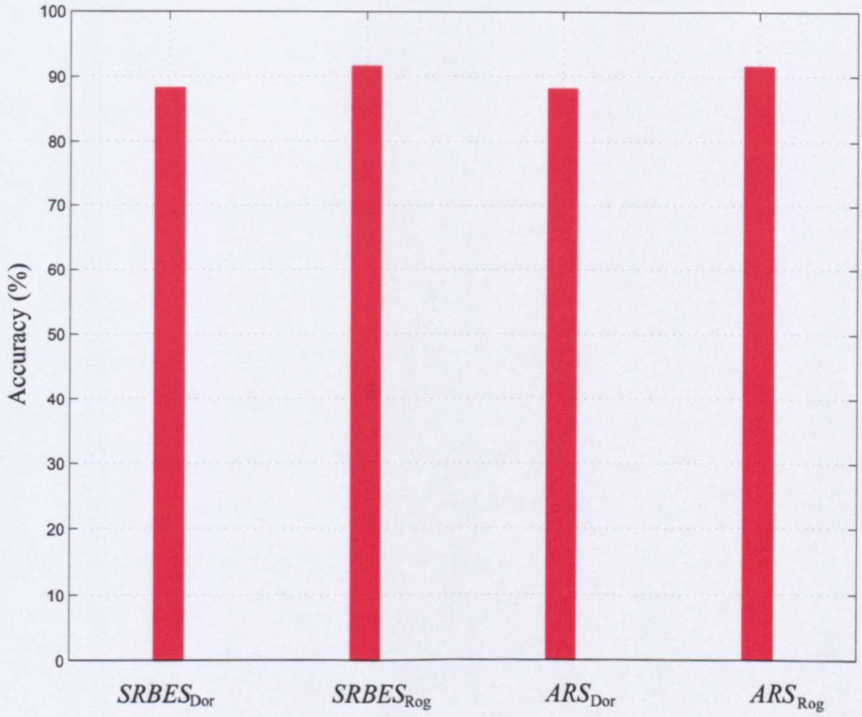


Figure 6.11: FD accuracy histograms of $SRBES_{\mathrm{Dor}}$, $SRBES_{\mathrm{Rog}}$, ARS_{Dor} and ARS_{Rog}

while, a histogram generated with these accuracy values is shown in Figure 6.11, in which columns one, two, three and four represent the FD accuracies of $SRBES_{\mathrm{Dor}}$, $SRBES_{\mathrm{Rog}}$, ARS_{Dor} and ARS_{Rog}, respectively. As can be seen from Table 6.1, the FD accuracy of $SRBES_{\mathrm{Dor}}$ was achieved as 88.14%, which was the same as that of ARS_{Dor}-based FD system, as demonstrated in Table 5.2. Meanwhile, the FD accuracy of $SRBES_{\mathrm{Rog}}$ was derived as 91.53%, which was the same as that obtained using ARS_{Rog}, as indicated in Table 5.3. Hence, with the obtained results, the capability of a SRBES on transformer FD has been demonstrated.

6.7 Conclusion

In this chapter, the development of a RBES for transformer FD, *i.e.,* a SRBES, using a set of useful association rules generated in Chapter 5, has been presented. A brief introduction to SWRL and Jess was given firstly. Meanwhile, several advantages of using a SWRL rule base in a RBES, which are not achievable by employing an ARS, were reported. Then, the system structure of a SRBES was described. The function of each module existed in a SRBES was explained in detail afterwards. Subsequently, the development process of a SRBES using a SWRL rule base and Jess was described. With the proposed implementation method, a practical FD scenario of a SRBES was demonstrated. The final results showed that, with the same test DGA records, a SRBES has achieved the same FD accuracy, compared with that obtained using an ARS-based FD system. As a conclusion, a SRBES, using a SWRL rule base, can be used as a viable solution for FD of power transformers.

Chapter 7

Implementation of An Agent-Based Substation Asset Management System

7.1 Introduction

Served as two main function modules of AAMS, the substation AM approaches introduced in previous chapters, *i.e.*, an ER-based ODSE for IR of power substations and an association rule-based transformer FD system, were implemented in AAMS. In the development of AAMS, the thesis author was dedicated to the design of the AAMS structure. The programming process for integrating the above two AM modules into AAMS then was carried out by the other researcher.

At the current stage, AAMS can be effectively used for IR and FD of power substations only. However, other functions of AAMS, concerning the different AM aspects of a power system, are currently being developed and will be added to AAMS in the near future.

The rest of this chapter is organised as follows: Section 7.2 states the motivation of developing AAMS based upon Multi-Agent System (MAS). An introduction to the Agent technology and MAS is given in Section 7.3. In

Section 7.4, firstly, the system architecture of AAMS is described. Then, the functions of the agents, employed in AAMS, are presented in detail. A set of AAMS implementation tests is demonstrated in Section 7.5, followed by a result discussion. The conclusions are addressed in Section 7.6.

7.2 Motivation of developing an agent-based asset management system

The drawbacks of the conventional AM system of power systems are summarised as follows:

- With the growing complexity of power systems, the structure of a traditional power AM system is getting more and more complicated. Therefore, the connection flexibility and robustness of the AM system is greatly restricted, due to its centralised control format.
- A traditional AM system is usually developed based on the power networks with fixed topologies. In the case that the structure of a power network is changed, there is no efficient way to update the existing AM system without any reconfigurations.
- The functions of a traditional power AM system are operated based on pre-defined procedures. The knowledge of the function modules cannot be rapidly and accurately updated, when a new function module is added into the existing system [127]. As a result, the coordination and cooperation capabilities of the system is greatly reduced.

Therefore, AAMS has been investigated to tackle the above problems and is introduced as the following content of this chapter. Briefly, an agent is an encapsulated software which is situated in computer operating systems, and is capable of flexible, autonomous actions in those systems in order to meet its design objectives [128]. MAS is an agent organisation or in other words,

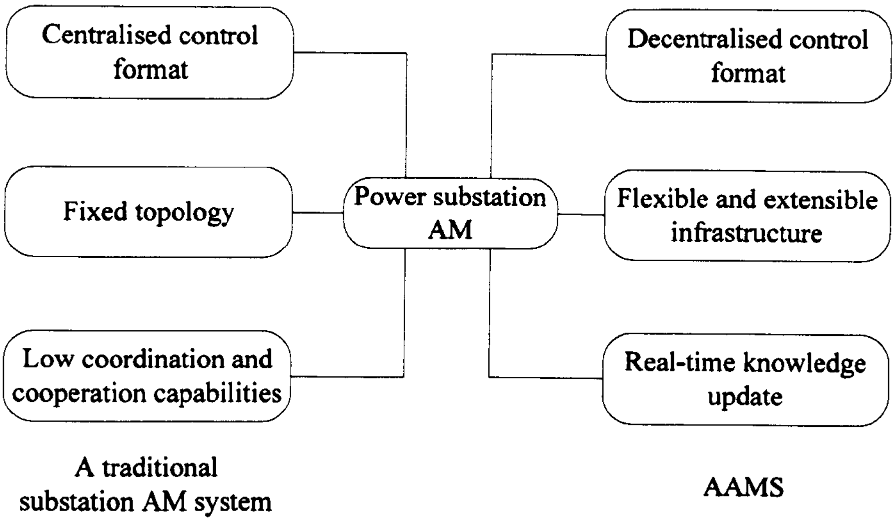

Centralised control format
Fixed topology
Low coordination and cooperation capabilities
A traditional substation AM system
Power substation AM
Decentralised control format
Flexible and extensible infrastructure
Real-time knowledge update
AAMS

Figure 7.1: Feature comparison between a traditional AM system and AAMS

as some artificial societies or organisations of agents [129]. With a local problem solving capability and an explicit agent negotiation mechanism, MAS has brought a growing concern in the field of distributed system control [130] [131]. Practically, in an MAS-based distributed control system, a collection of autonomous processors, *i.e.*, agents, and data storages can cooperatively interact to achieve an overall goal [132]. As a type of computer network technology, MAS was firstly introduced into power systems by the *e*-Automation laboratory at the University of Liverpool, in 2003, where it was proposed as a distributed industrial platform of power system automation [9].

The main benefits of a power system that can be obtained with AAMS, developed based upon MAS, are illustrated as Figure 7.1 and explained as follows:

- Firstly, there is no need for exclusive central control and can perform tedious, repetitive, time consuming, and analytically complex tasks more accurately and reliably.

- Secondly, in AAMS, a flexible and extensible infrastructure is provided that allows different tasks to be programmed in separate agents [133]. Particularly, a new function can be added into AAMS by just developing a specific agent in AAMS and without interrupting the running AAMS.

- Thirdly, with an agent communication approach, generated system information can be rapidly and accurately shared between agents. Therefore, a real-time knowledge update can be achieved by AAMS.

In the next section, the Agent technique and MAS are introduced in detail.

7.3 Intelligent agent and multi-agent system

The main purpose of this chapter is to illustrate the practical performance of AAMS on power substation AM. However, a brief introduction to the Agent technique and MAS is provided as below, which makes a much easier understanding of the working processes of AAMS.

The concept of an agent can be traced back to the early days of research into distributed AI in the 1970s. Hewitt developed an actor model [134] that was a self-contained, interactive and concurrently-executing object, possessing internal states and communication capabilities. The object, firstly regarded as a computational agent, had some encapsulated internal state and could respond to messages from other similar objects. Since 1990 there has evidently been another distinct strand to the research and development work on software agents. Wooldridge and Jennings [135] [136] firstly proposed the intelligent agent concept which complemented and broadened the typology of agents being investigated by agent researchers.

As mentioned above, a number of agents may be involved in MAS. Practically, the agents of MAS are mainly responsible for perceiving the state of their operating environment, updating their own knowledge, deciding future actions and finishing the assigned tasks and so on. Meanwhile, the agents can cooperate with each other to fulfill an overall goal. With the purpose of enabling seamless agent interactions in MAS, a standardisation agent development architecture is highly required. Previously, many efforts have been made to develop an agent development standard, *e.g.*, Homer [137], Belief-Desire-Intention (BDI) [138] and Foundation for Intelligent Physical Agent (FIPA) specification [139] and so on. In these standards, the FIPA specification has been accepted by IEEE computer science as the agent development standard. The core mission of the FIPA specification is to facilitate the interoperations between a pair of agents, which are situated in one or multiple MASs.

Java Agent DEvelopment framework (JADE) [140] is an agent development environment, which is one of the implementations of the FIPA agent specification and established based upon Java programming language. With JADE, developed agents are FIPA compliant. Hence, the functions defined in the FIPA specification can be programmed into the agents. In a JADE agent system, agents can be allocated in several hosts and interact with one another via a JADE agent platform. As shown in Figure 7.2, on a JADE platform, several containers may exist, out of which, only one can be registered as the main container. Then, an Agent Management System (AMS) agent and a Directory Facilitator (DF) agent are automatically created in the main container for initiating MAS. In addition, the host which owns the main container is treated as the server of the agent platform. In all the containers registered to the agent platform, a complete run-time environment for agent execution is provided by JADE.

