A Model-Based Alarm Processor for the Assessment of Protection System Performance

Catherine Joyce Edwards

Submitted for the Degree
Of
Doctor of Philosophy

Institute for Energy and Environment
Department of Electronic and Electrical Engineering
University of Strathclyde
Glasgow, G1 1XW
Scotland, UK

2013

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

Signed:

Date:

Abstract

Within the power industry engineers are increasingly faced with demands for improvements in performance and reliability of the electrical equipment under their supervision. Power grids are made up of a multitude of electrical components, combined with protection equipment in place to ensure expensive assets are protected from damage. Attempting to effectively manage the sheer volume of electrical plant becomes an impossible task without remote monitoring and automated analysis of the huge volumes of data produced. During major storm conditions the information produced from network monitoring can also become unmanageable. Volumes of operational alarms, for example SCADA (Supervisory, Control and Data Acquisition) data, can reach up to 65,000 alarms within 24 hours during storm conditions.

This thesis presents the use of model-based reasoning for the analysis of SCADA data for the validation of protection performance and the identification of network incidents. The base requirements were determined in conjunction with a transmission system operator within the United Kingdom with a major task identified in reducing software maintenance whilst maintaining an accurate diagnostic result. The novel approaches taken in this system include the provision of dynamically configurable protection device models for reuse on a multitude of different network connections and the modification of the GDE (General Diagnostic Engine) to identify the occurrence of missing alarms, previously not having been applied to data of this form. The approach taken ensures that the system is easy to update as the network evolves, new equipment is added and data formats change. Models of protection devices combined with the GDE provide the ability to diagnose the maloperation of protection equipment, including multiple faults, missing alarms and new or previously unknown faults.

The implemented system, PSDiagnosis, is examined including its performance as an on-line system developed for a utility partner.

Acknowledgements

I would like to thank Professor Stephen McArthur in the Advanced Electrical Systems Group, University of Strathclyde for his continued support and guidance. Special thanks must also go to Euan Davidson for sharing his experience in artificial intelligence and power system diagnosis, and his continued support when my confidence dwindled throughout this research. Within the Advanced Electrical Systems Group special thanks should go to Vic Catterson, Scott Strachan and Adam Brown.

I would also like to thank ScottishPower Energy Networks for supporting this research, in particular thanks must be paid to Ian Watt and Tom Cumming for their contributions and patience during this research.

I would like to give special thanks to my husband, Jason, without his encouragement, patience and support I would not be completing this thesis. I would also like to thank my family, in particular my parents, who have always been a huge influence and support in my life and I appreciate greatly. I would also like to thanks my friends Anna Johnson, Kirsty Murray and Rachael Storry for putting up with my rants and helping me maintain some level of sanity in the office.

Contents

List of Figures

List of Tables

Glossary of Abbreviations

AC	Alternating Current
AI	Artificial Intelligence
CB	Circuit Breaker
CML	Customer Minutes Lost
CT	Current Transformer
DAR	Delayed Automatic Recloser
DES	Discrete Event Systems
DFR	Digital Fault Recorder
DSS	Decision Support System
FR	Fuzzy Reasoning
GA	Genetic Algorithm
GDE	General Diagnostic Engine
GPS	Global Positioning System
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
INT	Intertrip
IS	Intelligent Systems
KADS	Knowledge Analysis and Design Support
KBR	Knowledge-Based Reasoning
KBS	Knowledge-Based System
KCL	Kirchoff's Current Law
MBD	Model-Based Diagnosis
MBR	Model-Based Reasoning
MMS	Manufacturing Message Specification
MP	Main Protection
PEDA	Protection Engineering Diagnostic Agents
PN	Petri-Net
RBES	Rule-Based Expert System
RBS	Rule-Based System
RTU	Remote Telemetry Unit
SCADA	Supervisory, Control and Data Acquisition

SCL	Substation Configuration Language
SIR	Source Impedance Ratio
SPEN	ScottishPower Energy Networks
TR	Trip Relay
TWL	Travelling Wave Locator
VT	Voltage Transformer
XML	eXtensible Markup Language

Chapter 1

Introduction and Justification of the use of Model-Based Reasoning for Protection Performance Assessment

1.1 Introduction to the Research

Over the last 25 years engineers have attempted to produce automated post-fault analysis systems. A major part of the analysis process involves validating the behaviour of the protection equipment which has operated during fault conditions. Without accurate protection analysis, faulty or incorrectly configured protection equipment can remain undiagnosed. This can then lead to increased Customer Minutes Lost (CML) during faults, as well as damage to other equipment. The analysis of operational data can involve upwards of 30 man days of analysis time when storm conditions have occurred. The combination of these large volumes and the increased demands on experienced power engineers, identified as one of the key engineering groups with a shortage of experienced workers [Grice et al., 2011], means an automated analysis solution is vital. The analysis problem heightened with the continued increase in smart grid technologies, and the uptake of new standards such as IEC 61850 [IEC, 2005]. Although this uptake will standardise data formats across utilities, the consequences of increased data volumes and data format changes must be accounted for. This thesis presents a solution utilising an adapted General Diagnostic Engine (GDE) [Forbus and de Kleer, 1993] and providing an accurate, low-maintenance solution for SCADA-based protection validation.

A common approach to the problem is the use of heuristic knowledge [Wei et al., 2011, Pfau-Wagenbauer and Nejdl, 1992a, Krivine and Jehl, 1996, Hor et al., 2007, Hossack et al., 2003b, Vale et al., 1999] in the form of an expert system. However expert systems are liable to suffer inaccuracies due to their reliance on expert knowledge rather than first principle knowledge of the system being analysed; these inaccuracies can include failure to diagnose new faults or even misdiagnosis. In contrast, various systems have taken a model-based diagnosis approach for post-fault analysis [McDonald et al., 1992, Chantler et al., 2000, Davidson et al., 2006, Malheiro et al., 2007, Kezunovic et al., 2010, Lamperti and Zanella, 2011], but only in some of these cases [Malheiro et al., 2007, Kezunovic et al., 2010, Lamperti and Zanella, 2011] has model-based diagnosis been used on SCADA (Supervisory, Control and Data Acquisition) alarm data. However these applications have a different focus to this thesis either in being based on post-fault diagnosis rather than protection validation, or on the validation of protection operation when further information is available such as distance to fault (i.e. enabling the validation of distance protection operation). In the other examples model-based techniques are applied to discrete time-series voltage and current measurements such as DFR (Digital Fault Recorder) data where higher fidelity information is available and details such as fault type and distance to fault can be calculated. The approach proposed in this thesis is unique in its use of the GDE to carry out protection validation using operational SCADA data alone. The GDE provides a means of creating a generic diagnostic process for the protection network. The adapted GDE discussed in this thesis has been developed to carry out missing alarm identification, and to allow automatically generated and configured models to be utilised. This is ideal for the protection validation problem as it allows different configurations to be dynamically modelled and accurately diagnosed without the need for fault models.

With the increased uptake in the use of IEC 61850 devices, there is evidence that systems are being designed to handle the new data formats relevant to this standard [Wei et al., 2011]. Unfortunately with the adoption of IEC 61850 devices currently in its preliminary stages and the majority of utilities relying on older legacy equipment, systems developed solely for IEC 61850 would need to be adapted to process the technologies currently in use. A problem area therefore remains in designing a system which is capable of providing accurate protection performance information which can handle both legacy data and new formats.

A method of providing this flexibility to the system is to provide a means to independently update each of the system's operational components. By utilising the GDE which relies on behavioural models of the network, the analysis system can be designed to abstract the alarm formats from both the connectivity and models of the network. This abstraction allows the independent modification of alarm data, network connectivity and devices. Therefore any format changes, either within a utility or between utilities, can easily be integrated into the system without affecting the integrity of the system as a whole. These format changes could include the introduction of IEC 61850 based protection models or data. This would also cater for updates to the SCADA system, such as new device identifiers or locations.

1.2 Principal Contributions

This section discusses the contributions, in terms of the academic novelty and industrial impact, of the research reported in this thesis. These contributions are summarised as follows:

- Research, design and development of a model-based alarm processor for the assessment of protection system performance.
  - Investigation of intelligent techniques, their application and applicability to SCADA data, protection validation and post-fault diagnosis.
  - Knowledge elicitation with protection experts providing the means to create configurable models of protection equipment.
  - Design, development and testing of rapid prototype for a single protection scheme.
  - Design, development and testing of a novel protection database providing a flexibly link between different SCADA data and protection models.
  - Extending previous model based approaches by including models of Delayed Automatic Reclosers (DAR) and missing alarm identification.
- Adapting the GDE to handle temporal discrete events on dynamically generated models.
- On-line case-studies on real world protection system SCADA data.
- Examination of system performance based on incident identification, classification and faulted component selection.
- Providing the potential for the system progression into both full and partially IEC 61850 networks.
- Proposal of extension of analysis into other forms of data such as DFR and utilising such data for circuit breaker condition monitoring and prognostics.

1.3 Thesis Overview

This thesis starts with two chapters of background information. Chapter 2 provides a description of the needs and manual processes involved in the validation of electric power system protection devices. Chapter 3 follows with a description of the state-of-the-art in the applications of intelligent techniques for the power industry and why they are being considered for protection performance assessment. Chapter 4 describes the reasoning behind the selection of the General Diagnosis Engine (GDE) for protection performance assessment, the adaptations involved and the inherent flexibility built into the performance assessment system. Chapter 5 examines the results and evidence gathered from off-line and deployment case-studies including the workings of the system for a range of circuits and fault conditions. Chapter 6 summarises the work presented in this thesis, and indicates where research questions remain for the future.

1.4 Associated Publications

- E. M. Davidson, S. M. Strachan, C. J. McKinlay, S. D. J. McArthur. Automated fault and disturbance analysis and its role in the smart grid. PAC World Conference, Dublin, June 2010.
- C.J Edwards, E.M. Davidson, S.D.J. McArthur, I. Watt, T. Cumming. Flexible Model-Based Alarm Processing for Protection Performance Assessment and Incident Identification. IEEE Transactions on Power Systems, 2013, Early Access Paper. [Edwards et al., 2013]

Chapter 2

Assessing the Performance of Power Systems Protection

2.1 Introduction

This chapter describes the process undertaken by the utility partner in the analysis and assessment of protection performance based on SCADA alarms and how this guides the further analysis on DFR and TWL (Travelling Wave Locator) data. It begins by describing the fundamental protection system devices, protection scheme operation and the relevant data sources. Following this, the protection performance assessment process is outlined thereby explaining the challenges behind the research and the starting point for the novel diagnostic system design.

2.2 Protection Devices

The power network must be operated safely to prevent damage to both the equipment on the network and the life or property of those around it. Dangerous conditions can be created on the power network by a range of issues including poor weather conditions, misconfiguration of or faulty equipment. In order to minimise the danger associated with faults the network is observed by protection devices which provide the ability to disconnect sections of the network when dangerous conditions are detected. This section discusses the high level function of a number of standard devices which make up typical power network protection schemes.

2.2.1 Measurement Transformers

In order to provide measurable voltage and currents, which are proportional to those occurring on the primary system, accurate voltage transformers (VTs) and current transformers (CTs) are required. Although the analysis of VT and CT performance cannot be determined through SCADA data alone the analysis of their behaviour can be performed using DFR data forming part of the further work for this project.

2.2.2 Main Protection

Main protection (MP) devices are a means to detect the occurrence of faults by utilising the measurements provided by CT and VT devices. For the scope of this thesis there are two primary main protection functions, each of which is expected to operate when voltage, impedance or current measurements deviate from the expected.

Unit Protection

Unit protection devices operate when some form of abnormal current is detected. This detection involves the comparison of (transformed) current in communication with the remote end of a line. Unit protection utilises Kirchoff's current law (KCL), to identify the occurrences of fault within the line of its protection. KCL on a single line can be simplified as:

- $I_{IN} = I_{OUT}$

Where KCL is broken, a fault must have occurred between the two points of measurement. Within AC systems this comparison must include the comparison of both the magnitude and phase angle of the current measurements.

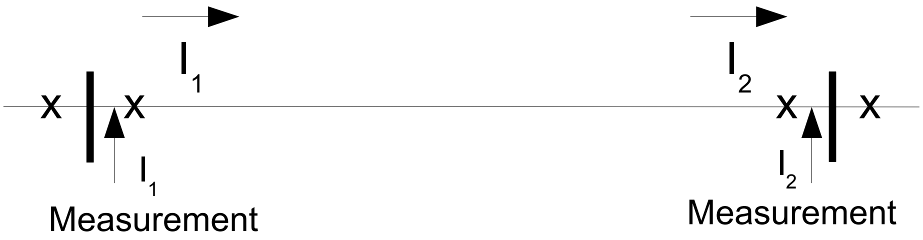


Figure 2.1: Unit Protection Measurement Points

Following the diagram shown in Figure 2.1, unit protection shall operate when

- $I_1 \neq I_2$

Figure 2.3 presents the theory behind unit protection in the detection of internal faults. During a fault within the line being protected both currents will flow into the fault, feeding the fault current and resulting in imbalance between each end of the line.

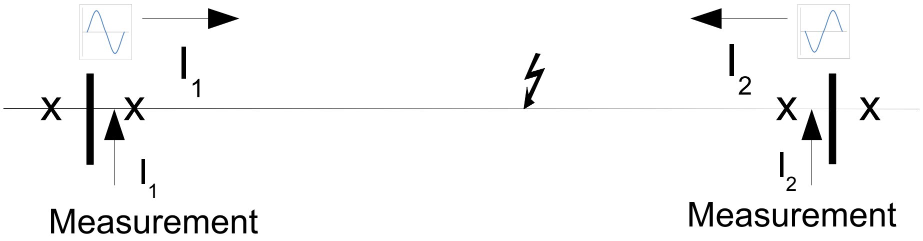


Figure 2.2: Unit Protection with Internal Fault

Figure 2.2 presents the theory behind unit protection during the occurrence of external faults. During a fault externally to the line being protected both currents will flow equally into the fault, feeding the fault current and may observe an increase in current measurements but will not have an imbalance between each end of the line.