In JADE, an individual agent is defined as a single thread with a globally unique Agent Identification (AID). Behaviour abstractions are used to model the multiple simultaneous activities that an agent is able to perform. Communication mechanisms among agents adheres to the FIPA specification. Moreover, FIPA-specified Agent Communication Language (ACL) [141] is used as a standard message language for agent communications. Particularly, an ACL message is formatted in the main fields as follows [141]:

- the *sender* of the message;
- the list of *receivers*;
- the *performative* indicating the intentions of the *sender* by sending the

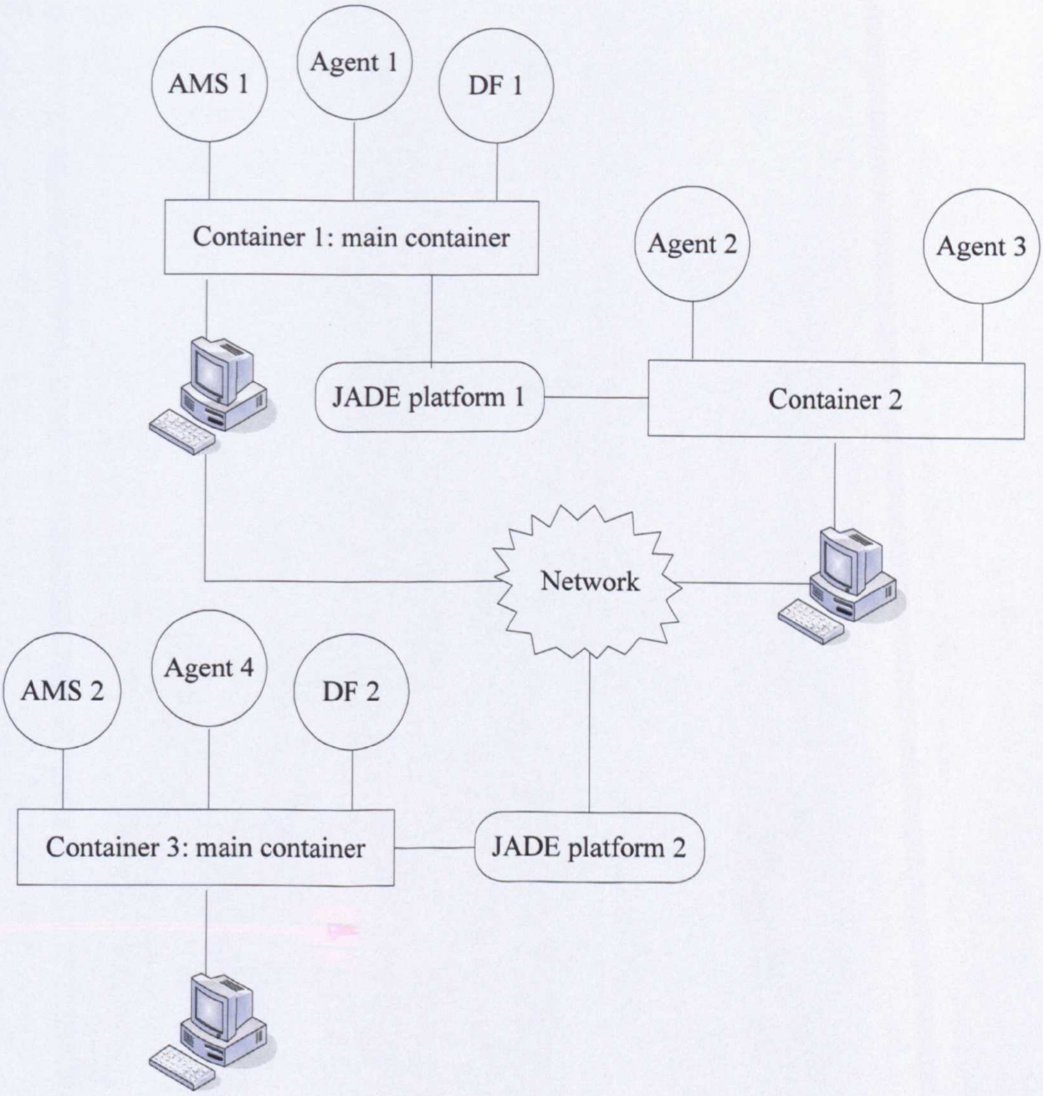


Figure 7.2: JADE agent platforms and containers

message; and

- the *content*, *i.e.*, the actual information included in the message.

AAMS was developed with JADE. In AAMS, the development of each software agent has strictly followed the FIPA specifications, and the messages delivered by each agent were completely structured by the FIPA ACL message formats. In the next section, the system architecture of AAMS is illustrated. Moreover, the functions of AAMS components are introduced detailedly.

7.4 System architecture and components

7.4.1 Architecture of an agent-based asset management system

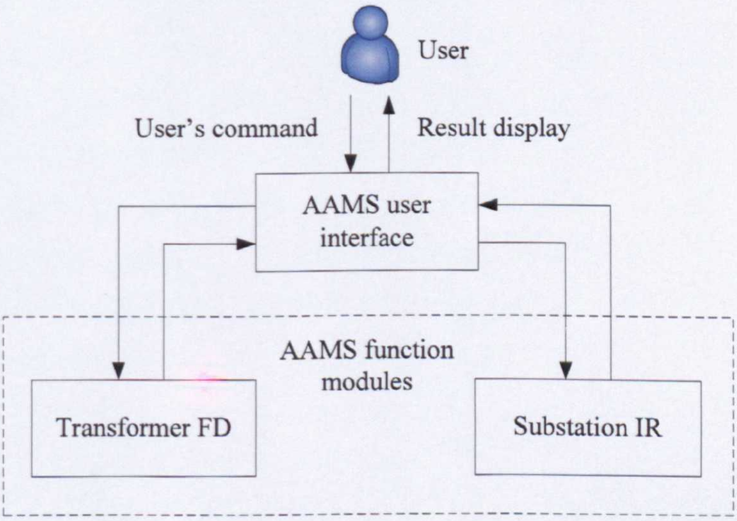


Figure 7.3: System architecture of AAMS

Figure 7.3 shows the overall architecture of AAMS. As can be seen from the figure, an AAMS user interface, a transformer FD module and a substation IR module were developed in AAMS. The AAMS user interface is capable for assisting interactions between a common user and AAMS, *e.g.*, inputting a query for IR, displaying a query result or the working state of a transformer. On the other hand, for each of the illustrated AM modules, a set of agents was constructed. With the cooperation of these agents, AAMS can be employed for IR of substations and FD of transformers, respectively. In the following subsections, the two function modules of AAMS are described in detail, separately.

7.4.2 Transformer fault diagnosis module

The architecture of the AAMS transformer FD module is presented in Figure 7.4. As illustrated in the figure, four types of software agents were developed in this module, *i.e.*, a Data Acquisition Agent (DAA), a Data Interpretation Agent (DIA), a Rule Base Agent (RBA) and an FD Agent (FDA). Moreover, a set of other components were also employed, *i.e.*, a transformer database, an association rule base and a fault record database.

The transformer database is used for saving the gas data collected from a power transformer. An ARS, obtained in Chapter 5, is utilised as a rule base for supporting transformer FD. Moreover, new transformer fault records, collected from the transformers of a power system, are stored in the fault record database and then added to a training DGA data set of ARM for generating potentially new classification rules. In the following subsections, the functions of the software agents are introduced, respectively.

DAA

As shown in Figure 7.5, DAA is used to receive real-time key gas data from the sensors which are installed in a transformer and then to save this data in the transformer database for backup, with its *DataCollectionBehaviour*. The received real-time key gas data are categorised into different columns in the database in accordance with their names, *i.e.*, H_2, CH_4, C_2H_6, C_2H_4, C_2H_2, CO and CO_2 and so on.

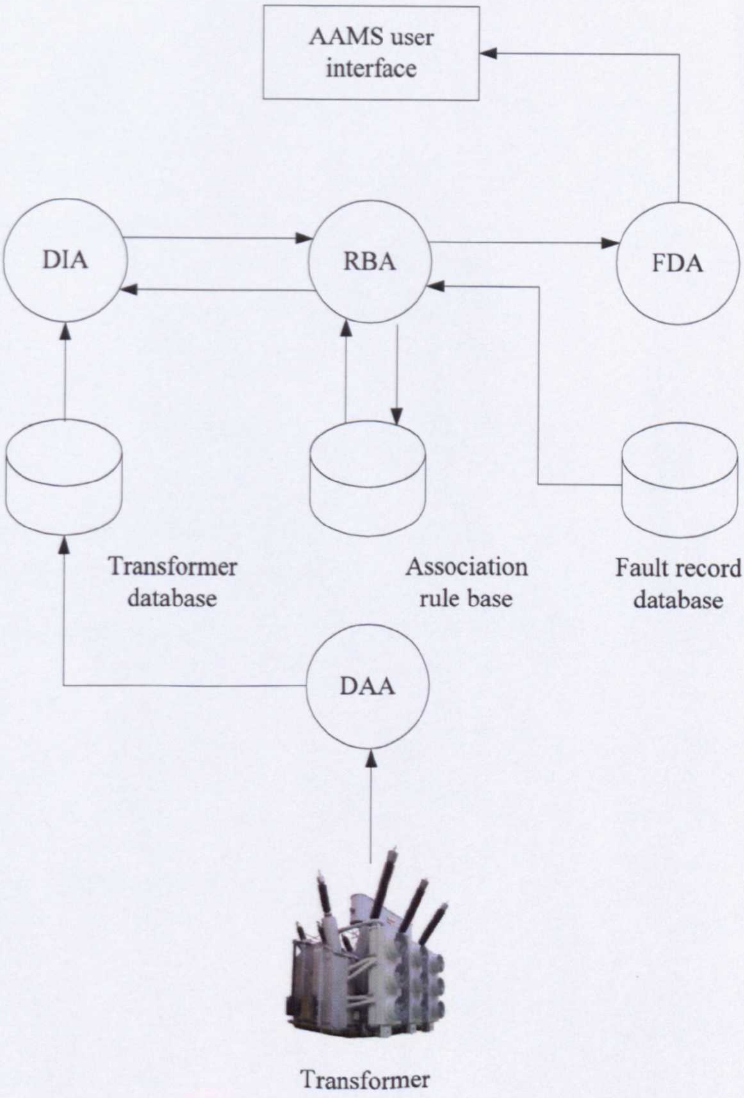


Figure 7.4: Architecture of transformer FD module in AAMS

DIA

DIA can receive a request message from RBA, in which the data types used in an FD process are defined, *i.e.*, gas/gas ratio types, according to the requirements of an AAMS rule base. As indicated in Figure 7.6, according to the required data types, *DataProcessingBehaviour* enables DIA to extract data

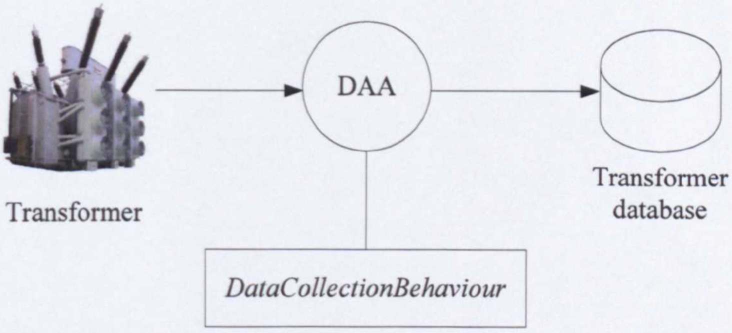


Figure 7.5: DAA behaviour

from the transformer database and calculate the gas ratios in accordance with the defined data types, if necessary. Sequently, a data set, composed of the derived data, is generated and transferred to RBA for further operations.

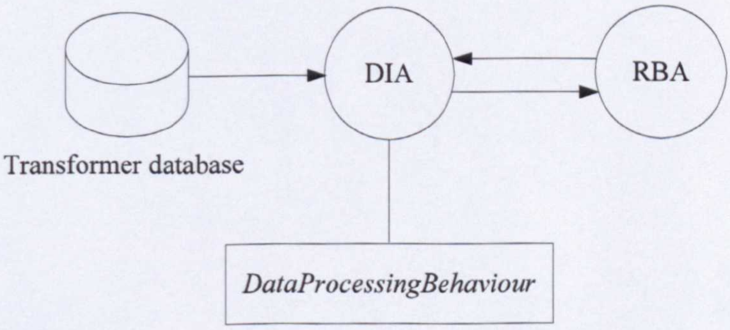


Figure 7.6: DIA behaviours

RBA

With the optimal rule selection method explained in Section 4.5, RBA is employed to select the most accurate rule for diagnosing the transformer data set received from DIA. On the other hand, new transformer fault records, collected from the transformers of a power system, are stored in the fault record database. Then, RBA periodically adds the fault records to the original

DGA training data set introduced in Section 5.2 and performs association rule generation processes with the updated training sets for generating potentially new classification rules. With a new rule set generated using a larger number of training DGA records concerning more possible transformer fault cases, a higher FD accuracy of AAMS may be achieved than using an original rule set.

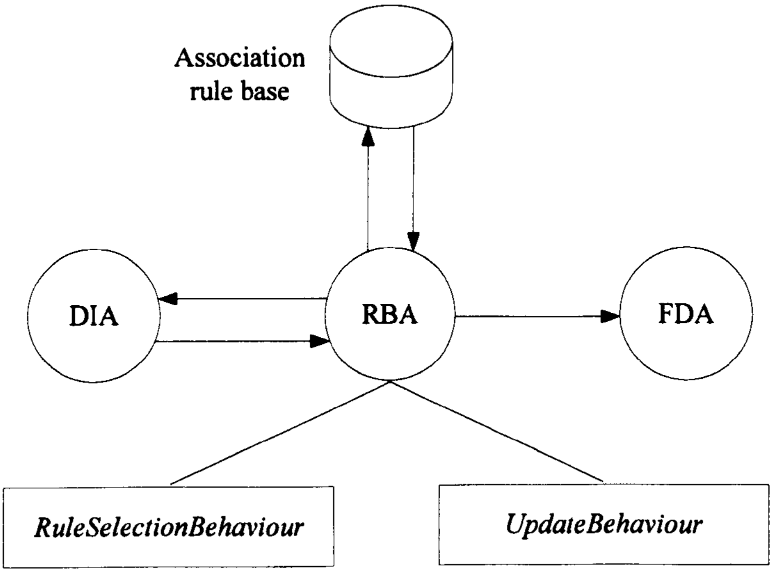


Figure 7.7: RBA behaviours

In RBA, two behaviours were defined, *i.e.*, *RuleSelectionBehaviour* and *UpdateBehaviour*, as shown in Figure 7.7:

- *RuleSelectionBehaviour*: When a transformer data set, with an AAMS processable data format, is received from DIA, the most accurate rule can be selected from the association rule base by RBA using *RuleSelectionBehaviour*. Sequently, the transformer data set and the selected rule are transferred to FDA for FD.

- *UpdateBehaviour*: As illustrated in Chapter 5, an ARS was generated from a number of training DGA records using the presented ARM-based DGA approach. Normally, the more training DGA records are employed, the more accurate association rules of transformer FD may be discovered. Consequently, a high FD accuracy may be derived. Therefore, when a certain number of new transformer DGA records, generated from the transformers of a power system, are collected, *UpdateBehaviour* will add them to the DGA training data set introduced in Section 5.2. Subsequently, an ARM process introduced in Chapter 5 is implemented using the updated training DGA records. Finally, the association rule base can be replaced with the new ARS obtained by the ARM process.

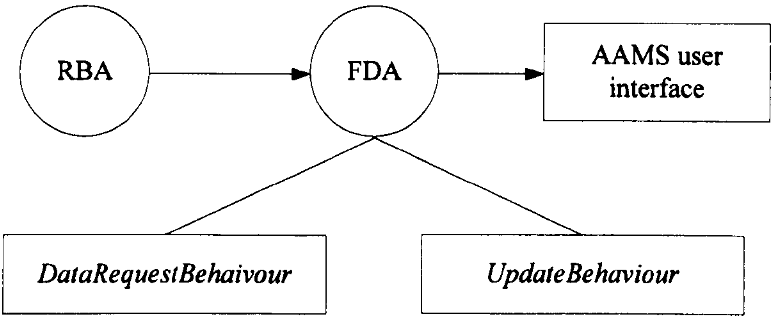


Figure 7.8: FDA behaviours

FDA

When the transformer data set and the rule selected for diagnosing the transformer data set are received from RBA, FDA is used to determine the working state of the transformer accordingly. In the case that a fault has been identified by FDA, an alarm message will be sent to the AAMS user interface for informing users. Sequently, further transformer maintenance operations will be implemented by power engineers with the received alarm message.

As illustrated in Figure 7.8, in FDA, two collaborative behaviours were developed, *i.e., RuleInterpretationBehaviour* and *FaultInfoBehaviour*:

- *RuleInterpretationBehaviour*: This behaviour is designed to diagnose the received transformer data set with the selected association rule.

- *FaultInfoBehaviour*: *FaultInfoBehaviour* is capable of sending a fault alarm message to the AAMS user interface.

7.4.3 Substation information retrieval module

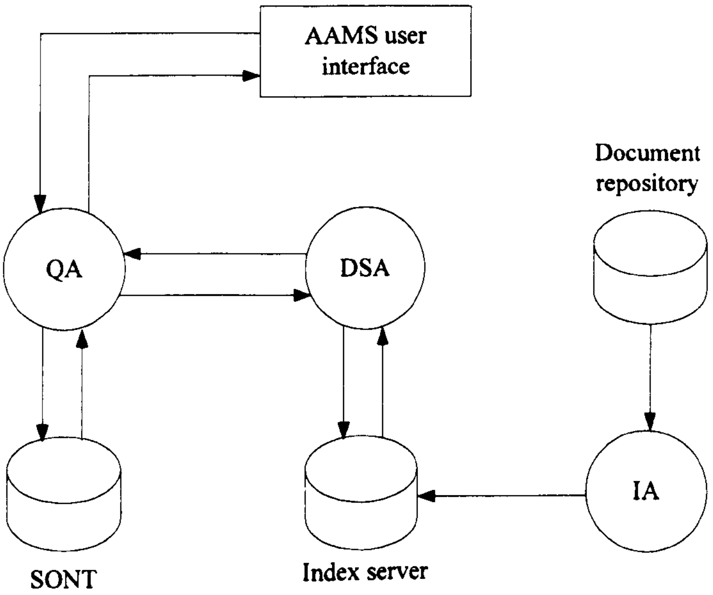


Figure 7.9: Architecture of substation IR module in AAMS

The architecture of the AAMS document retrieval module is shown as Figure 7.9. As can be seen from the figure, in total three different types of agents were developed, *i.e.*, a Query Agent (QA), a Document Search Agent (DSA) and an Index Agent (IA). Meanwhile, the Lucene-based index server and SONT, introduced in Section 3.2 and Section 2.6.1, were employed. Moreover, a document repository was used for saving the new documents of power substations, collected from a power enterprise, which can be further used for updating the index server. In the following subsections, the functions of the employed agents are explained, separately.

QA

QA is utilised to receive a query input from the AAMS user interface and then expand the query with *QueryProcessingBehaviour* based upon SONT. Sequently, the expanded query is delivered to DSA and used for document retrieval.

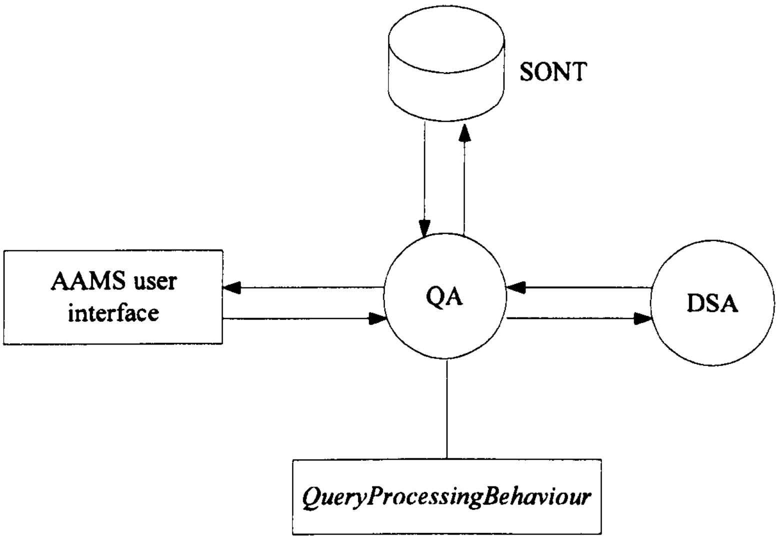


Figure 7.10: QA behaviour

IA

In AAMS, the document repository and the index server were connected via IA. Practically, IA is dedicated to generating an index of the documents stored in the document repository. Meanwhile, when a new document of power substations is collected from a power enterprise and delivered to the document repository, IA will index the document and update the index server with the generated index file accordingly. With the updated index server, more useful documents may be discovered during a document retrieval process than using the original index server.

In IA, two collaborative behaviours, *i.e.*, *IndexGenerationBehaviour* and

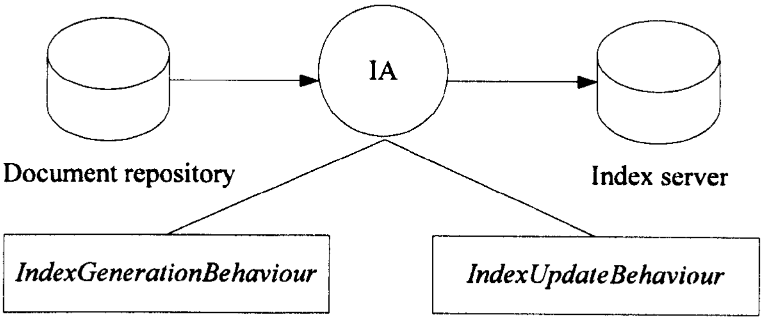


Figure 7.11: IA behaviours

IndexUpdateBehaviour were defined as that shown in Figure 7.11:

- *IndexGenerationBehaviour*: This behaviour is employed to generate an index for the index server using the documents stored in the document repository, when AAMS is initially launched. The purposes of building an index server in AAMS are to reduce search speed and improve search accuracy in a document search process, with a submitted query. Such an index generation process is explained in Section 3.2.

- *IndexUpdateBehaviour*: As mentioned above, when a new document concerning power substations is identified in the document repository, IA indexes the document with *IndexUpdateBehaviour* and then updates the index server with the derived index file.

DSA

DSA is designed to search the related document information in accordance with the expanded query received from QA. As soon as the document search process is completed, a list containing the information of the useful documents is transferred back to the AAMS user interface via QA and displayed to users for further operations.

As can be viewed from Figure 7.5, in DSA, two agent behaviours were developed, *i.e.*, *DocSearchBehaviour* and *ResultInfoBehaviour*:

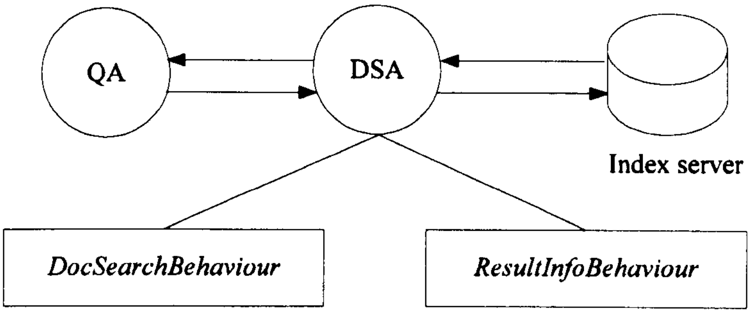


Figure 7.12: DSA behaviours

- *DocSearchBehaviour*: This behaviour employs the ER-based document ranking approach of SE_3, as introduced in Section 3.2, for identifying and ranking a set of related documents concerning an expanded query received from QA.

- *ResultInfoBehaviour*: *ResultInfoBehaviour* is used to deliver the searched document information back to the AAMS user interface, which is then used for the display of the search results.

7.5 Experiments

7.5.1 Experiment schemes

In order to illustrate the practical performance of AAMS for substation AM, a number of tests were implemented in the *e*-Automation laboratory at the University of Liverpool. For every function module of AAMS, two evaluation scenarios were employed for verifying its performance on substation AM. The detailed descriptions of these scenarios are as follows:

- **Test scheme for the power transformer FD module**: As illustrated in Section 5.7, the highest FD accuracy of an association rule-based FD system, *i.e.*, 91.53%, was achieved with the ARSs, which were generated based upon the training data set of D_{Rog} at nine different support-confidence value points, respectively. Therefore, in order to obtain a high transformer FD accuracy with AAMS, the association rule base embedded into AAMS was selected from these nine ARSs and used as the rule base of AAMS. In this study, the rule set generated at the support-confidence value point (1, 1%) was chosen by data mining experts and implemented in AAMS as the FD rule base. This was due to the fact that, compared with the other eight ARSs, this rule base contained the largest number of rules. Therefore, more kinds of transformer data sets may be correctly diagnosed by this rule base, in comparison with the others. As a result, a higher FD accuracy may be obtained when AAMS was implemented for FD of on-site power transformers.

In this part of the tests, firstly, the FD accuracy of AAMS was verified with 177 test DGA records of D_{Rog}, presented in Section 5.2. Then, the improvement of the FD accuracy with *UpdateBehaviour* of RBA was investigated. In this test, a certain number of new transformer fault records were periodically collected by RBA and added to the DGA training data set of D_{Rog} for re-generating a set of useful association rules. Once this was done, the new ARS was employed by AAMS as the FD rule base. Meanwhile, the FD accuracy with the new rule set was evaluated.

- **Test scheme for the substation IR module**: In this part of the experiments, two verification scenarios were implemented. Firstly, in order to illustrate the practical performance of AAMS on IR, the evaluation tests described in Chapter 3 were re-implemented, in which SE_3 was replaced by AAMS. Then, *IndexUpdateBehaviour* of IA was evaluated. In the experiment, a query *Fault diagnosis* was submitted to AAMS for document retrieval once the index server was updated with a number of new documents. The number of documents, discovered using an updated index server, was obtained and is illustrated in this thesis.