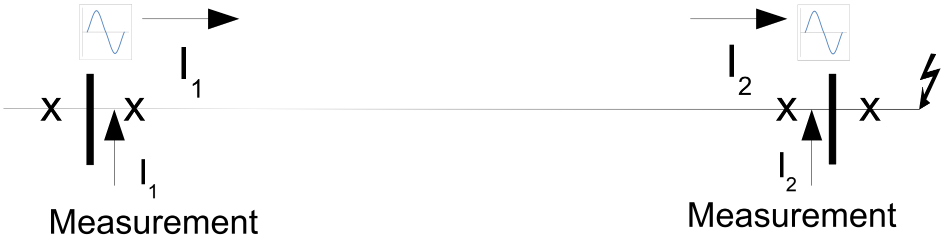


Figure 2.3: Unit Protection with External Fault

Distance Protection

Distance protection utilises the relationship between voltage, current and impedance along with their typical behaviours when faults occur at different distances along transmission lines. During faults the following relationships can generally be observed:

- Fault close to measurement:
  - LOW Voltage
  - HIGH Current
  - LOW Impedance
- Fault far from measurement:
  - HIGH Voltage
  - LOW Current
  - HIGH Impedance

It should be noted that during special cases, for example where the source impedance of a line is significant in comparison to the impedance of the line itself, these relationships can be affected. The effect of source impedance can be provided in the system impedance ration (SIR):

$$\frac{Z_{SOURCE}}{Z_{ZONE}}$$

Where Z_{SOURCE} is the source impedance and Z_{ZONE} is the impedance of the protected zone.

High SIR causes both increased resistive distortion during impedance calculations and reduces the level of fault current accross the line. Therefore the calculation of impedance can be distorted and calculations vary less between close and far away faults. Similar problems are applicable to short lines, where very small impedance changes can signify a high percentage change in fault position. Trying to measure these changes can be difficult and at some levels even the small tolerances involved in transforming the voltage and current levels for measurement would need to be taken into account.

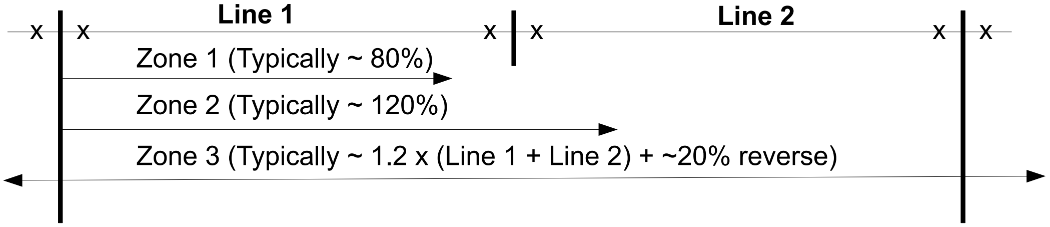

x
x
Line 1
x
x
Line 2
x
x
Zone 1 (Typically ~ 80%)
Zone 2 (Typically ~ 120%)
Zone 3 (Typically ~ 1.2 x (Line 1 + Line 2) + ~20% reverse)

Figure 2.4: Distance Protection Zones

Impedance measurments provide the ability to detect both the existence of faults but also dependent on the distance of the fault determine the correct time in which to react. I.e. Close faults have higher current, making their existence more damaging to local equipment, therefore need a faster reaction time. Further away faults would normally be cleared by the remote end so can be set up as a back up operation for local equipment.

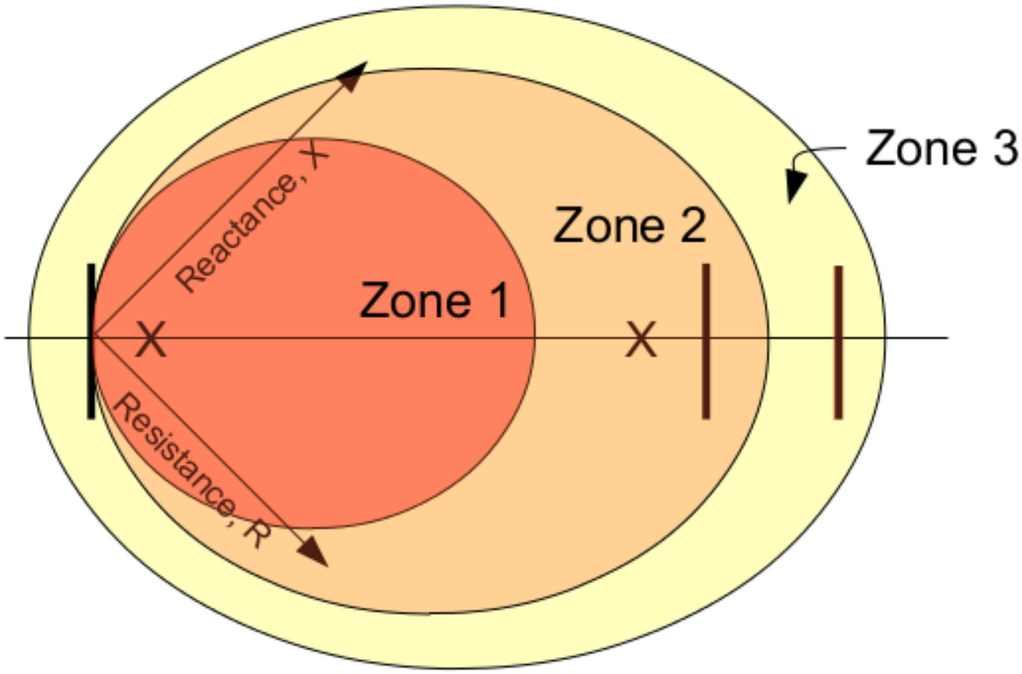


Figure 2.5: Distance Protection - Impedance Zones

Distance protection devices usually divide the network into three or four zones of protection. Each zone protects a particular length of the line, with increasing operation times as the distance to fault increases. Example distance and impedance zones are presented in Figure 2.4 and Figure 2.5. The zones presented are a typical Mho (as shown) configuration, other zone configurations include; Lenticular (Used for long heavily loaded lines) and Quadrilateral (Used for better earth fault coverage) This changes the shape of each zone in comparison with X and R.

Typical operational times for each zone are:

- Zone 1 - Instantaneous
- Zone 2 - Between 300 and 600ms (15-30 cycles)
- Zone 3 & 4 - >600ms (>30 cycles)

Communications Enhancement of Distance Protection

Distance protection can be enhanced by allowing remote distance protection devices to send communications to each other. This can provide a means to utilise distance protection more effectively on smaller lines and prevent both unnecessary trips and trip delays.

Accelerated distance protection operates by utilising the understanding that where a zone 1 fault is observed at one end of the line, this should be seen as either a zone 1 or zone 2 fault at the remote end. Where two zone 1 faults are observed instantaneous tripping will occur at each end.

However, if one end only sees the fault within zone 2, it will have a delayed trip of at least 300ms. Accelerated distance schemes provide a means for zone 1 faults to be transmitted to the remote end of a line, once transmitted the signal is used to accelerate the time in which the remote end would normally trip on a zone 2 fault. Therefore it allows internal faults to be cleared more quickly, reducing zone 2 trip time to 45ms (communications propagation and signal processing delays).

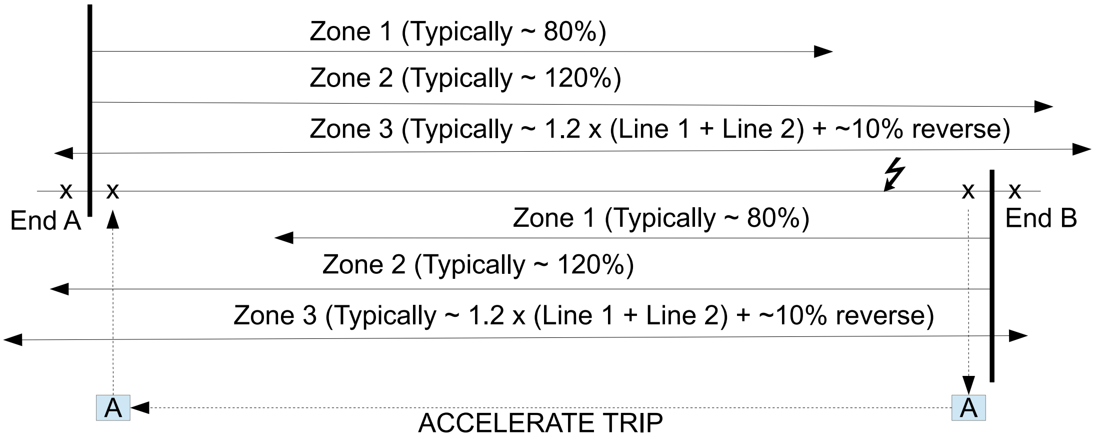
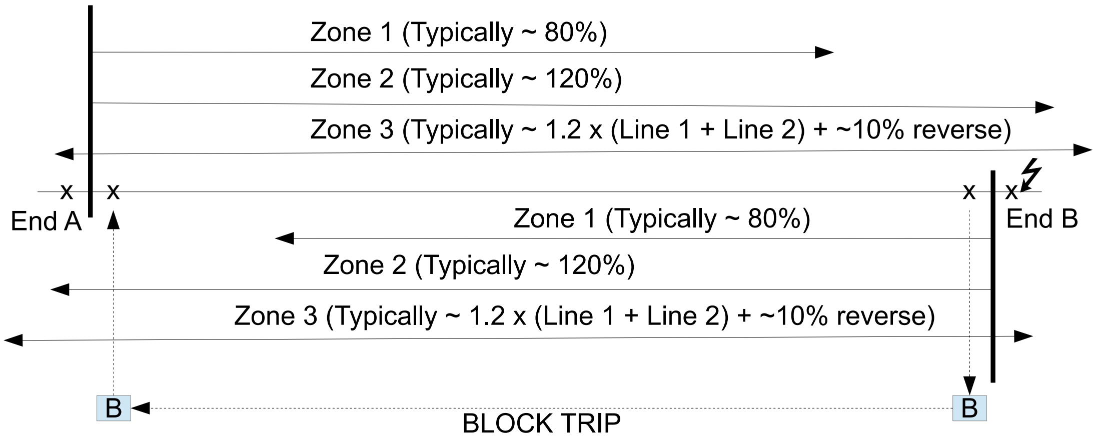

Zone 1 (Typically ~ 80%)
Zone 2 (Typically ~ 120%)
Zone 3 (Typically ~ 1.2 x (Line 1 + Line 2) + ~10% reverse)
x
x
End A
x
x
End B
Zone 1 (Typically ~ 80%)
Zone 2 (Typically ~ 120%)
Zone 3 (Typically ~ 1.2 x (Line 1 + Line 2) + ~10% reverse)
A
A
ACCELERATE TRIP

Figure 2.6: Accelerated Distance Protection

Within Figure 2.6 an example is given of a fault that lies within End B's Zone 1 and End A's Zone 2. For this situation End B would send an acceleration signal to End A, thus accelerating tripping time at End A.

Blocking distance protection operates by utilising the understanding that where a zone reaches onwards to a segment of a second line, the reverse zone, a fault detected within this zone can be used to inform remote ends that they should not need to trip. Blocking distance schemes can be seen to effectively operate as unit protection, making observations that the fault is either within or external to the line it is protecting and tripping or not tripping based on this information.

A blocking distance example can be observed in Figure 2.7 for this example the reverse zone identifies a fault lying within the adjacent line. For this situation End B's distance protection would send a block signal to the remote end, blocking its zone 2 operation.



Zone 1 (Typically ~ 80%)
Zone 2 (Typically ~ 120%)
Zone 3 (Typically ~ 1.2 x (Line 1 + Line 2) + ~10% reverse)
x
x
End A
x
x
End B
Zone 1 (Typically ~ 80%)
Zone 2 (Typically ~ 120%)
Zone 3 (Typically ~ 1.2 x (Line 1 + Line 2) + ~10% reverse)
B
B
BLOCK TRIP

Figure 2.7: Blocked Distance Protection

2.2.3 Intertrip

Intertrip (INT) devices provide a means of communicating the identification of a fault between substations. Intertrip devices are used to send unconditional tripping signals to remote ends of circuits as opposed to the signals associated with accelerated or blocking protection. Due to this unconditional tripping intertrips require additional levels of security to reduce the risk of incorrect tripping. As additional security is required before signals are able to be be accepted, intertripping is generally slower than main protection schemes but provides an important means of transmitting the need to trip to remote ends.

2.2.4 Trip Relay

The function of a trip relay (TR) on the protection system is to send the trip command to a circuit breaker when it receives notification of a fault, usually from one of the main protection devices in the scheme.

2.2.5 Circuit Breaker

The purpose of a circuit breaker (CB) is to open and interrupt fault current during fault conditions. It is important that a circuit breaker responds in a timely manner during a fault to limit the damage to itself and other equipment on the network. If a circuit breaker is delayed in its operation, any extended period of time spent feeding fault current can lead to increased system instability and potential damage to surrounding equipment such as transformers. Alternate delays caused by slow pole movement during faults can lead to damage on the circuit breaker itself, where the movement is having to take additional strain while extinguishing a fault arc.

2.2.6 Delayed Automatic Recloser

The purpose of a Delayed Automatic Recloser (DAR) is to re-close a circuit breaker after a delay, DARΔ. The delay is set to allow ample time for fault clearance while ensuring line downtime is kept to a minimum, typically in the region of 10 to 25 seconds. Longer DAR times are utilised to allow staggered tripping on double circuits, providing a means to prevent both circuits closing onto persistant tower fault which would effect both circuits. They are also utilised to allow staggered close over T-shaped circuits again to prevent any unnecessary circuit breaker cycles onto faults.

Although DARs are commonly used within the United Kingdom power network, other automatic reclosure systems such as high speed automatic reclosers (HSAR) are available. HSARs operate more quickly than DARs, typically between 60 and 400ms. Typical reasons for the selection of DAR over HSAR include:

- High speed reclosure onto persistent or permanent faults can result in system instability caused by multiple faults being introduced to the network over a very short time frame.
- Generator oscillations produced during faults are given more time to dampen before a potential switch onto fault.
- Ionisation clouds around circuit breakers are able to be given sufficient time to disperse
- Air-blast circuit breakers are able to be operated more easily as the risk of low air pressure is reduced if additional time is given between operations.
- Autoisolation systems can be used during DAR times to provide isolation of faulted plant, for example in the case of a damped transformer fault.

Therefore although HSAR schemes exist within the United Kingdom network they are generally installed within circuits which can handle system instability, for example DC interconnectors which by nature dampen system oscillations.