7.5.2 Test results and discussion

Power transformer FD

- An FD scenario with AAMS:

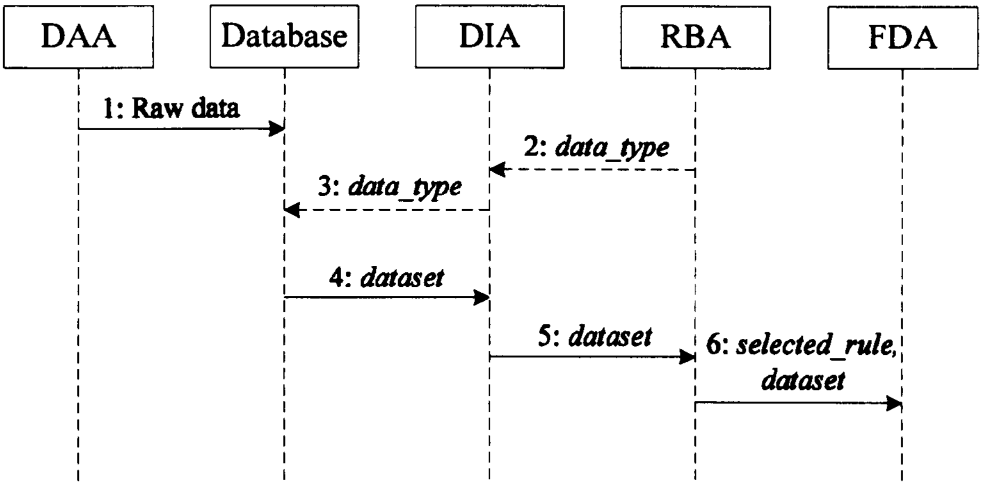


Figure 7.13: An agent coordination process in an AAMS transformer FD task

In order to obtain an in-depth understanding of employing AAMS for transformer FD, a practical FD example with AAMS is presented in this subsection. Figure 7.13 shows the trace of the messages delivered among the agents in the diagnosis process. As shown in the figure, raw data were acquired from a transformer and saved in the AAMS database by DAA. When RBA was executed, a message with content of "RBA, *data_type*", was sent to DIA which was dedicated to extracting data from the database and calculate the data values in accordance with the *dataset* requirements of RBA, *i.e.*, *data_type* = $(CH_4/H_2, C_2H_2/C_2H_4, C_2H_4/C_2H_6)$. Next, an agent message, with the content of "DIA, *dataset*" was delivered to RBA. With the received *dataset*, an optimal diagnosis rule, *i.e.*, *selected_rule*, was chosen from the association rule base. Finally, a message containing *dataset* and *selected_rule* was passed to FDA for the determination of a transformer working state. With this setup, in the case that a transformer fault is detected, an alarm message will be sent to the AAMS user interface by FDA for informing power engineers.

Taking that into account, *dataset* = (2.44, 0.93, 0.15) was sent from DIA to RBA. With the optimal rule selection method of RBA, *selected_rule* was chosen for diagnosing *dataset* from the AAMS rule base and shown as below:

> *selected_rule*: **If** $CH_4/H_2 > 1.0$; and $C_2H_2/C_2H_4 > 0.1$ and < 3.0; and $C_2H_4/C_2H_6 < 1.0$. **Then** PD.

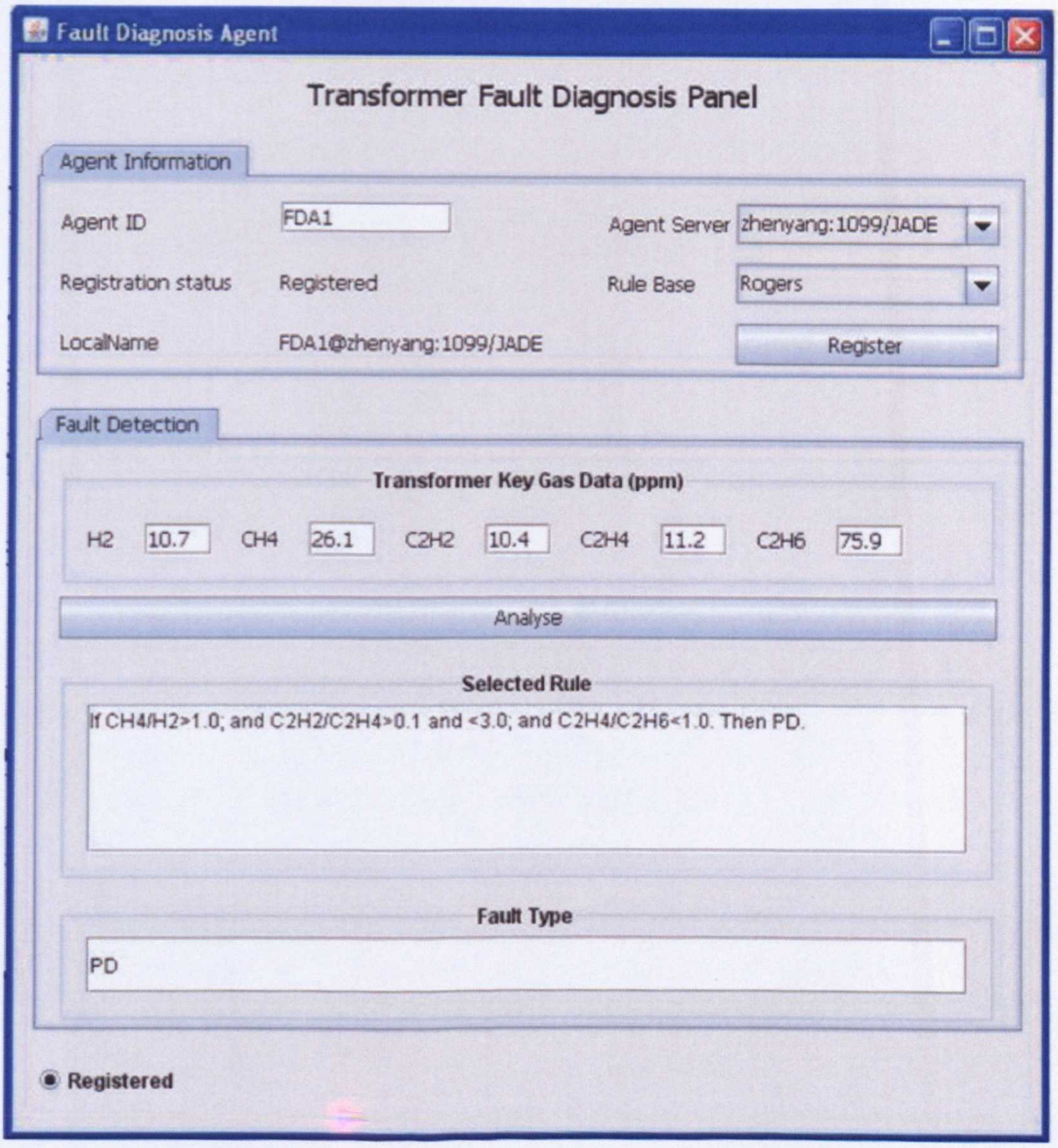
Then a message "DIA, *dataset*, *selected_rule*" was sent from RBA to FDA for a final diagnosis. The result was then derived as PD, which was the same as the final on-site diagnosis result obtained by power engineers. The alarm message sent to the AAMS user interface is shown as Figure 7.14. Finally, this fault record was recorded in the fault record database and used for further association rule generation.

- FD accuracy of AAMS:

Table 7.1: Transformer FD accuracy of AAMS using an association rule base

Rule number	Correct diagnosis	Incorrect diagnosis	Not processable	Diagnosis accuracy (%)
103	162	15	0	**91.53**

In order to verify the transformer FD accuracy of AAMS, the 177 test DGA records of D_{Rog}, explained in Section 5.2, were employed for the evaluation accordingly. The obtained results are illustrated in Table 7.1. As indicated by the table, the number of correctly diagnosed DGA records was 162. Meanwhile, in total 15 DGA records, were wrongly identified. The final diagnosis accuracy was then calculated as 91.53%, which was the same as that obtained by an association rule-based FD system using the same rule set, illustrated in Table 5.3. Therefore, a conclusion is



Transformer Fault Diagnosis Panel

Agent ID: FDA1

Agent Server: zhenyang:1099/JADE

Registration status: Registered

Rule Base: Rogers

LocalName: FDA1@zhenyang:1099/JADE

H2 10.7 CH4 26.1 C2H2 10.4 C2H4 11.2 C2H6 75.9

Selected Rule

If CH4/H2>1.0; and C2H2/C2H4>0.1 and <3.0; and C2H4/C2H6<1.0. Then PD.

Fault Type

PD

● Registered

Figure 7.14: Transformer FD panel of AAMS user interface

derived as that an association set-based FD system can be seamlessly implemented into AAMS and used for supporting transformer FD of AAMS.

- FD accuracies of AAMS with updated rule bases:

  As mentioned in Section 7.4, a rule base employed in AAMS can be

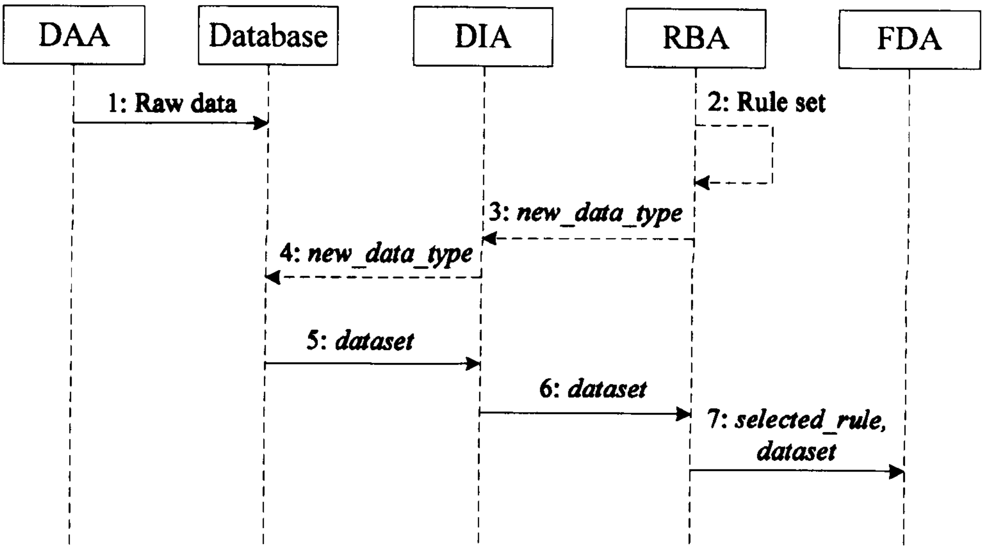


Figure 7.15: An agent coordination process in an AAMS rule base update task

periodically updated with new ARSs generated by *UpdateBehaviour* of RBA. Here, such a rule base update process is described. In tests, the FD accuracy of AAMS using an updated rule base was tested with the 177 test DGA records of D_{Rog}.

The message delivery of agent coordination for transformer FD, using an updated association rule base, is shown in Figure 7.15. When a new ARS was generated and used to replace the existing rule base of AAMS as introduced above, RBA sent a message ("RBA, *new_data_type*") to DIA. Consequently, with the received *new_data_type*, DIA reformatted the gas data of a transformer accordingly and continuously delivered collected transformer data to RBA for transformer FD.

In this test scenario, the 1016 training DGA records of D_{Rog} were provided, as explained in Section 5.2. An initialised ARS was generated with 600 DGA records, which were randomly extracted from the 1016 training DGA records. Then, an ARM process was re-implemented once 100 new DGA records were randomly collected from the training records

Table 7.2: Transformer FD accuracies of AAMS using updated association rule bases

DGA records number	600	700	800	900	1000
Rules in *Rule_set*	56	71	87	98	101
Correctly diagnosed	143	144	144	154	162
Wrongly diagnosed	16	15	15	15	15
Not processable	18	18	18	8	0
Diagnosis accuracy (%)	80.79	81.36	81.36	87.01	**91.53**

of D_{Rog} and delivered into the fault record database of AAMS. In a new ARS generated with an updated training data set, each association rule was assigned with a fitness value as well, as described in Section 4.4.3. Then, the 177 test DGA record of D_{Rog} were employed for evaluating the FD accuracy of AAMS with the updated rule base.

The obtained results are described in Table 7.2 and a corresponding FD accuracy curve derived from the test results is shown in Figure 7.16. As seen from Table 7.2, the FD accuracy increased when the rule base was replaced with a new ARS, which was generated with more training DGA records by *UpdateBehaviour* of RBA. The lowest diagnosis accuracy, *i.e.*, 80.79% was achieved with the rule set obtained using 600 DGA records. On the other hand, the largest accuracy value was derived as 91.53% using the rule set extracted from 1000 DGA training records. Therefore, it can be concluded that with *UpdateBehaviour* of RBA, the FD accuracy of AAMS may be improved with a new association rule base, generated with an increased number of transformer fault records.

IR of power substations

- A document search scenario with AAMS:

Here, a document retrieval process of AAMS is described. Figure 7.17

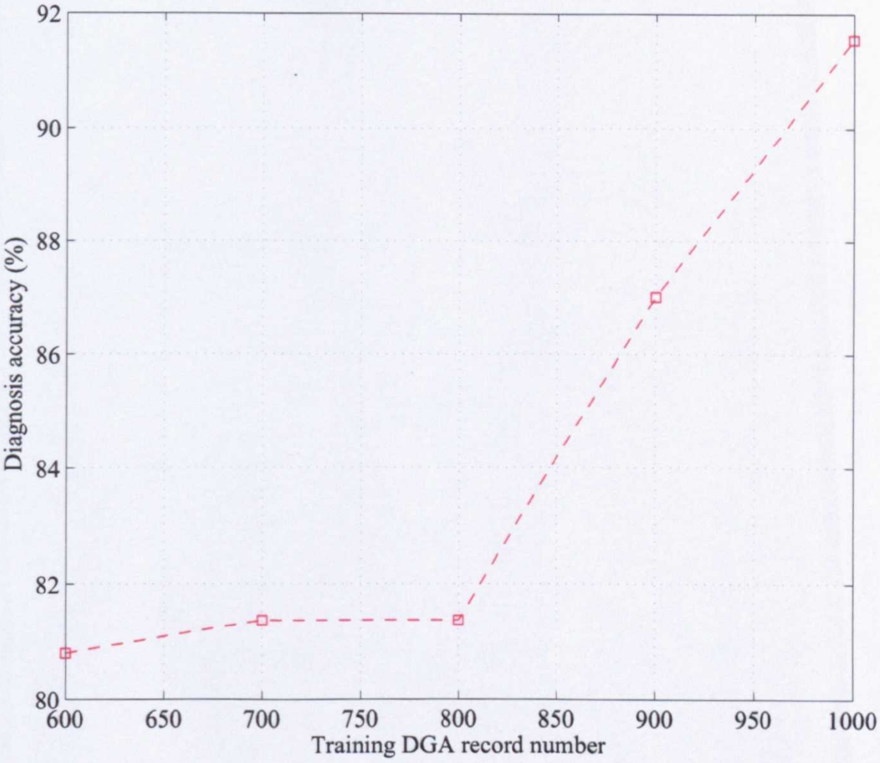


Figure 7.16: A transformer FD accuracy curve of AAMS using updated association rule bases

shows the coordination procedures of the software agents during the retrieval process. As indicated by the figure, when a query input was submitted and transferred from the AAMS user interface to QA, a QE process was carried out by QA with SONT. Afterwards, an agent message with the content of "QA, *expanded_query*" was delivered to DSA, which clearly represented the expanded query using in the document retrieval process. In the next step, a list of related documents (*doc_list*), regarding *expanded_query*, was discovered with DSA and returned to the AAMS user interface via QA. Finally, the obtained *doc_list* was displayed to users.

For illustration purposes, the query *Fault diagnosis* was provided for a

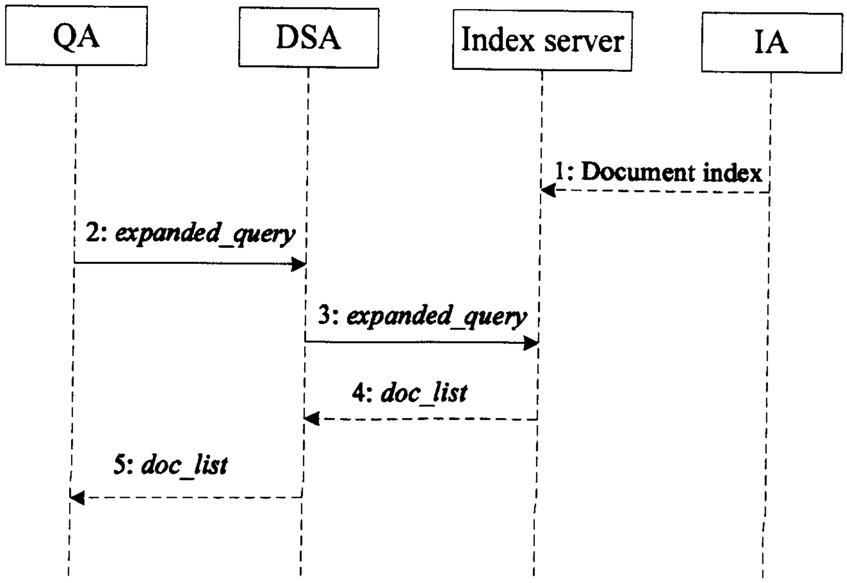


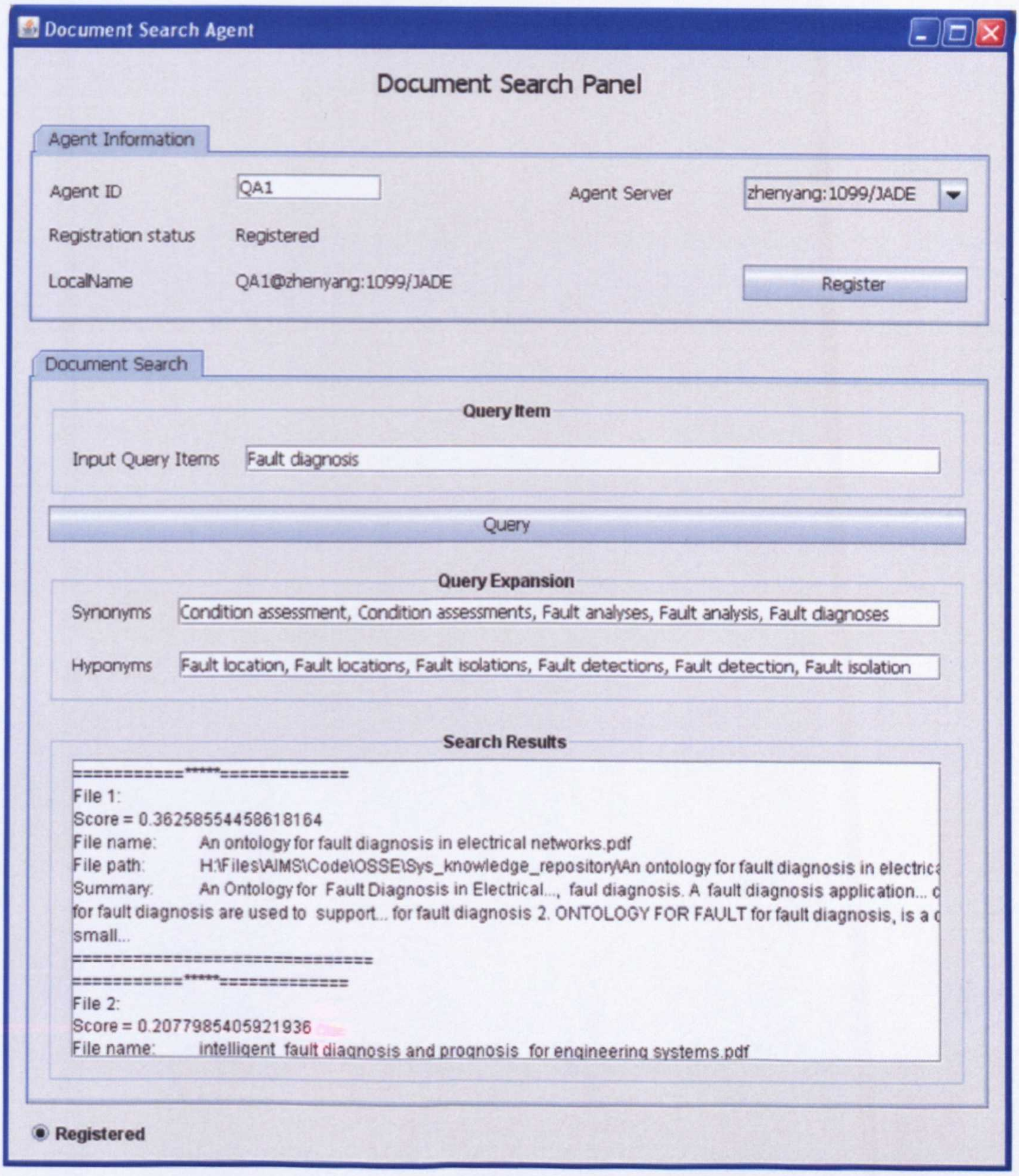
Figure 7.17: An agent coordination process in an AAMS document retrieval task

document retrieval process. With QA, a synonym set and a hyponym set of *Fault diagnosis* were derived and shown in Figure 7.18. Meanwhile, as can be see from the figure, a list of related documents of *Fault diagnosis* along with their meta-data were displayed in the "search results" box.

- Document search performance of AAMS:

  In order to evaluate the document search performance of AAMS, in terms of recall and precision, the evaluation tests introduced in Chapter 3 were re-implemented, in which SE_3 was replaced by AAMS. That is to say, three search engines in total were involved in the tests of this part, *i.e.*, SE_1, SE_2 and AAMS.

  The overall evaluation results with all the 20 queries of Table 3.3, generated by SE_1, SE_2 and AAMS, are illustrated in Figure 7.19. As indicated by the figure, the document search accuracies of AAMS, derived at 10 different recall levels, *i.e.*, from 10%, 20% to 100%, were the same as



Agent ID QA1

Agent Server zhenyang:1099/JADE

Registration status Registered

LocalName QA1@zhenyang:1099/JADE

Input Query Items Fault diagnosis

Synonyms Condition assessment, Condition assessments, Fault analyses, Fault analysis, Fault diagnoses

Hyponyms Fault location, Fault locations, Fault isolations, Fault detections, Fault detection, Fault isolation

```
===========*****=============
File 1:
Score = 0.36258554458618164
File name:      An ontology for fault diagnosis in electrical networks.pdf
File path:      H:\Files\AIMS\Code\OSSE\Sys_knowledge_repository\An ontology for fault diagnosis in electric
Summary:        An Ontology for  Fault Diagnosis in Electrical...,  faul diagnosis. A  fault diagnosis application... c
for fault diagnosis are used to  support... for fault diagnosis 2. ONTOLOGY FOR FAULT for fault diagnosis, is a c
small...
==============================
===========*****=============
File 2:
Score = 0.2077985405921936
File name:      intelligent  fault diagnosis and prognosis  for engineering systems.pdf
```

● Registered

Figure 7.18: Document search panel of AAMS user interface

that obtained by SE_3, which are illustrated in Figure 3.7. Meanwhile, the highest search accuracy of SE_3 was achieved as 80.9% at the recall level 10%, which was higher than the highest search precision of the other

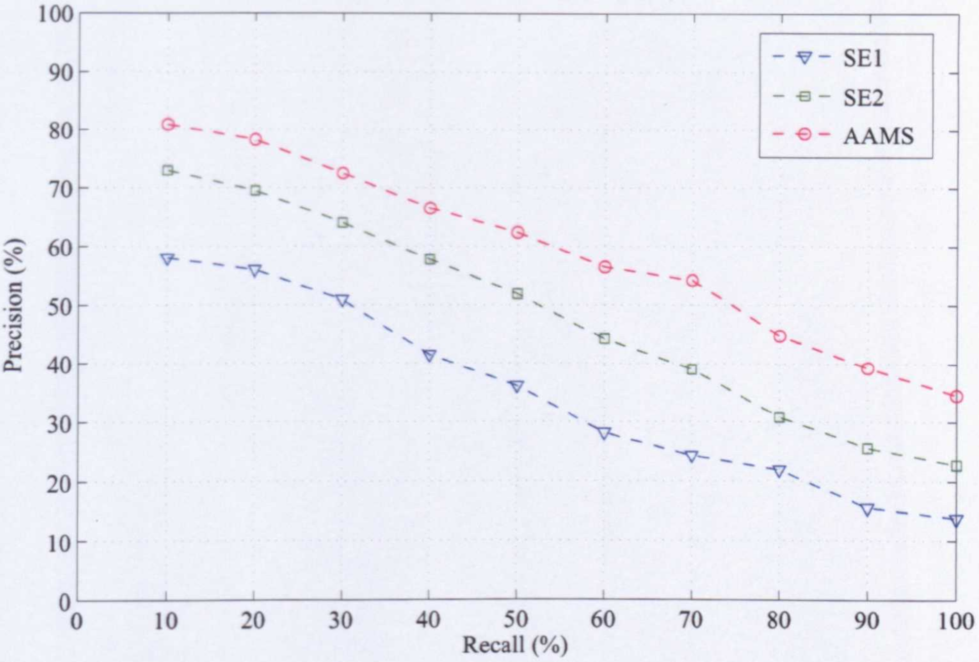


Figure 7.19: Average precision-recall curves for SE_1, SE_2 and AAMS with all 20 queries

two search engines, *i.e.*, 73.0% and 58.0%. The above results show that AAMS can be used as a suitable tool for IR of power substations, with an embedded ER-based ODSE.