Where automatic reclosure is utilised there is the potential for the circuit breaker to be closed onto a fault. When a circuit breaker has been closed onto a fault there is the potential for a few different scenarios:

- The automatic reclosure can be reset and attempt another automatic reclosure after a specific delay. This can be set to attempt reclosure for a set number of times.
- The circuit can be locked out, meaning that automatic reclosure will not be possible and either manual or remote reclosure is required. This would occur after the number of attempts on an automatic reclosure were exceeded.
- The circuit breaker, particularly air-blasted breakers, which rely on compressed air to operate could physically lockout. If the circuit breaker does not have enough air to carry out a full

close and open cycle it will not be able to be reclosed until the air pressure has increased.

2.3 Protection Schemes

Protection schemes are made up of different protection devices, interconnected in order to protect a part of the power network. For the scope of this thesis the protection scheme can be divided into three sections:

- Detection and Tripping - Including MP and TR devices
- Circuit Breaker Opening - Including INT, TR and CB devices
- Circuit Breaker Automatic Reclosure - Including INT, TR and DAR devices

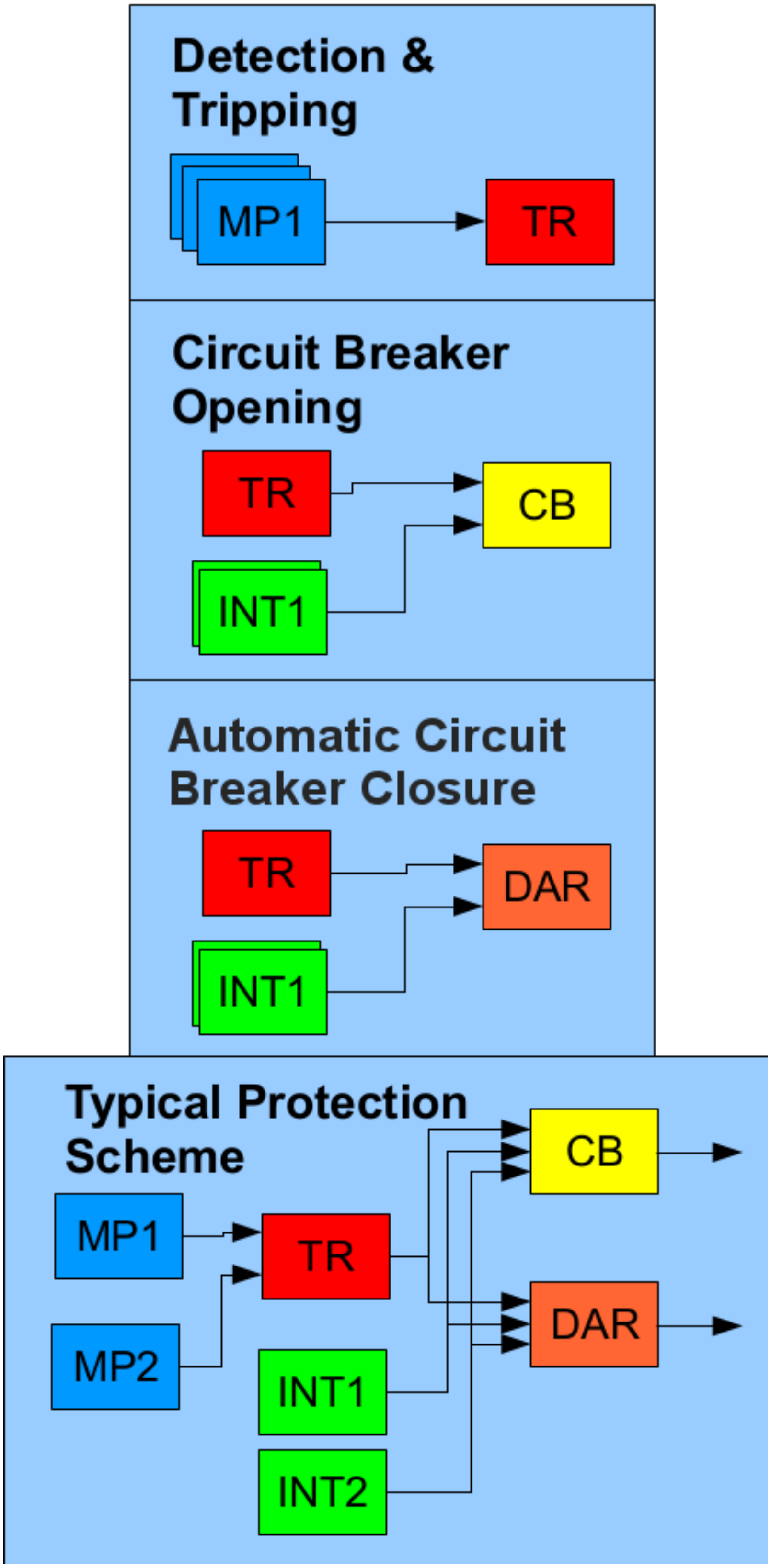


Figure 2.8: Typical Protection Scheme - The different sections of a protection scheme and their connectivity.

2.3.1 Detection and Tripping

In order to react to faults on the network the main protection devices must not only detect a fault but pass on this information to other devices on the protection scheme. Main protection devices send their signals to the trip relays, which are then triggered to operate circuit breakers at both ends of the faulted network section, isolating the fault. This tripping involves the transmission of trip signals to both local trip relays and through the use of intertripping signals for simultaneous tripping at both ends.

2.3.2 Circuit Breaker Opening

Once a fault has been identified, either locally via main protection devices or remotely via intertrips, a signal is passed from these devices to operate the circuit breakers surrounding the faulted section. Assuming circuit breakers operate correctly the result is the interrupting of any fault current on the line being protected.

2.3.3 Automatic Circuit Breaker Closure

Once a fault has been identified, either locally via main protection devices or remotely via intertrips, a signal is passed from these devices to start a timer which then allows a count down to automatic reclosure for circuit breakers. Where a transient fault has occurred, e.g. line clashes, the successful operation of the DAR device will reclose the circuit breaker and return the protected circuit to operation. Unfortunately permanent faults, or lengthy/repeating transient faults the protected circuit will be not be able to be returned to service via DAR. These occurrences generally result in DAR lockout, where the DAR device has attempted automatic reclosure a set number of times with no success, and the circuit is deemed to be unable to be returned to service at this time. On these occurrences engineers observing the network would be required to carry out manual reclosure after further investigation or on permanent faults initiate a repair.

2.4 SCADA & Other Data

In order to analyse and observe the operation and performance of protection schemes, their operation is recorded. This thesis focuses upon the utilisation of one of these recordings, SCADA data, with further work demonstrated for the integration of other data sources.

2.4.1 SCADA Alarms

SCADA alarms signify the operation of protection devices. SCADA alarms vary greatly amongst transmission operators however as SCADA alarms are generally the first indication operators receive of any network disturbance, their reception is commonly used as a trigger for performing post-fault analysis. The variations between transmission operators include: coverage, time synchronisation and alarm content (including different syntax and detail). This thesis focuses upon maintaining a loose coupling between alarm content and the protection scheme models in order to provide a more flexible system. The system has been trialed within a United Kingdom transmission operator where the prominent features are:

- Relatively low resolution, ±10ms (time-stamping can result in 10ms time-stamp referring to operations between 10 and 19ms)
- Limited to no GPS time synchronisation between sites - simultaneous events could have differing timestamps (e.g. typically in the region of tens of milliseconds to a few seconds apart)
- 100% network coverage

This means that while time stamps within a specific site may be accurate enough for alarm sequencing within that site, they are not accurate enough for comparison and sequencing across multiple sites. Therefore the construction of protection scheme models must be capable of operating on a site by site basis with the ability to move to a multi-site model in the future. It should be noted that although SCADA alarms can include additional identifiers, such as the zone in which protection operates, because not all devices can produce this type of alarm these are deemed an unreliable source of information and therefore are not analysed.

2.4.2 Digital Fault Recorders

Digital Fault Recorders (DFRs) are GPS (Global Positioning System) time-synchronised recorders used to monitor the signals observed during a fault. These signals include three phase voltage and current measurements, digital device operations and in some cases trip coil traces. Current and voltage signals can be sampled at greater than 25.6kHz, allowing 50Hz AC signals to be sampled over 512 times per cycle, therefore providing accurate representations of faults occurring on the network. Due to the accurate GPS provision of timings the DFR data can be used to determine not only the behaviour of the fault but also reveal the actual operating time of the protection equipment. DFR data is generally reported to engineers within one to twenty-four hours of production, therefore with DFR data suffering from significant delays in comparison to SCADA data, and often poorer network coverage, it cannot be relied on in the same manner. The reasons for delays of this type can be caused by a range of issues. A major issue faced by some network operators relates to the use of modems for connections to such devices, thus rendering the download of records in realtime infeasible. The network operator involved in this research as an example in 2013 has over 75% network coverage (lowest coverage 40% at 11kV, highest coverage 90% at 400kV).

2.4.3 Travelling Wave Locator Data

Travelling Wave Locators (TWL) are very high frequency, 1.25MHz, GPS time-tagged fault locators. They provide accurate fault location (distance-to-fault) information, within ±300m or one tower span. Although distance-to-fault calculations can be carried out using normal line impedance and digital fault records, there are cases where this is inaccurate (for example during more complex faults when the impedance to fault can be skewed) and therefore where available travelling wave locators should take precedence. TWL data as with DFR is generally delayed and available on fewer circuits than SCADA data.

2.4.4 Protection Database

The protection database is a relational database produced in order to provide an easily maintainable data store for network information. This network information includes:

- SCADA circuit identifiers - linking SCADA alarm identifiers to specific circuits and devices.
- Circuit information - number of CB's, INT's, MP's, DAR's
- Specific & Generic Timing information - generic acceptable timing information for each protection device plus the ability to add "exceptions to the rule" for different settings i.e DAR timings, specific CB models.
- Alarm text - linking utility specific alarm text, e.g. "1ST MAIN PROT OPTD" to the specific device, e.g. MP1.

This relational database although utility specific in its content has been produced to allow information from any source to be built into its architecture. It was designed as part of this project to bring together the utility data required for analysis to take place, and provide a means to store and update this data effectively.

2.5 Manual Analysis Process

The major motivation for this research is to propose and provide an automated analysis system to aid power engineers in the analysis of network incidents. This section provides the information gathered during knowledge elicitation into the process involved in incident identification and protection validation for the transmission network.

2.5.1 Data Volume

The volume of data received by ScottishPower Energy Networks has been rapidly increasing over the last decade. Comparing data from 2001 and 2010, the annual SCADA data volumes have doubled, with annual DFR volumes tripling. During a period of six months, February to July 2010, the incoming data at ScottishPower Energy Networks was collected and charted. This data graph can be seen in Figure 2.9.

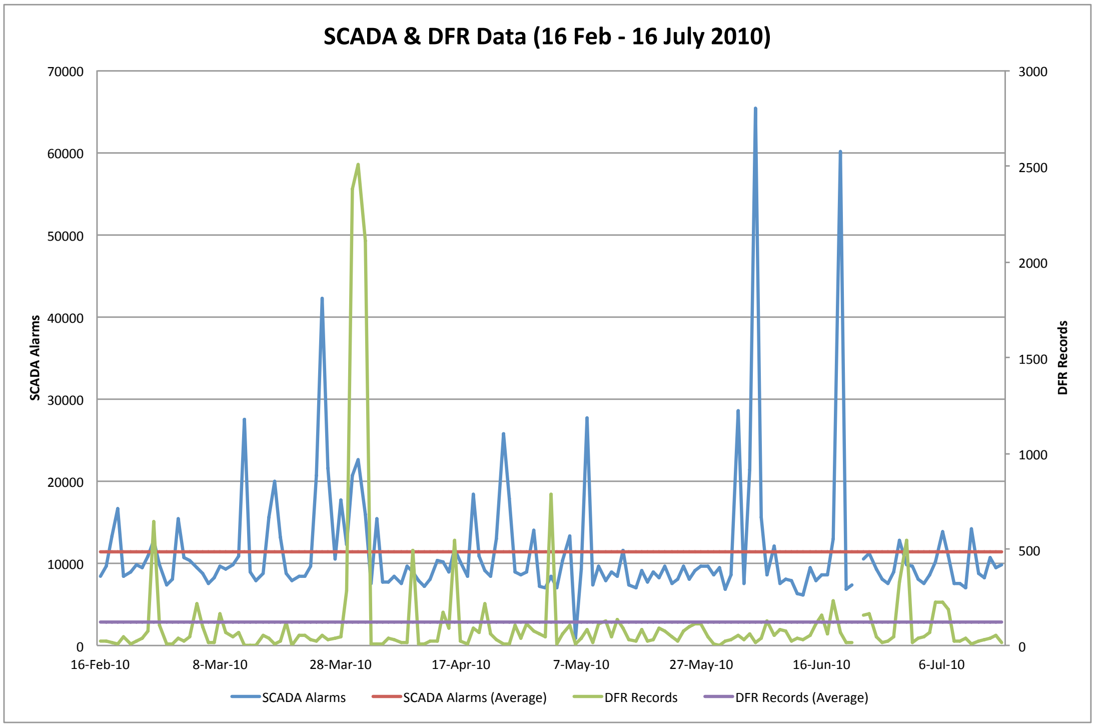


Figure 2.9: Six months of SCADA and DFR volumes.

The information gathered from the charting of these data volumes shows the need for not only an accurate analysis system but also data filtering. Major faults can be seen within the DFR peaks, 24th-25th February and 30th-31st March. However the major SCADA peaks can be seen on the 5th and 19th of June, where over 65,000 alarms were produced within a 24 hour period. This kind of peak is due to major communication issues and demonstrates the need for effective data filtering. Although certain communications issues could be silenced, the arrival of SCADA communication problem alarms is necessary to allow diagnostics to occur and for communication fixes to be confirmed.

2.5.2 Analysis Process

The current manual analysis process carried out during network disturbances is largely dependent upon the arrival time of the relevant data sources. Within different transmission operators this arrival time can vary. SCADA data is generally the first data to arrive, arriving within seconds of incident occurrence, followed by DFR and TWL data, with arrival time ranging from minutes to hours from incident occurrence [Bi and Li, 2007]. During requirements elicitation with operational protection engineers, it was found that a desirable time frame for initial incident identification would be one to two minutes from incident closure. Therefore although the analysis of DFR-type data, whether manual or automated [Davidson et al., 2006, Chantler et al., 2000, Luo and Kezunovic, 2005], is important the delay in its arrival results in the need for initial incident identification by SCADA alone. The requirements to automate this analysis process can be summarised as:

- *Determine disturbance location* - identify the section of the network affected by the incident including the identification of any protection devices where a maloperation has occurred.
- *Analyse "out of order" alarms* - identify and correctly analyse alarms which may have been delayed due to communications errors.
- *Filter irrelevant alarms* - up to 90% of SCADA alarms received can be unrelated to actual disturbances on the network, such as communications synchronisations. These alarms are unnecessary for this type of assessment, and therefore should be filtered out.
- *Group alarms into incidents* - utilising information about the protection network and SCADA alarm time stamps, group alarms as they arrive into "incident" groups which can provide a snap shot of current conditions on the network.
- *Identify missing alarms* - missing alarms have been reported in a number of transmission operators. The identification of missing alarms is vital in order to prevent the misdiagnosis of correctly operating devices. [Vale et al., 1999, Hor et al., 2007]
- *Validate the performance of protection equipment* - based on both the operation or non-operation of a device and temporal responses.
- *Provide links to further data sources* - based on SCADA data attempt to aid engineers by linking to the further data sources such as DFR and TWL so that when this data is available, analysis can occur in a timely manner. Make sure that if and when higher fidelity information is available, it is able to be utilised.

The analysis process can be divided into three subtasks based on the arrival time of relevant data sources; first SCADA based analysis, followed by DFR and TWL analysis.

First Pass - SCADA Analysis

Until more information becomes available, a first-pass of analysis of SCADA data is carried out. It should be noted that some transmission operator SCADA alarms are susceptible to severe time skew, caused by both a low resolution time-stamp (of 10ms) and a lack of GPS time synchronisation throughout interconnected substations. Therefore the first pass has a degree of inaccuracy in relation to actual protection operating times between sites and therefore the analysis must be carried out on a site by site basis where GPS synchronisation is unavailable.

This process can be summarised as follows:

Data Retrieval and Collation:

1. SCADA trigger alarm received, notifying engineers that an event has occurred.

2. The SCADA identifier is mapped to a specific circuit. Thus identifying the circuit/s involved in the event.
3. Collate all alarms identified to have occurred within the circuits, within 2 minutes. With 2 minutes providing an adequate window to receive alarms generated over a typical automatic reclosure (up to 30 seconds), re-trip and lockout (up to 60 seconds) and take into account polling delays associated with getting alarms from site back to the main SCADA databse.
4. After the appropriate SCADA alarms are collated, they are grouped by substation and sequenced according to their local SCADA time-stamps.

The next stages involve carrying out analysis on the SCADA alarms that have been received. The results of the analysis are then formatted into an incident report.

Data Analysis
High Level Event Information:

1. Given the sequences of SCADA alarms that have been produced, the high level event can be classified. This includes:
   - Event Timings
   - Event Severity
   - Event Status (Successful Automatic Reclosure?)
2. Using knowledge of the equipment connectivity the maloperation of equipment can then be identified. This includes:
   - Using generic knowledge of the acceptable response timings for equipment such as trip relays and circuit breakers the received alarms can be used to check that equipment operated within the acceptable response range.
   - Using the DAR settings stored within the protection database the operation of DAR can be verified to ensure it operated as expected.

Second-Pass - DFR Analysis

Unfortunately due to polling time scales for digital fault records the data cannot be analysed in real time, instead it can be hours before the system becomes aware of recorded data. However when the data arrives it can be used to further both the depth and accuracy of fault analysis and potentially provide condition monitoring information. The network knowledge combined with the earlier SCADA analysis can be used to identify all relevant DFR records. It should be noted that this requires further research and is included as further work upon the research proposed in this thesis. The following features and measurements can be extracted from DFR analysis:

Fault Classification
At this point fault classification can be carried out through analysis of DFR data. This is done

through a number of analysis techniques including extracting abnormal behaviours and patterns. The aim of this analysis is to define the following:

- Faulted phases
- Potential causes
- Peak fault current

Protection Performance Assessment
DFR data provides further levels of detail to validate protection performance. The characteristics of the fault can then be extracted as follows:

- Accurate fault timing (when the fault itself occurred)
- Clearance times (the time taken for a fault to be cleared)
- Protection operation validation (based upon protection set-up)
  - Measured fault current and impedance measurements (vs. equipment threshold settings)
- Protection timings
  - Response to fault
  - (Accurate) Trip relay response to main protection operation.
  - (Accurate) Circuit breaker response to trip relay.
  - (Accurate) DAR operation.

Condition Monitoring
DFR data provides further levels of detail specifically relevant to the monitoring of circuit breaker and VT devices. This information includes:

- CB - interrupting current
- CB - I^2t
- CB - Maximum fault current
- VT - Abnormalities in signal which could be identified as pointing towards specific VT problems. [Davidson et al., 2008]

Third-Pass - TWL Analysis

Similarly to DFR data, TWL data is not available immediately and is not available until the system polls the substation involved. However when the data arrives it can be used to further both the depth and accuracy of fault analysis, by accurately defining fault location and therefore allowing the validation of protection equipment. The theory behind TWL systems are that they measure the propagation delay of the travelling waves caused by faults to determine fault location. They are therefore ideal for the occurrence of high resistance faults where the use of DFR data would render inaccurate results. It should be noted that the automatic use of this data requires further research and enquiries to discuss the additional requirements - current developments in this area are discussed later.

The outcome of this analysis can be summarised as:

Fault Location
Accurate distance to fault information, if the fault has occurred within the transmission line, travelling wave fault locators can be used to identify the position along the line that the fault has occurred. The location is accurate to within 300m (or one tower span). For permanent faults this helps to focus the recovery operation.

Protection Performance Assessment
In order to validate protection equipment which is set to respond based on distance to fault, i.e. distance protection, accurate distance to fault information must be used. Distance protection operates based on fault zones and therefore can be validated by discovery of which zone the fault has occurred in. Where distance protection has failed to operate correctly given the discovery of the faulted zone, further investigations would be required. For example, accessing relay recordings to determine if the fault was a high resistance fault, which could result in a relay being unable to observe the fault moving into its protected zones.

As demonstrated in this section the manual analysis process for SCADA, DFR and TWL data is a lengthy progress. During knowledge elicitation with protection engineers it was found that in storm conditions, where multiple faults can be occurring on the network simultaneously, over 70 circuit faults occurred within four hours. The analysis of this volume of fault data can take engineers over 10 man days to complete. In comparison during case-studies carried out on the automated analysis system this volume of fault data can be analysed in less than 30 minutes, with prioritisation given to incidents which need immediate action. These results and case-studies are further discussed within Chapter 5.

2.6 Conclusion

This chapter has detailed the typical provision of protection devices, incident raw data and the steps taken to utilise that data during incident identification and protection validation. Whilst this thesis focuses on SCADA data alone with the potential for further work highlighted to combine DFR and TWL data, the SCADA analysis process provides the overview and information required for protection engineers to take action during normal network operation and storm related events. The preceding sections have also demonstrated the availability of both first principle protection device knowledge and the information required to relate this knowledge to specific network data as faults occur.

Chapter 3

Prior Art in Artificial Intelligence for Protection Performance Assessment & Fault Diagnosis

3.1 Introduction to Artificial Intelligence

Artificial intelligence (AI) techniques can and have been used to automate the post-fault analysis process for power systems. As discussed in chapter 2, the post-fault analysis problem for the scope of this thesis involves both large volumes of data and the expert interpretation required to analyse it. The problem therefore naturally lends itself to an AI based approach. This chapter introduces the term AI, with a number of AI techniques which have been used within post-fault analysis and in particular the post-fault analysis of power systems. This chapter will also detail some of the advantages and disadvantages associated with these techniques.

The term AI was first coined in 1955 by John McCarthy [McCarthy et al., 1955] as "the science and engineering of making intelligent machines", since then it has become a subject of considerable debate and research. The definition of AI itself is still the source of debate but can be divided into four main definitions [Russell and Norvig, 2003] shown in Figure 3.1, where a system can be classed as performing one of the following:

- Thinking Rationally - Carrying out logical inferences (Aristotle - "Socrates is a man; all men are mortal; therefore Socrates is mortal")
- Behaving Rationally - Given rational inferences, take rational actions.
- Thinking Human - Tracing the cognitive steps of the human mind.
- Behaving Human - Turing Test [Turing, 1950] - Natural language processing, knowledge representation, automated reasoning, machine learning. The Turing Test involves testing

human imitation against a human interrogator. Where a machine is deemed to have behaved humanly it is also deemed to have behaved intelligently.

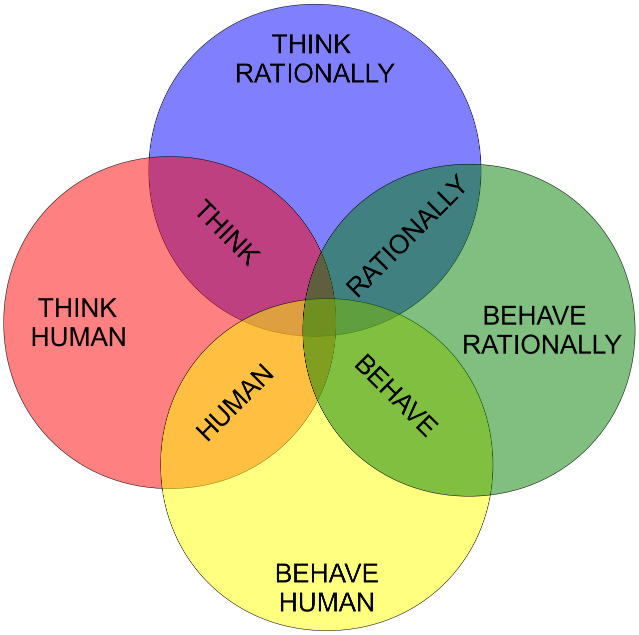


Figure 3.1: Categories of AI Systems

A definition, which is particularly relevant to this thesis, is “the ability of machines to do things that people would say require intelligence” [Jackson, 1985]. This quote then leads to the question of what tasks are deemed to require “intelligence”. Rich and Knight define a selection of intelligent task domains in [Rich and Knight, 1990]. These tasks are divided into three sections: mundane tasks, involving the perception of vision or natural language, formal tasks, such as chess or mathematical logic, and expert tasks such as engineering fault finding and scientific analysis. These definitions are particularly relevant for the system defined in this thesis, providing both fault finding and the modelling of the behaviour of protection engineers. The remainder of this chapter focuses upon the various techniques available for fault diagnosis, the motivations for each technique and various uses of these techniques for power systems diagnosis.

3.2 Intelligent Systems

Intelligent systems are systems which have implemented some form of AI; within the field of power system diagnosis many techniques have been utilised to implement AI. These techniques vary from expert systems which attempt to directly map an expert’s logical process, to model based systems which attempt to directly map the behaviour of the system being diagnosed. The use of these vary greatly depending on the sources of knowledge. The following sections discuss information sources and how this can effect the techniques available for diagnosis.

3.2.1 Information Sources

In order to apply intelligence to a problem, some form of prior information must be available. This can generally be divided into three main sources: experts in the field, data recorded during a systems operation and theoretical knowledge on how a system would be expected to operate. Given these sources, different techniques can be seen as more applicable in utilising the sources which are available and ensure the most complete sources are utilised effectively. The following is a summary of how these different sources may be used in different techniques:

- Expert Knowledge - Knowledge based reasoning (KBR) where knowledge is used to infer some solution to a problem. Expert knowledge is highly valuable and care must be taken both to correctly elicit knowledge from experts in the field and to correctly represent the knowledge. One of the representations often taken is the Rule-Base System (RBS) approach where expert knowledge is extracted into a set of rules.
- Historical Data - Historical data can be divided into two categories. The first category involves data which has little or no classification, i.e. data that can be mined for patterns. The second category involves data which has already been classified into sets of either normal or anomalous patterns or behaviours, i.e. a set of cases which can potentially be compared to the observations of a particular problem. Where classified cases are available, case-based reasoning could potentially be applied. Case-based reasoning is an approach which utilises the information from other similar problems in an attempt to solve a current problem. Therefore cases must be both classified and suitable for application to another problem. Historical data techniques are often combined with expert knowledge to provide more detailed classifications for anomalous patterns or behaviours.
- System Knowledge - Model based reasoning (MBR) involves the utilisation of the theoretical knowledge of how a device or system being analysed operates or fails. The models can then be used to emulate normal or abnormal behaviours and compared with the actual behaviour or any potential abnormal behaviour. As with case-based reasoning, models can be combined with expert knowledge to provide a further level of classification when discrepancies are discovered.

Each type of source requires some form of transferal into useful information, whether this be meeting with experts or mining data for patterns. Some of the steps which are able to be taken for each source are now detailed.

Expert Knowledge

Expert knowledge is one of the most important sources of information available when attempting to build intelligent systems. Experts are often able to combine their theoretical knowledge with their practical experience providing a full picture of the problem to be solved. The elicitation of information from experts in the field can often be a lengthy process involving a number of interviews before information is rich enough to be of technical use. The difficulties involved have led to the development of knowledge engineering methodologies, such as CommonKADS [Schrieber et al., 1999]. Such methodologies aim to provide a set of procedures or guidelines to follow during initial knowledge elicitation in an attempt to structure the elicitation process, the representation of gathered information and the required manipulation of the discovered knowledge.

Historical Data

Where unclassified historical data exists, a further level of analysis is required before the useful information can be represented for use in diagnosis. A common technique for analysing historical data is data mining. Data mining is generally implemented when a field expert is unavailable or unable to fully define the subject area. Data mining involves the analysis of historical data in an attempt to uncover patterns or behaviours hidden within the data. Having distinguished that certain behaviours or patterns are anomalous these can then be used to provide cases of anomalous behaviour. When data mining cannot be combined with a degree of expert knowledge it is mainly a flag mechanism, flagging anomalous behaviour. Therefore even in cases where an expert is not available, high level classifications can be made based on the occurrence of anomalous patterns within data.

System Knowledge

Knowledge of the operating behaviour of equipment/situations provides a means to create a functional model of a system to be diagnosed. As with other forms of knowledge the theoretical behaviours of a system can be complimented or compliment the knowledge available from experts and historical data. Theoretical behaviours can be either ideal system behaviours or abnormal system behaviours. These can be distinguished by which ideal behaviours provide a means to identify a discrepancy and abnormal behaviours allow for the comparison with these behaviours to identify a particular failure mode.