- Document searching of AAMS using updated index servers:

Table 7.3: Document retrieval results of AAMS using updated index servers

Indexed documents	100	200	300	400	500
Retrieved documents	53	71	82	94	112
Useful documents	37	41	44	50	63

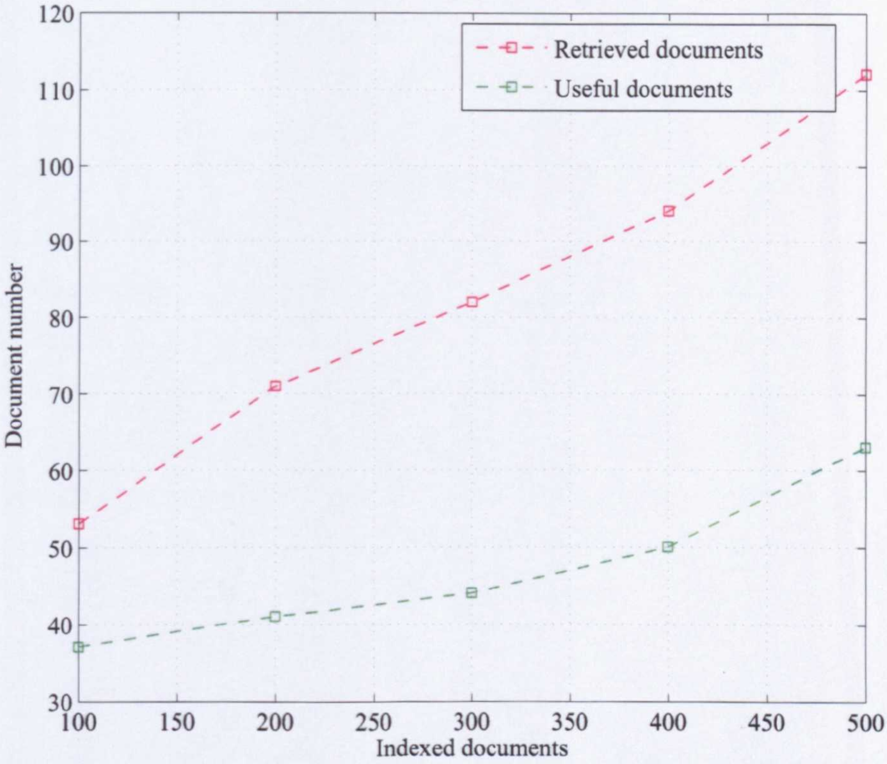


Figure 7.20: Document retrieval result curves of AAMS using updated index servers

As introduced in Section 7.4.3, the index server of AAMS can be updated by *IndexUpdateBehaviour* of IA, when a new document is discovered in the document repository of AAMS. Consequently, with an updated index server, a bigger set of useful documents may be retrieved in a document search process than using the original index server. In this test scenario, a document search process, using *Fault diagnosis*, was implemented when the index server was updated with a certain number of new documents. This was aimed towards identifying the AAMS capability for retrieving useful documents that were newly added to the index server.

A document indexing process is usually time-consuming. Therefore, in the tests, an employed index server initially contained a small number of indexed documents (100 documents) only, which were randomly selected from the document repository introduced in Section 3.3. With this index server, a document retrieval process was implemented. Then, this search process was carried out again once the index server was updated with another 100 new documents, randomly picked from the document repository of Section 3.3.

The derived results are illustrated in Table 7.3. Also, the curves of these results are obtained as Figure 7.20. As indicated by Table 7.3, in the search processes regarding *Fault diagnosis*, the number of the totally retrieved documents and the number of the discovered useful documents increased, when generated with updated index servers. The smallest sets of retrieved documents and useful documents contained 53 documents and 37 documents respectively, which were achieved with the initial index server including 100 indexed documents. Meanwhile, the largest sets of retrieved documents and useful documents were generated as 112 documents and 63 documents separately, using an updated index server holding the index information of 500 documents. Conclusively, useful documents, newly indexed by *IndexUpdateBehaviour* of IA, can be discovered during a document search process of AAMS.

7.6 Conclusion

A substation AM system namely AAMS, incorporating the ER-based ODSE and the association rule-based transformer FD system, has been presented in this chapter. Firstly, a brief introduction to the Agent technique and MAS was provided, in which the agent development platform of this study, *i.e.*, JADE, was explained. Then, the system architecture of AAMS, as well as its main functions were illustrated. Subsequently, the development of the AAMS function modules was described. Meanwhile, the functions of the software agents developed for each module of AAMS were explained in detail. Finally, a set of tests was implemented for evaluating the performance of AAMS for substation AM. Final results showed that AAMS can fulfill the requirements of substation AM for both IR and FD aspects.

Chapter 8

Conclusions

8.1 Introduction

This chapter concludes this thesis and summaries the major achievements of the presented research in the field of power substation AM. Firstly, the summary of the research results reported in the thesis is given in Section 8.2, by which the main contributions of the thesis are highlighted. Then, a number of problems and opportunities for future work are suggested in Section 8.3.

8.2 Summary

In Chapter 1, the definition of AM and the background of AM for power systems were provided firstly. Then, the brief reviews of the conventional AM techniques related to the research areas of the thesis, *i.e.*, IR of power substations and FD of power transformers, were inspected. Afterwards, the significance of employing novel intelligent techniques for tackling the drawbacks, existed in the conventional substation AM solutions, was explained. Moreover, the substation AM approaches reported by this thesis, *i.e.*, the ER-based document ranking approach to an ODSE and the ARM-based approach to transformer DGA, were listed and briefly introduced. Finally, the thesis outline and the major contributions derived from the thesis were indicated, followed by a list of academic papers that have been published or submitted by the thesis author.

In order to overcome the drawbacks of a traditional ODSE, the ER-based document ranking approach has been presented in Chapter 2. In the chapter, the historical literature review of IR was provided firstly. Meanwhile, a number of published document ranking techniques of IR, such as VSM and so on, were presented. Subsequently, the basic introduction to an ODSE, the Ontology technique, ER and DS, which were the development foundations of the proposed ER-based approach, was given. The development process of SONT, employed for QE in an ODSE, was also illustrated. The methodology used for organising the terms of an expanded query into a MADM tree model then was discussed. Subsequently, the ER algorithm, developed based upon DS and used for generating the relatedness between the expanded query and a specific document, was explained in detail.

The experimental work of the proposed ER-based document ranking approach has been reported in Chapter 3. For comparison purposes, in total three different search engines, involving the traditional keyword-matching search engine without QE and ER, and the two ODSEs with and without ER, were developed and tested based upon the same test scheme, respectively. The obtained results clearly showed that, in an ODSE, the ER algorithm has provided a suitable solution for combining multiple relevance scores generated between the terms of an expanded query and a document. More significantly, the ER-based ODSE has achieved the highest search accuracy in all the three search engines, which indicated a search accuracy improvement of an ODSE by applying the ER-based document ranking approach. Therefore, the ER-based approach can be employed as a viable solution for ranking document in an ODSE.

In Chapter 4, the ARM-based DGA approach to FD of power transformers has been presented. In this chapter, the literature reviews of DGA, including several traditional DGA methods and a set of widely used AI-based DGA classifiers, were given firstly as a knowledge basis for understanding the development of the ARM-based DGA approach. Subsequently, the ARM technique was briefly introduced. Several techniques, used in an ARM process for generating useful association rules of DGA, then were explained. *In the ARM* process, firstly, the attribute selection method and the continuous datum attribute discretisation method were presented for choosing the user-interested attributes of ARM from a provided DGA data set. Apriori-TFP, utilised for generating a raw ARS with a set of training DGA records, was discussed afterwards. In order to select useful fault classification rules from the obtained rule set, the rule set simplification and rule fitness evaluation methods were introduced and employed, respectively. Subsequently, a transformer FD system was developed based upon the extracted useful association rules. Further in the section, the optimal rule selection method, utilised for selecting the most accurate rule from the developed FD system for a specific diagnosis case, was presented.

The tests for evaluating the practical performance of the proposed ARM-based DGA approach has been expressed in Chapter 5. Totally six different transformer FD methods were comparatively implemented in the tests using the same training and test data sets, which included the ARM-based DGA method, the Dornenburg and Rogers ratio methods, the ANN, SVM and KNN classifiers. The generated results showed that the proposed ARM-based DGA approach was capable of generating a number of meaningful association rules which could cover the empirical rules of a conventional DGA method, *e.g.*, Dornenburg or Rogers. More significantly, an improved FD accuracy has been achieved using the ARM-based DGA approach, compared with the other methods. Thus, the proposed ARM-based DGA approach can be proposed as a feasible solution for FD of power transformers.

The development and evaluation processes of a SWRL rule base, which was derived from a set of useful association rules generated in Chapter 5, have been illustrated in Chapter 6. In the chapter, the brief introduction to SWRL was provided firstly. Meanwhile, the several advantages of using a SWRL rule base in a RBES, which are not achievable by employing an ARS, were discussed. Then, Jess was presented for constructing a SRBES with a SWRL rule base. Such a SRBES development process, using SWRLTab, was clearly demonstrated afterwards. Meanwhile, the functions of the SRBES modules were introduced in detail. In the tests, the same FD accuracy has been achieved by a SRBES, compared with that obtained using an association rule-based FD system. Thus, the capability of a SRBES for transformer FD has been verified.

With the ER-based ODSE constructed in Chapter 3 and the association-based FD system generated in Chapter 5, AAMS has been developed and employed for substation AM in Chapter 7. In the section, the literature review of the Agent technique and MAS was provided firstly. Subsequently, the system structure of AAMS and the AM services of AAMS were described. Then, the specific functions provided by the software agents of AAMS were briefly explained. Meanwhile, the agent coordination and cooperation processes for substation AM were illustrated. At the end of this chapter, the tests used for evaluating the performance of AAMS for substation AM were shown. Conclusively, the proposed AAMS with the two proposed AM approaches can be used as a feasible solution for IR and FD of power substations.

8.3 Suggestions for future work

As illustrated in this thesis, satisfactory results have been achieved by the presented substation AM approaches. However, several limitations still exist in these approaches and corresponding improvements need to be addressed in the future.

- For the ER-based document ranking approach to an ODSE: Firstly, SONT was developed manually. As a result, the related terms of an original query could be accurately discovered with SONT during a QE process. However, an ontology model building process is a time-consuming task, and the knowledge of the ontology model requires a continuous update in order to refine the context of the ontology model. Apparently, it is an unsuitable way to manually develop a more general ontology model, or a domain ontology model regarding a complex domain. In the future, automatic ontology constructing approaches, presented in [142], [143] and [144], may be employed to extend the context of SONT from a power *substation domain* to a power system domain.

Secondly, as mentioned in Section 3.6, the time consumption performance of the three search engines was obtained in the tests. However, as discussed in that section, the ER-based ODSE did not achieve a distinct advantage in most cases, which represents a limitation of the proposed methodology. Thus, future work will be addressed on it, in order to reduce the average time consumed for a search process with the ER-based ODSE.

- For the ARM-based DGA approach: Firstly, the tests were implemented based upon the provided training and test DGA records. Although a satisfactory diagnosis accuracy has been achieved in the study, an improved FD accuracy may be derived, if a larger DGA training data set can be used for ARM. Currently, more DGA records are being collected and the tests of this study will be re-implemented with an updated training data set, in order to further verify the capability of the proposed ARM-based DGA approach to FD of power transformers.

  Secondly, Jess was used to develop a SRBES with a generated SWRL rule base. In practice, some other rule engines are also available to infer SWRL rules, *e.g.,* Hoolet, Algernon and SweetRules and so on, as mentioned in Section 6.2. Therefore, in the next step, with each of these rule engines, the tests of Section 6.6 will be implemented again for illustrating the possibility of reusing the SWRL rule base among different RBESs. FD accuracies obtained using these rule engines will be compared with that generated with Jess as well.

- For AAMS: As explained in Section 6.2, compared with an ARS, a SWRL rule base offers several advantages in a RBES. Therefore, $SRBES_{\mathrm{Rog}}$, constructed in Chapter 6, will be employed to replace the association rule-based FD system, which has been embedded in AAMS.

Secondly, in this study, all the tests of AAMS were implemented simulatively. In order to verify the AM capability of AAMS for on-site power substations, an implementation strategy will be developed, with which AAMS can be applied to IR and FD of on-site power substations in NG. In such an implementation process, the system stability of AAMS and the network security of NG Local Area Networks (LANs) will be treated as the most vital elements and thus corresponding configurations will be schemed.