3.3 Intelligent Techniques Applied To Diagnostics

Given the three types of knowledge available to form a diagnostic system, a number of techniques have been developed in an attempt to harness the knowledge and utilise the theory of artificial intelligence to provide some form of automated diagnosis. This section discusses a range of techniques which have been applied within post-fault diagnosis.

3.3.1 Rules - Rule Based Systems

Rules can be used to represent knowledge by creating hypotheses based on new data conforming to a set of rules (i.e. IF ... THEN ...). Rule based systems are generally developed as an extraction of human knowledge into a set of heuristic rules, which has led to the term Rule Based Expert Systems (RBES) for heuristic rule based systems. Once rules are formulated there are a number of engines available which help to execute rules in an efficient manner. Many of these are domain independent and domain configurable, meaning that they can be used for generalised rule processing as well as specialised processing. Two of the most common rule engines are JESS (Java Expert System Shell) and Drools. This type of rule engine provides developers with a platform upon which to inject knowledge into a system via a sequence of rules. Different rule engines provide different formats for writing rules as well as different solvers. In general a rule engine takes in a sequence of rules and from this determines some outcome. Both of the rule engines discussed in this section use the Rete algorithm for pattern matching. The Rete algorithm [Forgy, 1982] is a popular and efficient pattern matching algorithm, which separates problems in to two types, one pattern and multi-pattern where;

- One-Pattern: One-Pattern is where a problem has a single matching pattern or path to reach a rule execution.
- Multi-Pattern: Multi-Pattern is where a problem has multiple pattern matches, meaning that within the rule base there are multiple patterns or path which correspond to the execution of a single rule.

Figures 3.2 & 3.3 present example Rete trees which are a graphical representation of the patterns produced by different sets of rules. The first example, shown in Figure 3.2, presents a one-pattern network where a single path is present between the insertion of a fact or object and the execution of the rule. The corresponding rule for this example is:

```
rule "Did Main Protection 1 operate?"
  when
    device: Device(type == "Main Protection 1", operated=true)
  then
    do-something
end
```

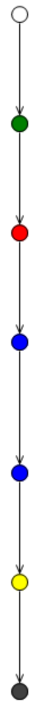


Figure 3.2: Graphical Rete Tree presenting a one-pattern rule-base

The second example, shown in Figure 3.3, presents a multi-pattern network where multiple paths are present between the insertion of a fact or object and the execution of the rule. The corresponding rule for this example is:

```
rule "Did Trip Relay Operate within 10ms of Main Protection?"
when
    deviceMP: Device(type == "Main Protection")
    deviceTR: Device(type == "Trip Relay")
    eval((deviceTR.getOpTime()-deviceMP.getOpTime())<=10)
  then
    do-something
end
```

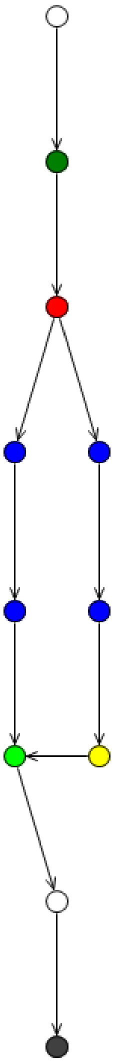


Figure 3.3: Graphical Rete Tree presenting a more multi-pattern rule-base

As shown, Rete trees are colour coded with each coloured node representing some action or meaning. These nodes can be broken down as follows:

- White - “Rete” or “Root” node, which represents the first entry point for a new fact/object.
- Dark Green - “Default Entry” node, which represents the entry point for all facts/objects.
- Red - “ObjectType” node, which represents each single object type which can be filtered on. (i.e. for multi-pattern example two “Device” objects can still be presented in a single red node.)
- Blue - “Alpha” node, which represents the filtering of an object. (i.e. for one-pattern example does “Device” have type “Main Protection 1”, then does “Device” have operated set to true)
- Yellow - “Adapter” or “Transformation” node, which represents the modification of any

facts/objects into a suitable state to be used by a "join" node (only used for left entry where a collection of objects can be expected - N.B can be collection of one) or the "then.." action in the rule (where a collection of objects may be required for some action).

- Bright Green - "Beta" node (only used in multi-patterns), which represent "join" points for facts/objects, where they both belong to a single rule. (i.e. for multi-pattern example where the two "Device" objects are both identified as needed for the same terminal node/rule.)
- Black - "Terminal" node, once a fact/object has reached this point the rule has met the "when" conditions and therefore will be executed.

The Rete algorithm is based upon the philosophy of "use more memory, find a solution faster". Unfortunately this means that the algorithm can be extremely demanding of memory resources. Over time this problem has become less pronounced, with the price of memory reducing and the advancements of the algorithm to inhibit less demanding behaviour contributing to its position as one of the most popular algorithms used within rule based systems. For further information regarding the Rete algorithm the following publications are advised [Albert, 1989, Forgy, 1982]. Although the Rete algorithm is a popular basis for many rule-engines, it should be noted that a number of other techniques, such as Prolog [Bratko, 2001], do exist to aid in the execution of rules.

Within power system fault diagnosis many systems exist which utilise a rule-based approach, both using heuristic classifiers [Tomsovic et al., 1987, Ma et al., 1992, Marathe et al., 1991] and taking into account theoretical system knowledge [Brunner et al., 1993, Pfau-Wagenbauer and Nejdl, 1992b]. The designers of CRAFT [Tomsovic et al., 1987, Ma et al., 1992, Marathe et al., 1991], which carries out diagnosis for determining the location of faults, discuss the reasoning behind taking a heuristic approach as being based on the post-fault analysis process by control room staff. Other examples such as in Davidson et. al [Davidson et al., 2003], discuss the behaviour of engineers being based on a comparison of the expected and observed behaviour and therefore more naturally based on a model-based approach.

Applications of Rule-Based Systems in Power System Fault Diagnosis

Rule-based systems, in particular RBES have been utilised for post-fault diagnosis within power systems. The alarm processing system reported in [Brunner et al., 1993, Pfau-Wagenbauer and Nejdl, 1992b] utilises a hybrid approach, which has low-level device models in the form of rules, with a hierarchical structure mapping from low-level, e.g. main protection operated, to high-level, e.g. cable fault. The system consists of over 190 rules and aims to detect both network faults and protection malfunctions. The network under observation is the distribution network consisting solely of main protection, backup protection, circuit breakers and earth fault indicators. The system has been built as a hybrid approach to reduce the time overheads associated with model-based systems.

The Telemetry Processor, part of PEDA (Protection Engineering Diagnostic Agents) [Hossack et al., 2002] is another alarm processing system which aimed to provide both high level event information and detect the maloperation of protection devices. This system aims to reduce maintenance overheads by providing network topology estimation, rather than a hard coded network topology. The system estimates which alarms are associated to one another by analysing the content of SCADA alarms (i.e. substation name, time stamp) and from this builds a picture of the circuit currently experiencing problems and therefore allows for the operation of protection devices to be determined and analysed.

CRAFT (Customer Restoration And Fault Testing) [Tomsovic et al., 1987, Ma et al., 1992, Marathe et al., 1991] is another rule-based expert system within power system fault diagnosis. This system has been developed with a different focus than those previously, in that its key aim is in fault localisation to aid in the isolation of faults and therefore the restoration of as many customers as possible. The system consists of over 500 rules, which collectively aim to provide engineers with advice on what actions to take during faults to restore the network appropriately and as quickly as possible.

SPARSE [Vale and Machado e Moura, 1993] carries out alarm synthesis, by which it processes alarms to extract useful information that can be presented to control center operators in a timely manner especially during faults when the volumes of alarms being received can be overwhelming. Utilising an expert system approach with over 120 rules, incoming alarms are grouped appropriately based on their location, timing and the activity they relate to. For example a group of breaker alarms could be combined to a conclusion of the opening of line x, or a sequence of voltage level alarms combined into a single list of discrete voltage levels. This system aims to reduce the alarms reported to engineers by combining alarms into fewer more detailed messages, which contain all of the relevant information for network control.

APEX and RESPONDD [McDonald et al., 1992] are two knowledge based systems combined to carry out alarm processing and fault diagnosis. APEX carries out alarm processing to summarise the events which are being reported by SCADA alarms. Utilising a database of network topology and of rules, the system is able to reduce the amount of raw data received by engineers, giving a summary of any activity occuring on the network. RESPONDD utilises SCADA alarms which contain information regarding voltage measurements to enhance the diagnosis of faults to provide the position, type and status (permanent/temporary) of the fault, therefore providing users with both an event summary and fault information.

Remaining Challenges

Whilst these systems use a rule-based approach for their diagnosis there remain many challenges associated with their application. Although fast execution times are often classed as a major advantage for RBS, with the low expense of memory and processing power this is becoming less of a clear advantage over other techniques. Other issues involve RBES where heuristic classifications are used and can introduce diagnostic errors. These errors include the inability to diagnose new or previously unseen faults and the potential for misdiagnosis when both new or multiple faults are observed. Within [Beschta et al., 1993] this effect is classified as not degrading gracefully; RBS are not aware of the boundaries of their knowledge, and can therefore misdiagnose new or unknown faults, unaware of where knowledge may be being misapplied. Another issue associated with rule-based systems is discussed in [Ma et al., 1992], with one of the major issues being associated with system maintenance and testing. During operational experience the authors found that maintaining a consistent and accurate system during updates and modification to be a complex task. The deficiencies associated with these systems are overcome by the system discussed in this thesis, namely:

- Model-based approach removes risk of heuristic misclassification
- Abstraction of models from diagnoses reduces complexities carried out during updates to data formats or models
- Storage of network topology within an easily maintained database enables the system to update as network configurations are changed and new circuits and equipment added.

3.3.2 Artificial Neural Networks

Artificial neural networks (ANN) consist of a series of interconnected nodes which are based upon the structure of biological neurons. Neurons (or nodes) are parametrised non-linear functions which produce a single output. The outputs of one node can then be used as a parameter for another node within the network. A graphical representation of this can be seen in Figure 3.4.

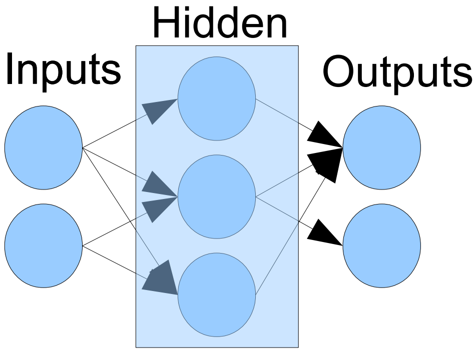


Figure 3.4: Neurons interconnected to represent the steps from input values to outputs

ANN can be trained to respond to input variables based on specific thresholds and weightings. The system is then able to make decisions based on what inputs it receives and the decision making thresholds and weightings it has been taught. To achieve the best results ANNs are trained on large volumes of training data, so that their decisions are able to be based on a large collection of previous observations, allowing the ANN to learn from experience. It is used as a means of discovering hidden information or for completing feature extraction on large volumes of historical data. Once formatted appropriately, data can be used for ANN training as discussed in [Wang et al., 1998] for transformer diagnosis. However in the area of post-fault diagnosis for power system protection schemes, a repository of information is available and is already well understood, therefore there are few examples of the use of ANNs for this type of diagnosis. For more information regarding ANN and their applications the following are advised [Mitchell, 1997, Dreyfus, 2005].

Applications of ANN in Power System Post-Fault Analysis

Within [Fritzen et al., 2010] a hybrid ANN and GA approach has been taken to carry out alarm-based protection validation. The ANN approach allows the identification of the type of protection which has operated, for example for transformer protection, two protection types are given as "Transformer Selective Protection" and "Transformer Non Selective Protection". Where a fault is identified within a transformer the ANN would then, having been trained on 126 transformer protection operation cases, identify the type of protection to have operated. The ANN is trained based on the expected collection of alarms and operations which can be attributed to each protection type. The GA utilises a knowledge base of 108 alarms and the associated 132 events they could be attributed to. The GA is then able to identify the events which have occurred based on the arrival of alarms during faults. The system will identify both events which completely map to its knowledge base and where a single discrepancy has occurred, for example only a single relay or circuit breaker device can have failed, and where simultaneous events have occurred, for example where the alarms can be fully explained by two events within the knowledge base.

Within the wider community of power system fault diagnosis the use of hybrid ANN and expert system approaches have been demonstrated. Within [Fukuyama and Ueki, 1991] an ANN for waveform recognition is combined with an expert system for fault section estimation within power systems. The system analyses DFR-type data with three phase voltage and current samples to diagnose particular fault types, such as phase to ground and three phase faults. The ANN were trained to recognise specific patterns associated with different fault types from three phase voltage and current measurements. After fault type identification has occurred expert knowledge is utilised to perform a verification layer for the ANNs. This section uses knowledge of fault profiles to verify a recognition is accurate, i.e. single phase to ground - voltage level decreases on single phase, current level increases on single phase, and voltage appears on earth. The authors discuss that the need for the ANN to be verified by an ES approach is due to the difficulties in creating a reliable ANN.

Within [Wang et al., 1998] the diagnostic problem is based on the diagnosis of problems within transformers. This system separates the diagnostic process into sections with a combination of the results being presented to users. Each of the ANNs were trained on 150 samples of data, which allowed each ANN to diagnose a single fault type. The results from this system found that issues with training data, from inconsistent sampling, was leading to a reduced accuracy. The authors discussed that where faults exist, as yet unknown to human experts, the ANN is able to provide further diagnostic information and continue to train when new information is available. However by utilising both human expertise and trained ANN, their system can provide better results than either alone.

Remaining Challenges

Within the field of post-fault analysis for power systems, ANNs are utilised where more complex data requires analysis, or where expert knowledge is unavailable or sparse. As discussed in [Fukuyama and Ueki, 1991], without some level of expert knowledge, it can be difficult to achieve reliable diagnosis from ANNs. Although ANNs are adept at handling noisy, incomplete or new fault data, it is challenging to ensure that they do not become "over trained" where fault profiles become too precise, meaning that the identification of "general characteristics" is reduced and the misclassification of new faults can occur. With the complexities in constructing a reliable ANN and issues associated with understanding the rationale of the ANNs results, ANN systems often fail to gain the user confidence required for an industrial system. The deficiencies associated with these systems are overcome by the system discussed in this thesis, namely:

- GDE approach allows the use of first principle knowledge and removes the need to train a system on large volumes of data where the information required is already well established.
- Hybrid approaches between ANNs and ES risk issues with misclassification through both over training and heuristic classifier problems.

3.3.3 Modelling Inexactness

Within logical reasoning it is often important to model inexactness or account for uncertainties in data. Fuzzy logic allows associations to be provided with a probability factor. A simple example of this would be based upon the use of the word cool to describe temperature. Figure 3.5 demonstrates the difference between standard logic and fuzzy logic. This comparison of non-fuzzy and fuzzy logic demonstrates that a probability factor of up to 100% can be given to a value, thus preventing absolutes from blocking "common sense". In this example the correct operation of a device is either statically "correct" between 0 and 20ms within a non-fuzzy environment but can be judged as having a percentage of probability for its "correct" operation between 0 and 20ms with the "perfect" behaviour lying at 10ms.

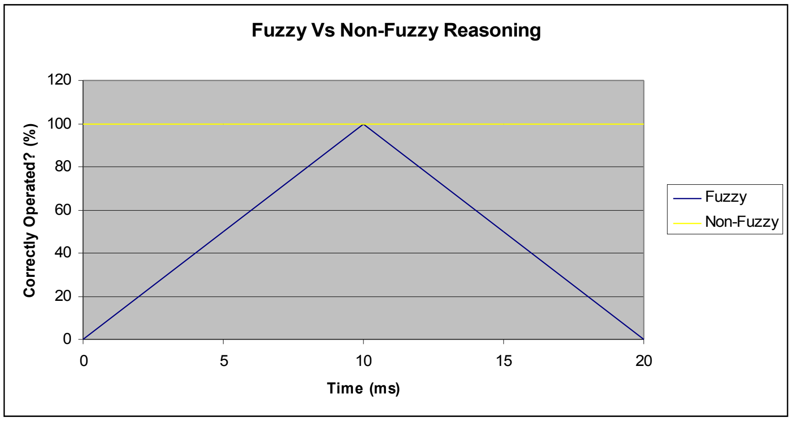


Figure 3.5: Comparison of Non-Fuzzy and Fuzzy Logic

Fuzzy logic is utilised where degrees of certainty are required for an accurate diagnosis. This is of particular use where there are levels of association to a set of correct or abnormal behaviours. However in the case of SCADA based analysis the use of fuzzy sets over "normal" sets for membership of correct or abnormal behaviour groups is unlikely to add value. For further information regarding the modelling of inexactness the following are advised [Zadeh et al., 1996, Kosko and Isaka, 1993, Rich and Knight, 1990, Luger, 2005, Pawlak, 1985].

Applications of Fuzzy Reasoning within Power System Post-Fault Analysis

Within [Luo and Kezunovic, 2008] fuzzy reasoning is used to compliment petri-nets, another intelligent technique, to allow for a certainty to be given to specific diagnoses. The fuzzy logic allows the transitions within a petri-net, from data received to fault location, to be given a constantly updating "truth factor" where the actual observation of an event would have a higher truth factor than an implied operation. For example an observed protection operation would have a high truth factor compared to the implication that where a trip relay has operated a protection operation must have operated, but it is not an observed fact. The result is that once a fault location has been determined it will have a resulting truth factor taking into account all of the available information.

Another example of fuzzy reasoning within power system fault diagnosis is discussed within [Monsef et al., 1997]. This system uses fuzzy set associations to provide a truth factor based on the likelihood of different problematic scenarios for example missing alarms, where devices did operate, or where the belief is that the device did not operate but an alarm was incorrectly received. This system is used to locate faults and validate whether relays and circuit breakers operated during faults.

Remaining Challenges

Although fuzzy reasoning allows diagnoses to be given a certainty, the value of this when making decisions on the state of the power network is not necessarily great. Although the certainty may allow some observability of how the diagnosis has been reached, engineers will still have to make decisions assuming the diagnosis is correct, regardless of its certainty factor. The deficiencies associated with these systems are overcome by the system discussed in this thesis, namely:

- Fuzziness associated with discrete events such as SCADA data adds an extra level of complexity, the GDE approach is able to provide a truth factor based on the theory of minimal diagnosis. Therefore creating a simpler version of diagnostic ranking.

3.3.4 Finite State Machines

Finite State Machines (FSM), a type of finite automata, are models of discrete behaviours where a set of states and transitions between them are defined. The transitions between states are generally assigned a condition, which when true permits the machine to change state.

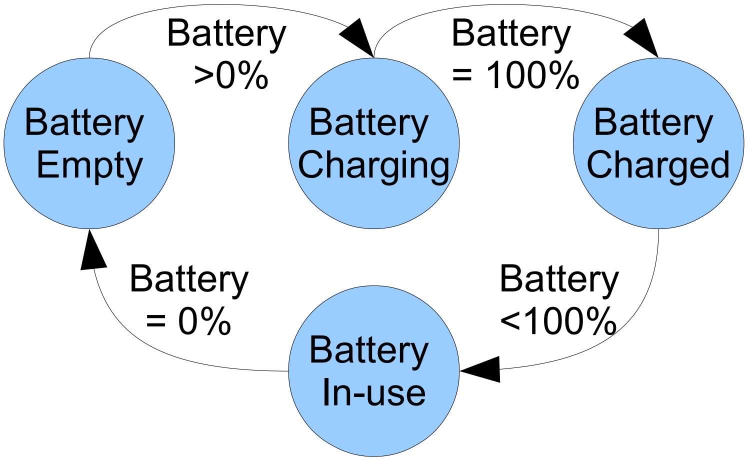


Figure 3.6: Finite State Machine - Representing the states and transitions a typical battery may take.

Finite-state machines have been utilised in the diagnosis of discrete-event systems, where a discrete model is produced to represent a generalised active system to be diagnosed. A diagnosis is then able to be identified when a device enters a specific "non-normal" state. Within the area of post-fault diagnosis in power systems there are examples of FSM [Chantler et al., 2000] where augmented reactive models (ARM) are utilised. An ARM is an extension of reactive model (RM) which can take into account generic conditions such as if a specific variable reaches a specific threshold value, an ARM allows additional time constraints to also be taken into account. Therefore models can be configured to transition not only on specific values but if an impedance or current were to reach a specific limit or threshold. Not only can a limit/threshold be set based on the magnitude of a variable but additionally a time restraint can be considered. For example if the impedance or current reach the limit within a specific time period, then the transition shall occur. This is demonstrated in Figure 3.7 where a transition occurs from "Distance Protection OFF Correct" when both impedance Z reaches a specified value of Z_M and the time delay is within the ranges defined, for example where Z is $< Z_M$ for longer than X ms but less than Y ms the model will transist to "Distance Protection ON Correct", returning to "Distance Protection OFF Correct" when the impedance Z returns to a normal level.

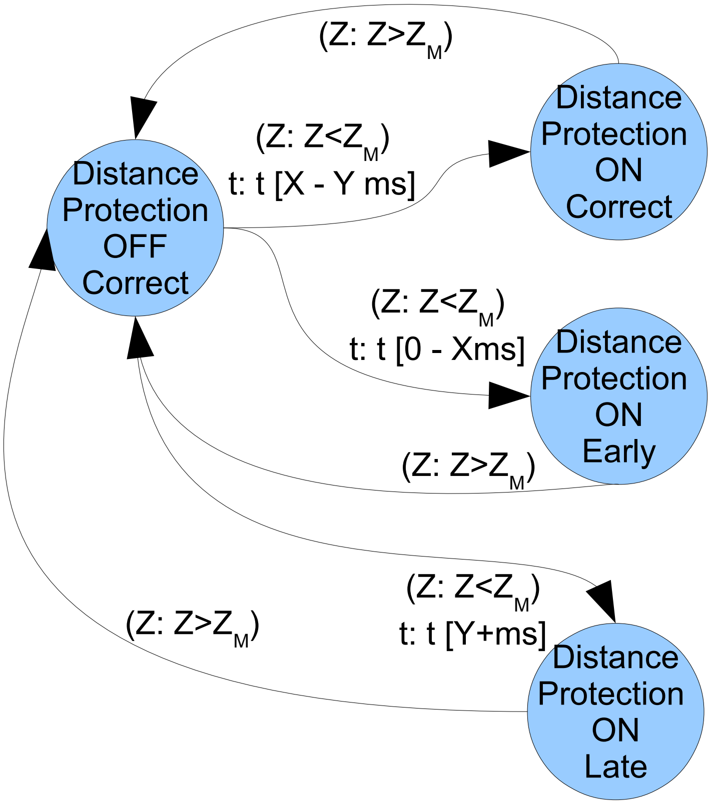

(Z: Z>Z_M)
Distance Protection ON Correct
(Z: Z<Z_M)
t: t [X - Y ms]
Distance Protection OFF Correct
(Z: Z<Z_M)
t: t [0 - Xms]
Distance Protection ON Early
(Z: Z>Z_M)
(Z: Z<Z_M)
t: t [Y+ms]
(Z: Z>Z_M)
Distance Protection ON Late

Figure 3.7: Augmented Reactive Model

For more information regarding FSM, automata and their applications the following book is advised [Hopcroft et al., 2007].

Applications of FSM in Power System Post-Fault Analysis

Applications of FSM within power system post-fault analysis includes the use of ARM within [Chantler et al., 2000]. This system utilises an extended form of FSM to analyse DFR-type data for protection validation and the determination of fault type, e.g. Phase x to Ground. The system observes voltage and current levels and determines whether protection devices intervened correctly, did not intervene when they should have detected a fault or in the case of distance protection devices whether they operated in a timely manner. ARMs are used to model both ideal and fault behaviours for distance protection devices, meaning that they model when a distance protection device should have operated, i.e. impedance level and time delay have reached settings, or would have operated incorrectly i.e. where an early or late operation occurred based on timings or impedance measurements. Other examples of the use of FSM within power system analysis are within the field of discrete event systems, [Kwong and Yonge-Mallo, 2011, Jiang et al., 2003], where the use of FSM is taken from a diagnostic approach rather than attempting to solve the specific problems found within power systems analysis. Although these papers demonstrate a system which utilises FSM for fault diagnosis and in [Kwong and Yonge-Mallo, 2011] the application to power systems, they are unable to demonstrate how this would operate in a real power system.

Remaining Challenges

The use of the specialised FSM, such as ARM, as demonstrated is of particular value when analysing the operation of distance protection where not only are impedance values required for an operation but also time delays. However for the level of analysis required for SCADA fault diagnosis each individual device i.e. main protections, trip relays etc. have only two states - ON or OFF, and therefore the use of FSM at device level would be unnecessarily complex. Unfortunately in terms of modelling the protection system as a whole from main protections to circuit breakers, a model with over 1000 state transitions would be required (potential scheme size of 10 devices), and with the creation of different scheme set-ups adding an additional complexity with hundreds of different scheme setting possibilities, the modelling of a complete protection system becomes a major challenge. Another problem with FSM is related to the non-chronological arrival of alarm data, meaning that attempting to match all potential transitions within the machine would have to preempt the occurrence of a delayed or missing alarm to prevent the machine transitioning into an erroneous state. The deficiencies associated with these systems are overcome by the system discussed in this thesis, namely:

- GDE enables the modelling of entire protection schemes to be made up of a collection of models reducing the need to account for thousands of state transisitions to be modelled for small changes in network topology.
- GDE allows the insertion of data independently of arrival time, the non-chronological order of alarms will not lead to incorrect transitions within the system model.

3.3.5 Petri-nets

Petri-nets are a graphical technique used to represent discrete events. This technique is based upon the idea of movement through a decision tree to move from raw data to high level hypotheses. Petri-nets are focused around the idea of places and transitions.

- Places: Places represent states, which are either start states or states which have been reached following transitions. The last state or place within a petri-net would provide some form of high level hypotheses.
- Transitions: Transitions are logical operations within the petri-net which are only fired if all of its inputs have “tokens”, meaning that they are of a discrete value greater than 1. On a diagram these token values are represented by internal dots for places.

An example of a simple petri-net is shown in Figure 3.8. This example demonstrates that a transition to "Feeder 1 Fault" will occur if TR1, CB1 and CB2 all operate, and to "Feeder 2 Fault" where TR2, CB2 and CB3 operate. The dots within the 3 states arced to the transition are tokens, which identify that for this example they are all "true" or have a discrete value above 1, and "Feeder 2 Fault" is the diagnosis. For a full example all potential paths through to different faults and potential maloperations would be mapped. Thus allowing all operations and non-operations to be visualised and therefore enhance understanding during validation.

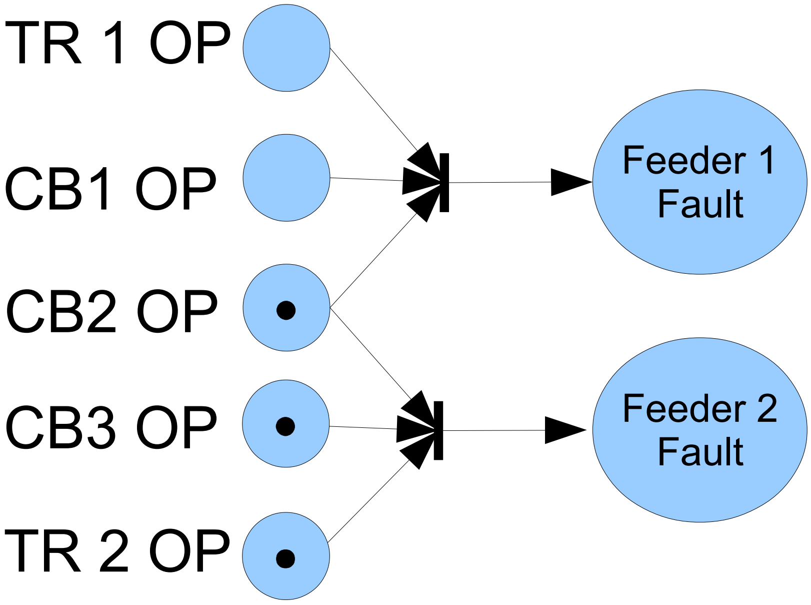


Figure 3.8: Petri-net Example

For more information regarding petri-nets and their applications the following is advised [David and Alla, 2005].

Applications of Petri-nets in Power System Post-Fault Analysis

Petri-nets have been utilised in a number of systems within the field of power system post-fault analysis. The system discussed in [Luo and Kezunovic, 2008] uses fuzzy reasoning petri-nets to carry out fault section estimation, as discussed within the fuzzy reasoning section.