Appendix A

Public Softwares Used in This Research

Table A.1: Public softwares

Software 1	**Apriori-TFP**
Function	Apriori-TFP is an ARM algorithm used for generating association rules.
Web link	http://www.csc.liv.ac.uk/~frans/KDD/Software/Apriori-TFP/aprioriTFP.html
Software 2	**JADE**
Function	JADE is an agent development environment, which is one of the implementations of the FIPA agent specification and established based upon Java programming language.
Web link	http://jade.tilab.com/
Software 3	**Java Development Kit (JDK)**

Continued on next page...

Table A.1 – continued from previous page

Function	JDK is composed of the Java standard edition Runtime Environment (JRE) and command-line development tools that are used for developing Java programs.
Web link	http://java.sun.com/javase/downloads/index.jsp
Software 4	**Jena**
Function	Jena is a Java framework for building semantic web applications.
Web link	http://jena.sourceforge.net
Software 5	**Jess**
Function	Jess is both a rule execution engine and a rule programming environment fully written in Java.
Web link	http://herzberg.ca.sandia.gov/
Software 6	**Lucene**
Function	Lucene is a high-performance, full-featured text search engine library written entirely in Java.
Web link	http://lucene.apache.org/java/docs/
Software 7	**Protege**
Function	Protege is a free, open source ontology editor and knowledge-base framework.
Web link	http://protege.stanford.edu/
Software 8	**SWRLTab**
Function	SWRLTab is a development environment for working with SWRL rules in Protege-OWL.

Continued on next page...

Table A.1 – continued from previous page

Web link	http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab

Bibliography

[1] Transformers Committee of the IEEE Power Engineering Society. IEEE guide for the interpretation of gases generated in oil immersed transformers, IEEE standard c57.104-1991. Technical report, The institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017, U.S., 1994.

[2] Hydroelectric Research and Technical Service Group. *Transformer maintenance.* United States Department of the Interior, Bureau of Reclamation, Denver, Colorado, U.S., October 2002.

[3] National Asset Management Steering Group Australia. *International infrastructure management manual.* Association of Local Government Engineering New Zealand, Thames, New Zealand, 2002.

[4] M. Beardow. Economics of asset management: drawing it together. Technical report, ESAA 2003 Residential School in Electrical Power Engineering, Brisbane, Australia, February 2003.

[5] J. Crisp and D. Birtwhistle. System dynamics modelling: application to electricity transmission network asset management. *Australian J. of Electrical and Electronics Engineering*, 2(3):263–271, 2005.

[6] L. Bertling, R. Allan, and R. Eriksson. A reliability-centered asset maintenance method for assessing the impact of maintenance in power distribution systems. *IEEE Trans. Power Systems*, 20(1):75–82, February 2005.

[7] R. Merritt. Asset management keeps plants running smarter. *Control (Chicago, III)*, 13(3):6 pages, 2000.

[8] M.D. Judd, S. McArthur, J.R. McDonald, and O. Farish. Intelligent condition monitoring and asset management: partial discharge monitoring for power transformers. *Power Engineering J.*, 16(6):297–304, December 2002.

[9] D. Buse and Q. H. Wu. *IP Network-based Multi-agent Systems for Industrial Automation.* Springer London, 2007.

[10] J.B. Yang and M.G. Singh. An evidential reasoning approach for multi-attribute decision making with uncertainty. *IEEE Trans. System, Man and Cybernetics*, 24(1):1–18, Jan 1994.

[11] J.B. Yang and P. Sen. A general multi-level evaluation process for hybrid MADM with uncertainty. *IEEE Trans. System, Man and Cybernetics*, 24(10):1458–1473, 1994.

[12] R. Agrawal, T. Imielinkski, and T. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 207–216, May 1993.

[13] R.B. Yates and B. R. Neto. *Modern information retrieval.* Addison Wesley, first edition, May 1999.

[14] K. Eguchi, H. Ito, A. Kumamoto, and Y. Kanata. Adaptive and incremental query expansion for cluster-based browsing. In *Proc. 6th Int. Conf. on Database Systems for Advanced Application*, pages 25–34, April 1999.

[15] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *J. American Society for Information Science*, 41(4):288–297, Jan 1990.

[16] B.R. Schatz, E.H. Johnson, P.A. Cochrane, and H. Chen. Interactive term suggestion for users of digital libraries: using subject thesauri and co-occurrence lists for information retrieval. In *Proc. DL-96, 1st ACM Digital Library Conf.*, pages 126–133, Bethesda, U.S., 1996.

[17] W.W. Chu, Z. Liu, and W. Mao. Textual document indexing and retrieval via knowledge sources and data mining. *Communication of the Institute of Information and Computing Machinery(CIICM)*, 5(2), 2002.

[18] M.J. Bates. After the dot-bomb: getting information retrieval right this time. *First Monday*, 7(7), 2002.

[19] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. WordNet: An on-line lexical database. *Int. J. Lexicography*, 3(4):235–312, 1990.

[20] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information Processing and Management*, 43(4):866–886, July 2007.

[21] E. Voorhees. Query expansion using lexical-semantic relations. In: W. Bruce Croft and C.J. van Rijsbergen edition. In *Proc. 17th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 61–69, Dublin, Ireland, 1994. Springer-Verlad, New York, Inc.

[22] G. Fu, C.B. Jones, and A.I. Abdelmoty. Ontology-based spatial query expansion in information retrieval. In *Int. Conf. on Ontologies, Databases and Application of Semantics*, Lecture Notes in Computer Science 3761, pages 1466–1482, Agia Napa, Cyprus, October-November 2005. Springer-Verlag.

[23] P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Trans. Knowledge and Data Engineering*, 19(2):261–272, February 2007.

[24] D. Bonino, F. Como, L. Farinetti, and A. Bosca. Ontology driven semantic search. *WSEAS Trans. Information Science and Application*, 1(6):1597–1605, December 2004.

[25] C. Lin, J. Ling, and C. Huang. An expert system for transformer fault diagnosis using dissovled gas analysis. *IEEE Trans. Power Delivery*, 8(1):231–238, January 1993.

[26] K. Tomsovie, M. Tapper, and T. Ingvarsson. A fuzzy information approach to integrating different transformer diagnostic methods. *IEEE Trans. Power Delivery*, 8(3):1638–1646, July 1993.

[27] H. Yang and C. Liao. Adaptive fuzzy diagnosis system for dissolved gas analysis of power transformers. *IEEE Trans. Power Delivery*, 14(4):1342–1350, 1999.

[28] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Mathematical Biophysics*, 5(115-133):18–27, 1943.

[29] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley and Sons, Inc., New York, U.S., second edition, 2001.

[30] B.E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proc. 5th Annual ACM Workshop on COLT*, pages 144–152, Pittsburgh, U.S., 1992. ACM Press.

[31] V.N. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, Inc., New York, U.S., 1998.

[32] L.S. Davis and A. Rosenfeld. Noise cleaning by iterated cleaning. *IEEE Trans. System, Man and Cybernetics*, 8:705–710, 1978.

[33] S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Academic Press, London, U.K., second edition, 2003.

[34] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communication of the ACM*, 18(11):613–620, 1975.

[35] C.D. Manning, P. Raghavan, and H. Schutze. *Term weighting and vector space models*, pages Chapter 6: 93–116. C.D. Manning and P. Raghavan and H. Schutze, preliminary draft edition, August 2007.

[36] J.L. Guardado, J.L. Naredo, P. Moreno, and C.R. Fuerte. A comparative study of neural network efficiency in power transformers diagnostic using dissolved gas analysis. *IEEE Trans. Power Delivery*, 16(4):643–647, 1999.

[37] M.J. Heathcote. *The J and P Transformer Book: A Practical Technology of the Power Transformer.* Newnes, Oxford, twelfth edition, 1998.

[38] G. Shafer. *A mathematical theory of evidence.* Princeton University Press, Princeton, NJ, 1976.

[39] V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, July 1945.

[40] C.N. Mooers. Application of random codes to the gathering of statistical information. Thesis (m.s.), Massachusetts Institute of Technology, Dept. of Mathematics, Massachusetts, U.S., 1948.

[41] H.P. Luhn. A statistical approach to mechanised encoding and searching of literary information. *IBM J. Research and Development*, 1(4):309–317, 1957.

[42] C.W. Cleverdon. The Cranfield tests on index language devices. *Aslib Proc.*, 19(6):173–192, 1967.

[43] Editor G. Salton. *The SMART retrieval system: experiments in automatic document processing.* Englewood Cliffs, NJ: Prentice-Hall, G. Salton edition, 1971.

[44] D.K. Harman. The first text retrieval conf. (TREC-1), Rockville, MD, U.S., 4-6 November, 1992 . *Information Processing and Management*, 29(4):411–414, 1993.

[45] *Google search engine.* Available at http://www.google.com/.

[46] *Yahoo search engine.* Available at http://www.yahoo.com/.

[47] *Microsoft MSN search engine.* Available at http://www.search.msn.com/.

[48] M.E. Maron and J.L. Kuhns. On relevance, probabilistic indexing, and information retrieval. *J. ACM*, 7(3):216–244, July 1960.

[49] S. Robertson and K. Jones. Relevance weighting of search terms. *J. ASIS*, 27(3):129–146, 1976.

[50] S. Robertson, S. Walker, M. Beaulieu, A. Gull, and M. Lau. Okapi at TREC-1. In *Proc. TREC-1 Notebook*, pages 21–30, Gaithersburg, MD, U.S., 1992.

[51] *H.R. Turtle.* Ph.d. thesis, University of Massachusetts, Amherst, MA, U.S., 1990.

[52] P. Lee. *Bayesian statistics: an introduction.* Hodder Arnold, Third edition, March 2004.

[53] E.A. Fox. *Extending the Boolean and vector space models of information retrieval with P-norm queries and multiple concept types.* University microfilms, Cornell University, Ann Arbor, MI, U.S., 1983.

[54] T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[55] T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Human-Computer Studies*, 43:907–928, November 1995.

[56] N. Guarino. Formal ontology and information systems. In *Proc. 1st Int. Conf. on Formal Ontologies in Information Systems*, pages 3–15, Trento, Italy, June 1998. IOS Press, Amsterdam.

[57] Cycorp. *Cyc 101 Tutorial.* Available at http://www.opencyc.org/doc/tut/?expand_all=1.

[58] L. Chen, N.R. Shadbolt, and C.A. Goble. A semantic web-based approach to knowledge management for grid applications. *IEEE Trans. Knowledge and Data Engineering*, 19(2):283–296, February 2007.

[59] M. Missikoff, R. Navigli, and P. Velardi. Integrated approach to web ontology learning and engineering. *IEEE Computer*, 35(11):60–63, November 2002.

[60] R. Navigli, P. Velardi, and A. Gangemi. Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems*, 18(1):22–31, January-February 2003.

[61] W3C. *Extensible Markup Language (XML).* Available at http://www.w3.org/XML/.

[62] W3C. *Extensible Markup Language (XML) Schema.* Available at http://www.w3.org/XML/Schema.

[63] W3C. *Resource Description Framework (RDF).* Available at http://www.w3.org/RDF/.

[64] W3C. *Resource Description Framework (RDF) Schema Specification 1.0.* Available at http://www.w3.org/TR/2000/CR-rdf-schema-20000327/.

[65] W3C. *Annotated DAML+OIL Ontology Markup.* Available at http://www.w3.org/TR/2001/NOTE-daml+oil-walkthru-20011218/.

[66] W3C. *OWL: Web Ontology Language.* Available at http://www.w3.org/TR/owl-ref/.

[67] M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe. *A practical guide to building owl ontologies using the Protege-owl plugin and co-ode tools edition 1.0.* The University of Manchester, Manchester, U.K., August 2004.

[68] G. Antoniou and F. Harmelen. *A semantic web primer.* MIT Press, Cambridge, Massachusetts, London, England, April 2004.

[69] W. Hodges. *Classical logic 1: first order logic*, pages 9–32. The Blackwell Guide to Philosophical Logic. Blackwell, lou goble edition, 2001.

[70] M. Ushold, M. Kind, S. Moralee, and Y. Zorgios. The enterprise ontology. *The Knowledge Engineering Review*, 13:31–89, 1998.

[71] P. Sen and J.B. Yang. Multiple criteria decision making in design selection and synthesis. *J. Engineering Design*, 6(3):207–230, 1995.

[72] Y.M. Wang, J.B. Yang, and D.L. Xu. Environmental impact assessment using the evidential reasoning approach. *European J. Operational Research*, 174(3):1885–1913, November 2006.