Further work was carried out utilising petri-nets within [Biswas et al., 2004] where the system is modelled with petri-nets to diagnose whether the main or backup protection schemes operated during the detection of a fault. The system has modelled the expected behaviours when the main protection scheme has operated correctly or where the backup protection schemes have operated correctly. Therefore the system can only observe which protection scheme has failed to operate and does not handle missing alarms or timing issues associated with the protection scheme. The system also assumes that all faults are transient and will be cleared, however makes no assessment of the operation of reclosers.

Within [Zhanjun et al., 2012] the system utilised a petri-net approach which at run-time dynamically builds a petri-net model for each protection scheme associated with a fault in progress. Operations of protection schemes for the clearing of each bus/line are each given a priority, therefore once the petri-net has confirmed that the fault lies within the specific bus/line the correct operation of protection devices is validated based on these priorities. Priorities are based on the location to the faulted section, if the faulted section is deemed to be closest to a specific set of components they are given the highest priority, however the fault could be cleared by further out components on their failure, assigning them a lower priority. For example, protection schemes assigned a higher priority should have responded to any faults on the bus/line, if they did not operate, they are deemed to have not operated correctly. Inversely if a lower priority scheme has operated alongside a higher priority scheme, they are deemed to have not operated correctly as the higher priority scheme had already operated, therefore carrying out high level protection validation.

Remaining Challenges

Petri-nets, although less common than RBES, are becoming more common with more than five thousand examples in the last decade for fault diagnosis. Kezunovic et. al [Luo and Kezunovic, 2008] discuss the benefits of petri-nets in terms of the ability to process information in parallel but comment on the difficulties involved in modelling the inexactness or uncertainties with the protection system; as discussed before their solution is to utilise fuzzy reasoning. Although a number of examples exist within fault diagnosis [Biswas et al., 2004, Zhanjun et al., 2012], a major issue associated with petri-nets, similarly to expert systems, is the difficulty in diagnosing multiple-faults and the issues involved in utilising heuristic classifiers. Petri-nets, similarly to FSM, can also be liable to problems involved with the non-chronological arrival of alarms and the complexity of applying themselves to multiple protection scheme configurations. The deficiencies associated with these systems are overcome by the system discussed in this thesis, namely:

- Similarly to the issues associated with FSM:
  - GDE enables the modelling of entire protection schemes to be made up of a collection of models reducing the need to account for thousands of state transisitions to be modelled for small changes in network topology.
  - GDE allows the insertion of data independently of arrival time, the non-chronological order of alarms will not lead to incorrect transitions within the system model.
- GDE enables the diagnosis of multiple faults, something which is highly complex to implement within petri-nets.

3.3.6 Model-Based Reasoning (First Principles)

To create a MBR system an in-depth knowledge of the device or system which is to be modelled must be acquired. A model based system must model some form of device however it can be developed to handle ideal device operation or device operation during specific fault conditions. Modelling from first principles means that the models produced are based on the theoretical behaviour of a device.

The theoretical behaviour of a device can be divided into two types, the ideal or normal expected behaviours and the abnormal or expected behaviours during faults. Figure 3.9 presents how these behaviours can be compared to "known" behaviours. This comparison provides the means to identify discrepancies which can be used to identify a fault has occurred or where fault behaviours are available, identify a matching fault signature.

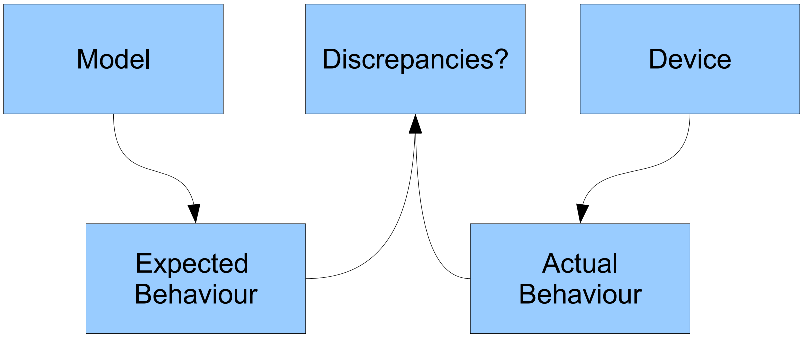


Figure 3.9: Diagnosing Abnormal Behaviour

When working with ideal behaviour, models must be based upon the desired operation of a device. This behaviour is then compared to the observable behaviour of the actual device to determine any discrepancies and therefore deviations from "normal" operation.

An example of this can be demonstrated using a Multiplier and Adder circuit, as seen in Figure 3.10. Where consistent results are observed at points X, Y, Z, F and G, the circuit can be identified as operating correctly. Where discrepancies occur, for example in Figure 3.11, inconsistencies are able to be identified, in this case at point X, potential fault candidates are able to be identified. With the discrepancy identifiable as occurring at point X, the fault can be diagnosed as within "MULTI-1" i.e. the problem is only observed at the output of "MULTI-1".

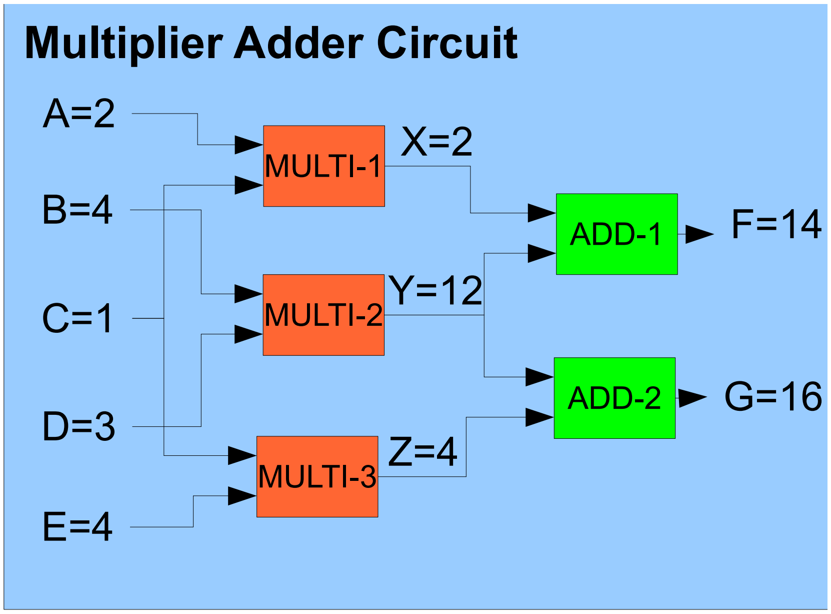


Figure 3.10: Ideal Behaviour Models - Multiplier Adder Circuit - Example taken from [De Kleer and Williams, 1987]

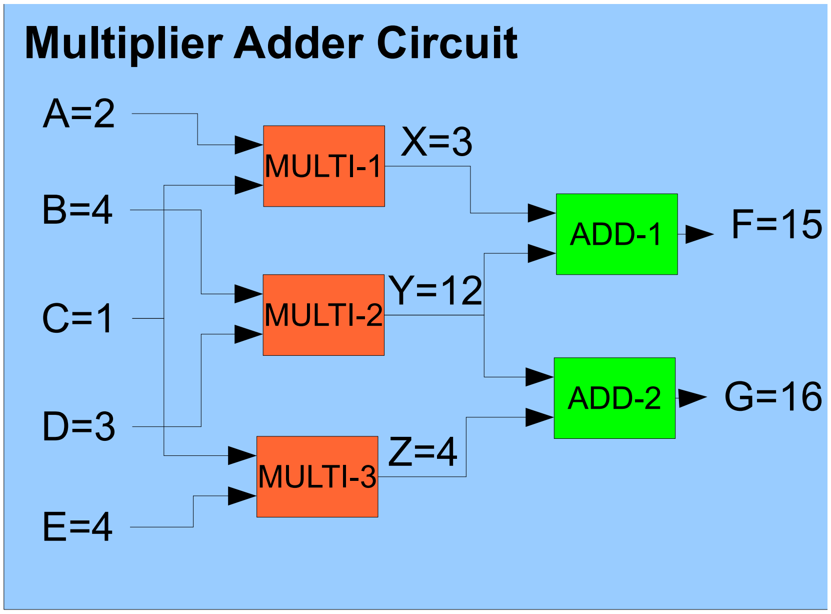


Figure 3.11: Ideal Behaviour Models - Multiplier Adder Circuit with Fault - Example taken from [De Kleer and Williams, 1987]

When utilising fault models, the behaviour of a system would need to be compared with both ideal and fault behaviours to identify when discrepancies occur, and when they occur whether they match the modelled fault behaviours. A simple example of this is the swapping of “MULTI-1” to “MULTI-1 (+1 Fault)”. Figure 3.12 presents the addition of this fault model, where the observed value of X=3 can be identified as matching the fault profile of “+1” in the “MULTI-1” fault model.

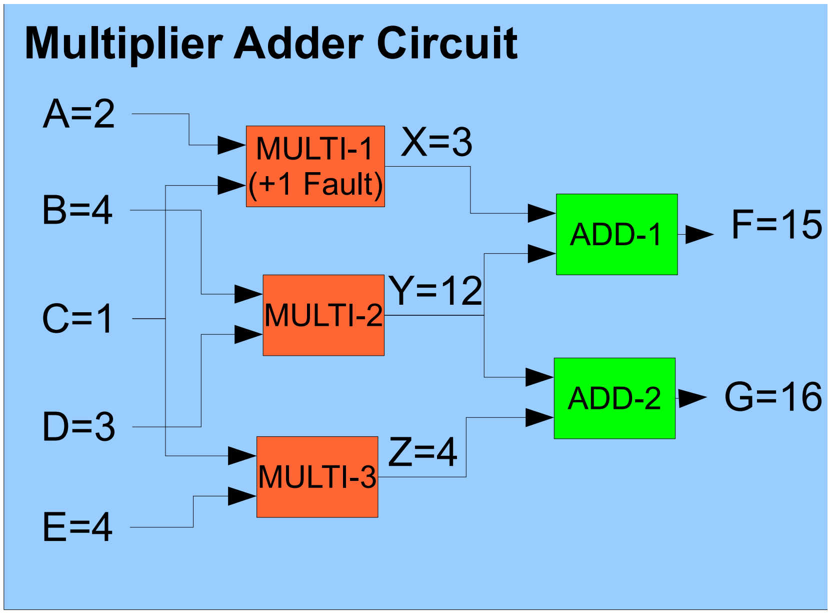


Figure 3.12: Fault Behaviour Models - Multiplier Adder Circuit with Fault

The addition of fault models can identify not only the occurrence of faults, but also the type of fault being observed. However, the building of fault models is often a time consuming task, with the ability to provide an exhaustive collection of potential fault models being extremely challenging. Fault models, although useful to deepen the level of diagnosis available within model-based techniques, are not necessary to provide a valuable model-based diagnosis system.

Application of MBR to Power System Post-Fault Analysis

The DSS (Decision Support System) described within [McArthur et al., 1996, Bell et al., 1998] is made up of two knowledge based systems and one model based. APEX and RESPONDD [McDonald et al., 1992], the two knowledge based systems operate to carry out alarm processing and fault diagnosis and are described previously within rule-based systems. The DSS makes up the third part of this system and carries out protection validation by means of a model-based approach. This system utilises the GDE and both SCADA operation alarms, e.g. circuit breaker open and the current values observed at a CT. The modelling of devices from CTs through to circuit breakers allows the behaviours observed from SCADA to be validated against their expected behaviours. The use of current measurements allows for the validations of not only protection devices responses to each other but also verify the behaviour during the detection of and reaction to faults.

Within [Malheiro et al., 2005] a model-based approach is utilised for fault detection and validation of protection scheme operation. The arrival of alarms triggers the system to produce a set of expected behaviours, from the components which should be affected by the arrival of the alarm, which can then be identified as having operated or malfunctioned. As a means of handling incomplete information, for example missing alarms, some modifications have been made, for example allowing the correct operation of a circuit breaker to be identified by the occurrence of trip operation or circuit breaker open alarms. Therefore should one of these alarms be missing no conflict will be produced; only the non-reception of both alarms would result in a conflict notification.

DPNet [Beschta et al., 1993] is another example of a model based reasoning system within the power industry. DPNet uses the GDE method of model-based diagnosis to determine fault location on the network and is able to identify the false intervention of distance protection devices based on additional information during faults regarding their distance to each end of the line.

PEDA (Protection Engineering Diagnostic Agents) [McArthur et al., 2003, Hossack et al., 2003a, Davidson et al., 2005] is a multi-agent approach to diagnosing faults and protection maloperations on the power network, utilising both DFR and SCADA data. The SCADA based approach, the Telemetry Processor is discussed earlier within rule-based systems. This system is combined with a model-based approach on DFR data, where accurate protection validation is able to be carried out using a GDE approach and extracted topology information stored within the DFR data.

Remaining Challenges

MBR systems building upon first principle knowledge of devices have many advantages. By developing a model which is able to fall back on the theory of device behaviour allows for new faults and discrepancies to be identified. Malheiro et. al [Malheiro et al., 2005] describe an advantage of model based diagnosis as being able “to handle unexpected situations” thus enabling the diagnosis of new and unknown faults without the risk of misdiagnosis. Within Davidson et. al [Davidson et al., 2003] where protection operation is validated based on DFR data it is discussed that protection engineers in fact carry out their first pass diagnosis based on their model of the network. Therefore MBD of this type can be seen as modelling both the behaviour of the system and the diagnostic procedure of protection engineers. One of the major pitfalls associated with MBD is related to the general execution times of MBD systems with the computational overheads that they require. However, with the reduced cost of both processing power and memory this is generally no longer an issue with MBD systems being capable of real-time diagnosis. Another issue associated with MBD systems is related to the complexity associated with their models, however with care models can be designed to be both configurable and human readable. In terms of the use of model-based diagnosis, within [Beschta et al., 1993], the benefits discussed include the production of a component library where models are able to be configured for domain specific circuits and the ability for models to identify unknown symptoms and faults as they rely on first principle knowledge of the system rather than particular symptom to fault relationships. The deficiencies associated with these systems are overcome by the system discussed in this thesis, namely:

- Utilising GDE model-based diagnosis therefore gaining the ability to carry out diagnosis of multiple faults and keeping diagnostics abstracted from the system beng modelled, thus reducing the complexities on system changes and expansion.