[73] W.H. Tang, K. Spurgeon, Q.H. Wu, and Z.J. Richardson. An evidential reasoning approach to transformer condition assessments. *IEEE Trans. Power Delivery*, 19(4):1696–1703, Oct 2004.

[74] J.B. Yang, B.G. Dale, and C.H.R. Siow. Self-assessment of excellence: an application of the evidential reasoning approach. *Int. J. Production Research*, 39(16):3789–3812, November 2001.

[75] J. Wang. A subjective methodology for safety analysis of safety requirements specifications. *IEEE Trans. Fuzzy System*, 5(3):1–13, June 1997.

[76] A.P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Statistics*, 38:325–339, 1967.

[77] A. Andreou. Ontologies and query expansion. Master's thesis, School of Informatics, University of Edinburgh, Edinburgh, U.K., 2005.

[78] Ch.G. Wu, W.P. Jiao, and Q.J. Tian. An information retrieval server based on ontology and multiagent. *J. Computer Research and Development*, 38(6):641–647, 2001.

[79] B. McBride. *Jena: implementing the RDF model and syntax specification.* Hewlett Packard Laboratories, Available at http://www.hpl.hp.com/personal/bwm/papers/20001221-paper/, 2001.

[80] K. Jarvelin, J. Kekalainen, and T. Niemi. ExpansionTool: concept-based query expansion and construction. *Information Retrieval*, 4(3-4):231–255, September 2001.

[81] O. Gospodnetic and E. Hatcher. *Lucene in action.* In Action. Manning Publications, Manning Publications Co., Cherokee Station, PO Box 20386, New York, NY 10021, December 2004.

[82] T.L. Saaty and L.G. Vargas. *Models, methods, concepts and applications of the analytic hierarchy process.* Kluwer Academic Publisher, Dordrecht the Netherlands, 2001.

[83] D.R. Anderson, D.J. Sweeney, and T.A. Williams. *An introduction to management science: Quantitative approaches to decision making.* South Western College Publishing, Cincinnati Ohio, 2000.

[84] J. Karlsson, C. Wohlin, and B. Regnell. An evaluation of methods for prioritising software requirements. *Information and Software Technology*, 39(14-15):938–947, 1998.

[85] M. Svahnberg. An industrial study on building consensus around software architectures and quality attributes. *J. Information and Software Technology*, 46(12):805–818, 2004.

[86] Apache, Available at http://lucene.apache.org/java/docs/scoring.html. *Apache Lucene Scoring*, 2008.

[87] T. Coelho, P. Calado, L. Souza, B. Ribeiro-Neto, and R. Muntz. Image retrieval using multiple evidence ranking. *IEEE Trans. Knowledge and Data Engineering*, 16(4):408–417, April 2004.

[88] F. Coenen. *The LUCS-KDD TFP association rule mining algorithm.* LUCS-KDD Research Team, Department of Computer Science, The University of Liverpool, U.K., 2004.

[89] A. Shintemirov, W. H. Tang, and Q. H. Wu. Power transformer fault classification based on dissolved gas analysis by implementing bootstrap and genetic programming. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 2008. in print.

[90] L. Zadeh. Fuzzy sets. *J. Information and Control*, 8:338–353, 1965.

[91] Ch. Hsu and Ch. Lin. A comparison of methods for multiclass support vector machines. *IEEE Trans. on Neural Networks*, 13(2):415–425, March 2002.

[92] L. Zhang, L.B. Jack, and A.K. Nandi. Fault detection using genetic programming. *Mechanical Systems and Signal Processing*, (19):271–289, 2005.

[93] O.R. Zaiane. Introduction to data mining. In *Principle of Knowledge Discovery in Databases*, Lecture Notes, chapter 1. 1999.

[94] S. Konias and N. Maglaveras. A rule discovery algorithm appropriate for electrocardiograph signals. In *Proc. the 31st Computers in Cardiology*, pages 57–60, Chicago, 2004.

[95] A. Kocatas, A. Gursoy, and R. Atalay. Application of data mining techniques to protein-protein interaction prediction. In *Computer and Information Science - ISCIS 2003*, volume 2869/2003 of *Lecture Notes in Computer Science*, pages 316–323. Springer Berlin / Heidelberg, October 2003.

[96] D. Liu, Y. Chen, Y. Fan, and G. Shen. The application of association rule mining in power system restoration. In *Proc. IEEE Power Engineering Society General Meeting*, pages 5–9, Montreal, Canada, June 2006. IEEE.

[97] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proc. 11th Int. Conf. on Machine Learning*, pages 121–129, New Brunswick, Morgan Kaufmann, 1994.

[98] R. Jin, Y. Breitbart, and C. Muoh. Data discretisation unification. In *Proc. IEEE Int. Conf. on Data Mining*, Omaha NE, U.S., March 2007. IEEE.

[99] H. Liu, F. Hussain, C.T. Lan, and M. Dash. Discretisation: an enabling technique. *Data Mining and Knowledge Discovery*, 6(4):393–423, October 2002.

[100] M.J. Pazzani. An iterative improvement approach for the discretisation of numeric attributes in bayesian classifiers. In *Proc. First Int. Conf. on Knowledge Discovery and Data Mining*, pages 228–233, Montreal, Quebec, Canada, August 1995.

[101] G. Goulbourne, F.P. Coenen, and P. Leng. Algorithms for computing association rules using a partial-support tree. In *Proc. ES99 Conf.*, pages 132–147, London, 1999. Springer.

[102] G. Goulbourne, F.P. Coenen, and P. Leng. Algorithms for computing association rules using a partial-support tree. *J. of Knowledge-based Systems*, 13:141–149, 2000.

[103] F. Coenen. *The LUCS-KDD Apriori-T association rule mining algorithm.* LUCS-KDD Research Team, Department of Computer Science, The University of Liverpool, U.K., 2004.

[104] R.J. Bayardo. Brute-force mining of high-confidence classification rules. In *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining*, pages 123–126. AAAI Press, 1997.

[105] A.A. Freitas. A survey of evolutionary algorithms for data mining and knowledge discovery. In *A. Ghosh and S. Tsutsui. (Eds.) Advances*

in Evolutionary Computation, pages 819–845. Springer-Verlag, August 2002.

[106] J. Dean and R. Dean. *Introduction to programming with Java: a problem solving approach.* McGraw-Hill Higher Education, January 2008.

[107] E.F. Hill. *Jess: Java expert system shell.* Sandia National Laboratories, Livermore, Canada, Available at http://www.jessrules.com/jess/index.shtml.

[108] *Hoolet.* Available at http://owl.man.ac.uk/hoolet/.

[109] B. Kuipers, M. Hewett, S. Bishop, and J. Crawford. *Algernon.* Available at http://www.cs.utexas.edu/users/qr/algy/.

[110] *SweetRules.* Available at http://sweetrules.projects.semwebcentral.org/.

[111] Rule Markup Initiative, Available at http://www.ruleml.org/. *Rule Markup Language (RuleML).*

[112] W3C. *Metalog - towards the semantic web.* Available at http://www.w3.org/RDF/Metalog/.

[113] P. Deransart, A. Ed-Dbali, and L. Cervoni. *Prolog: the standard.* Springer, 1996.

[114] W3C. *SWRL: A semantic web rule language combining OWL and RuleML.* Available at http://www.w3.org/Submission/SWRL/.

[115] W. Zhao and J.K. Liu. OWL/SWRL representation methodology for EXPRESS-driven product information model. *Computers in Industry*, 59(6):590–600, August 2008.

[116] Y. Hu, Z. Wu, and M. Guo. Ontology driven adaptive data processing in wireless sensor networks. In *Proc. 2nd Int. Conf. on Scalable Information Systems*, volume 304, SuZhou, China, June 2007.

[117] R.D. Shankar, S.B. Martins, M. O'Connor, D.B. Parrish, and A.K. Das. Epoch: an ontological framework to support clinical trials management. In *Proc. Int. Workshop on Healthcare information and knowledge management*, pages 25–32, Arlington, Virginia, U.S., 2006.

[118] J. Giarratano and G. Riley. *Expert system: principles and programming*. Brooks/Cole Publishing Co., Pacific Grove, CA, U.S., 1989.

[119] A. Abraham. *Rule-based expert systems, handbook for measurement systems design*, pages 909–919. John Wiley and Sons Ltd., London, England, Peter Sydenham and Richard Thorn edition, 2005.

[120] C. Forgy. Rete: a fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1):17–37, September 1982.

[121] M. Kuan. Using SWRL and OWL DL to develop an inference system for course scheduling. Master's thesis, Chung Yuan Christian University, Taiwan, 2004.

[122] J. Mei and E.P. Bontas. Reasoning paradigms for SWRL-enabled ontologies. In *Proc. 8th Int. Protege Conf.*, Madrid, Spain, July 2005.

[123] C. Golbreich and A. Imai. Combining SWRL rules and OWL ontologies with Protege OWL plugin, Jess and Racer. In *Proc. 7th Int. Protege Conf.*, Bethesda, Maryland, July 2004.

[124] J. Giarratano. *CLIPS user's guide*. CLIPS Expert System Group, Available at http://clipsrules.sourceforge.net/documentation/v630/ug.htm, December 2007.

[125] J. McCarthy. LISP: a programming system for symbolic manipulations. In *Proc. 14th national meeting of the association for computing machinery*, pages 1–4, Cambridge, Massachusetts, 1959. ACM, New York, U.S.

[126] M. O'Connor, S. Tu, A. Das, and M. Musen. Querying the semantic web with SWRL. In *Proc. Int. RuleML Synposium on Rule Interchange and Applications (RuleML2007)*, pages 155–159, Orlando, 2007. Springer Verlag.

[127] C. Ma, W.H. Tang, Z. Yang, Q.H. Wu, and J. Fitch. Asset managing the power delemma. *IET Control and Automation Magazine*, pages 40–45, October 2007.

[128] N.R. Jennings and M. Wooldridge. Agent-oriented software engineering. In *Proc. 9th European Workshop on Modelling Autonomous Agents in A Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99)*, volume 1647, pages 1–7, London, U.K., February 1999. Spinger-Verlag.

[129] M. Wooldridge, N.R. Jennings, and D. Kinny. The Gaia methodology for agent-oriented analysis and design. *J. Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, September 2000.

[130] V. Honavar, L. Miller, and J. Wong. Distributed knowledge networks. In *Proc. IEEE Information Technology Conf.*, pages 87–90, Syracuse, New York, U.S., September 1998. IEEE.

[131] D. Caragea, A. Silvescu, and V. Honavar. Towards a theoretical framework for analysis and synthesis of agents that learn from distributed dynamic data sources. In *Emerging Neural Architectures Based on Neuroscience*, volume 2036/2001 of *Lecture notes in computer science*, pages 547–559. Springer Berlin / Heidelberg, Berlin, January 2001.

[132] M. Sloman. *Network and distributed system management.* Addison-Wesley Longman, Edinburgh Gate, Harlow, Essex, U.K., first edition, June 1994.

[133] A. Zaher and S. D. J. McArthur. A multi-agent fault detection system for wind turbine defect recognition and diagnosis. In *Proc. PowerTech Conf.*, pages 22–27, Lausanne, 2007.

[134] C. Hewitt. Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 8(3):323–364, 1977.

[135] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[136] M. Wooldridge and N. R. Jennings. *Intelligent Agents.* Heidelberg: Springer Verlag, 1995.

[137] S. Vere and T. Bickmore. A basic agent. *Int. J. Computational Intelligence*, 6(1):41–60, 1990.

[138] P.R. Cohen and H.J. Levesque. Intention is choice with commitment. *Int. J. Artificial Intelligence*, 42(3):213–261, 1990.

[139] Foundation for Intelligent Physical Agents (FIPA). Available at http://www.fipa.org/, 2000.

[140] F. Bellifemine, G. Caire, and D. Greenwood. *Developing multi-agent systems with JADE.* John Wiley, U.K., 2007.

[141] S. Poslad, P. Buckle, and R. Hadingham. The FIPA-OS agent platform: Open source for open standards. In *Proc. of the 5th Int. Conf. and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, pages 355–368, Manchester, U.K., 2000.

[142] M. Shamsfard and A.A. Barforoush. Learning ontologies from natural language texts. *Int. J. Human-Computer Studies*, 60(1):17–63, January 2004.

[143] M. Grobelnik, D. Mladenic, and M. Jermol. Automatic ontology construction from education materials on the web for a large publishing house. In *Proc. Data Mining and Warehouses Conf. at Multi-Conf. IS-2002*, Sydney, Australia, 2002.

[144] C. Blaschke and A. Valencia. Automatic ontology construction from the literature. *ACM Trans. Information Systems*, 13:201–213, 2002.