- Storage of network topology within an easily maintained database enables the system to update as network configurations are changed and new circuits and equipment added.

3.4 Conclusion

Although the utilisation of heuristic knowledge within rule-based systems [Pfau-Wagenbauer and Nejdl, 1992b, Hossack et al., 2002, Ma et al., 1992, Vale and Machado e Moura, 1993, McDonald et al., 1992] is a common approach to the post-fault analysis problem, due to their reliance on expert knowledge over first-principle knowledge, they may suffer from an inability to diagnose previously unseen faults, multiple faults or an increased likelihood of misdiagnosis. Other approaches taken within post-fault analysis, such as FSM and petri-nets, share the problems associated with expert systems by modelling heuristic knowledge over purely system knowledge. Implementations of FSM and petri-nets [Jiang et al., 2003, Zhanjun et al., 2012] also suffer from problems when attempting to diagnose multiple faults as transitions within them would need to specifically be defined for the occurrence of multiple faults. FSM and petri-nets also suffer from added complexity when attempting to deal with missing or unchronological alarms, a common problem with SCADA data, by which extra care must be taken to prevent incorrect transitions occuring. Within [Luo and Kezunovic, 2008] fuzzy reasoning is utilised in an attempt to mitigate this problem, however the system would still suffer from problems associated with the size of models required to cover a range of protection schemes and settings. For the problem of post-fault diagnosis where both the behaviour of devices and expert knowledge are well known and of great depth, the use of historical knowledge for diagnosis would appear to be of little use. Another issue involved with the use of trained diagnostics is the complexity of applying them to different network schemes and settings. Various systems have taken a model-based approach for post-fault analysis [McArthur et al., 1996, Malheiro et al., 2005, Beschta et al., 1993, McArthur et al., 2003], and only in some of these cases [Malheiro et al., 2005] has model-based diagnosis been used on SCADA alarm data alone. These applications also have a different focus to this thesis either in being based on fault type diagnosis (i.e. phase to phase, approximation of distance) rather than protection validation, or on protection validation when further data sources are available allowing calculations such as distance to fault to be utilised (i.e. enabling the validation of distance protection operation).

The approach proposed in this thesis is unique in its use of the GDE method of model-based reasoning to carry out protection validation using operational SCADA data alone. This approach provides a means of creating a generic diagnostic process for the protection network. The adapted GDE discussed in this thesis has been developed to carry out missing alarm identification, and to utilise automatically generated and configured models. This method is ideal for the protection validation problem as it allows different configurations to be dynamically modelled and accurately diagnosed, for both unknown and multiple faults without the need for fault models.

The following chapter discusses the design decisions and implementation carried out for the model-based system, PSDiagnosis, developed for ScottishPower Energy Networks to carry out automated incident identification and protection validation on their transmission network.

Chapter 4

A Model-Based Alarm Processor for the Assessment of Protection System Performance

4.1 Introduction

This chapter introduces a system, PSDiagnosis, which carries out post-fault incident identification and diagnosis of protection operations. The requirements of this system was produced during knowledge elicitation interactions with protection engineers working for ScottishPower Energy Networks. Therefore given the information available in SCADA alarms, using utility partner data as a benchmark, and extensive interaction with protection engineers the following requirements have been identified:

- *Determine disturbance location* - identify the section of the network affected by the incident including the identification of any protection devices where a maloperation has occurred.
- *Analyse "out of order" alarms* - identify and correctly analyse alarms which may have been delayed due to communications errors.
- *Filter irrelevant alarms* - up to 90% of SCADA alarms received can be unrelated to actual disturbances on the network, such as communications synchronisations. These alarms are unnecessary for this type of assessment, and therefore should be filtered out.
- *Group alarms into incidents* - utilising information about the protection network and SCADA alarm time stamps, group alarms as they arrive into "incident" groups which can provide a snap shot of current conditions on the network.
- *Identify missing alarms* - missing alarms have been reported in a number of transmission operators. The identification of missing alarms is vital to prevent the misdiagnosis of correctly

operating devices. [Vale et al., 1999, Hor et al., 2007]

- *Validate the performance of protection equipment* - based on both the operation or non-operation of a device and temporal responses.
  - *Minimal Maintenance* - the system must be able to produce reliable results with a low level of maintenance required for updates, such as: new data formats, circuit information or temporal dependencies between devices.
  - *Universal Access* - allow engineers to access the information on both the network settings (including the links between SCADA and physical devices) and incident classifications from anywhere in a corporate intranet.

Given the identification of these requirements model-based diagnosis was identified as an ideal approach for this problem.

4.2 Technique Selection

As discussed previously many techniques have been used within post-fault diagnosis. However the requirements for this alarm processor have made the use of model-based diagnosis and in particular the utilisation and modification of the GDE. Ultimately this decision was made taking into account the knowledge available, both expert and system knowledge, and the valuable attributes available with a model-based approach, in particular those available through use of GDE.

4.2.1 Easily Configurable Models

In order to accurately identify the relationship between alarms and devices on the network this connection must be both available and updatable for any network changes. By utilising a model-based approach, models of each device on the network can be built by utilising a "standard" model which can then be configured based on the real configuration of the device. These models can then be connected based on the protection scheme at each site and create models of the protection scheme when an incident is identified as having occurred at that site. The system developed, which allows engineers to easily update network configurations, is described further in Appendix A.

4.2.2 Order of Alarms

In order to carry out analysis during an incident the technique used must be able to revise hypotheses as more alarms are received. Model based diagnosis with the GDE allows for the insertion of "alarms" or "facts" into the model at any stage. The arrival of alarms is independent from the final diagnoses. However for both FSM and petri-nets a delay would be required to prevent incorrect paths being followed. Therefore analysis would need to be delayed until the incident can be identified as completed.

4.2.3 Missing Alarms

When an alarm fails to arrive it does not necessarily mean that its associated device failed to operate, but may signify that a communications malfunction occurred. For example if a trip relay alarm fails to arrive, but the circuit breaker operates (assuming no other trip signal could have been received) the trip relay must have operated and its alarm must have gone missing. Similar to the alarm order problem, for FSM and petri-nets any missing alarm could lead to a false state or hypothesis being reached as a state change cannot occur until the alarm is received. In comparison modelling from first principles using the GDE the missing alarm problem would be eradicated as a missing alarm would not necessarily cause the device to flag a discrepancy.

4.2.4 Independent Diagnosis

Utilising the GDE as a model-based diagnostic approach allows for the independent diagnosis of systems using the theory of Occam's razor (term named after William of Occam by William Hamilton, 1852 [Thorburn, 1915]) or minimal diagnosis (the diagnosis with the simplest or fewest assumptions is more favourable). This independence allows models of many different domains to be diagnosed in a similar way and therefore partitions the diagnostic process from the models themselves. In particular the GDE approach is valuable for its inherent ability to diagnose multiple faults, key to the diagnosis of protection systems.

4.3 Introduction to the General Diagnostic Engine

The GDE as defined in [De Kleer and Williams, 1987, de Kleer, 1990] is valuable as a domain independent system which relies on the theory of minimal diagnosis rather than any domain knowledge to carry out fault diagnosis. As with other Model-Based Diagnosis approaches the system requires a behavioural model of the individual components being modelled as well as the structure of system being modelled (the connections between components). The GDE also requires a method of conflict detection to identify where a component's behaviour has deviated from normal behaviour. After conflicts have been generated the theory of diagnosis must be applied. This diagnosis [Hamscher et al., 1992] is generally divided into two characterisations:

- Consistency Based Diagnosis - where a minimal set of node assignments or fault "candidates" can explain the observed behaviour and therefore provides a minimal diagnosis.
- Abductive Based Diagnosis - where a minimal set of hypotheses can explain the observations seen.

For more information on the different approaches to diagnoses the following sources are recommended, [Poole, 1989, Knill et al., 1996].

The GDE carries out consistency based diagnosis, meaning that it utilises a device's normal behaviour to identify abnormal behaviour and therefore potential diagnoses, thus meaning that previously unseen abnormal behaviour can be diagnosed. If abductive based diagnosis had been utilised, a link would be required between faulty behaviours and diagnoses. The result being that abnormal behaviour must be associated with a particular diagnosis and therefore prior knowledge of any abnormal behaviour must be known.

The requirements and functions of the GDE are discussed in the following sections.

4.3.1 Behavioural Models

Model-based diagnosis requires some form of model to represent the system of interest. In the case of a fully observable system, such as the SCADA alarm processing problem models can be produced of each individual element in the overall system and then new facts can both be inserted and propagated at each connection between the devices.

4.3.2 Conflict Detection

A method of conflict detection is required to identify where the behaviour of a component has deviated from normal behaviour. During the operation of GDE new facts and assumptions are added to their appropriate input nodes. A conflict must be identified when a propagated assumption does not match the actual data inserted at a particular node or two propagated assumptions do not match at a particular node. Appropriate conflict detection methods are vital in order to identify the points at which the observed behaviour of the modelled system have deviated unacceptably from the expected behaviour.

4.3.3 Fault Candidate Generation

After conflicts have been generated the theory of diagnosis must be applied. The GDE carries out consistency based diagnosis exploiting the theory of diagnosis, a minimal diagnosis is defined as the minimal set of node assignments or fault “candidates” that can explain the observed behaviour. The GDE process of reducing the “candidate sets” into a minimal diagnosis utilises the theory of set covering.

The GDE minimal diagnosis strategy carries out conflict refinement into diagnosis by exploiting the following [De Kleer and Williams, 1987, Beschta et al., 1993]:

- “Exploiting Minimality” - select the minimum number of inconsistencies which could explain all inconsistencies.
- “Monotonicity of Measurements” - after a measurement, cache all predictions so that the addition is incremental to the set of predictions.
- “Monotonicity of Assumptions” - after new assumptions are built they expand only the set of assumptions they lie within.

- “Redundant Inferences” - record inferences as dependencies so that they are only executed once.
- “Exploiting the Sparseness of Search Space” - utilising the first four refinement strategies the system is able to only exploit the assumptions and behaviours of “interesting” or relevant environments.

This process can be simplified into the following steps once conflicts have been identified:

1. Remove Duplicates
2. Remove Super-sets
3. Multiply Out

4.3.4 GDE Example - Multiplier Adder Circuit

To demonstrate the standard operation of the GDE some examples utilising the multiplier adder circuit shown in Figure 4.1 are now demonstrated.

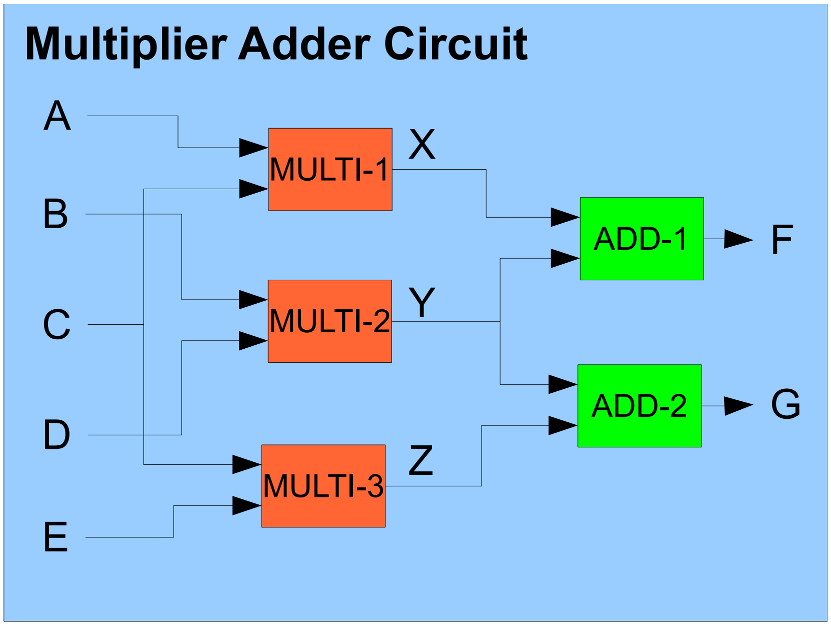


Figure 4.1: Multiplier Adder Circuit

The multiplier adder circuit [De Kleer and Williams, 1987] is commonly used to demonstrate the functions of a model based diagnosis systems. The notation used within the GDE environment can be seen in Figure 4.2. This notation, identified by A{B} indicates that where A represents the value present at the node, B represents the assumption made to infer that value. As such, empty brackets signify actual observations. Where a value appears within the brackets, it would signify “if B worked correctly, then the inferred value would be A”.

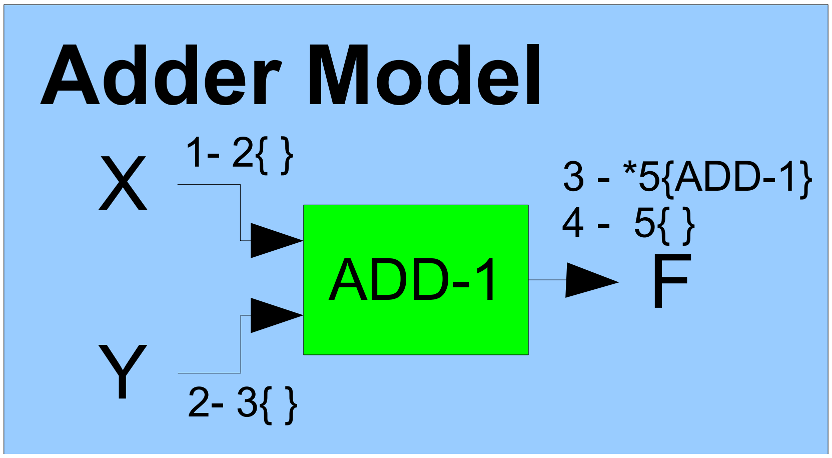


Figure 4.2: Adder Model

This figure presents the following order of events:

1. X = 2
2. Y = 3
3. F = 5

Following the arrival of both X and Y, the model is able to predict the output of F=5, this is denoted by 3 - *5{ADD-1}, meaning that having received X=2 and Y=3 and on the assumption that ADD-1 is functioning correctly, the output of F should be 5. This assumption can be seen as step 3 within Figure 4.2

The GDE approach is shown in two examples showing a functioning multiplier adder circuit and a faulty circuit.

Consistent Multiplier Adder Example

The example shown in Figure 4.3 presents an ordered list of observations and assumptions as they propagate through the multiplier adder circuit. The observations inserted into the model are: A=2, B=3, C=4, D=3, E=2, F=17, G=17.

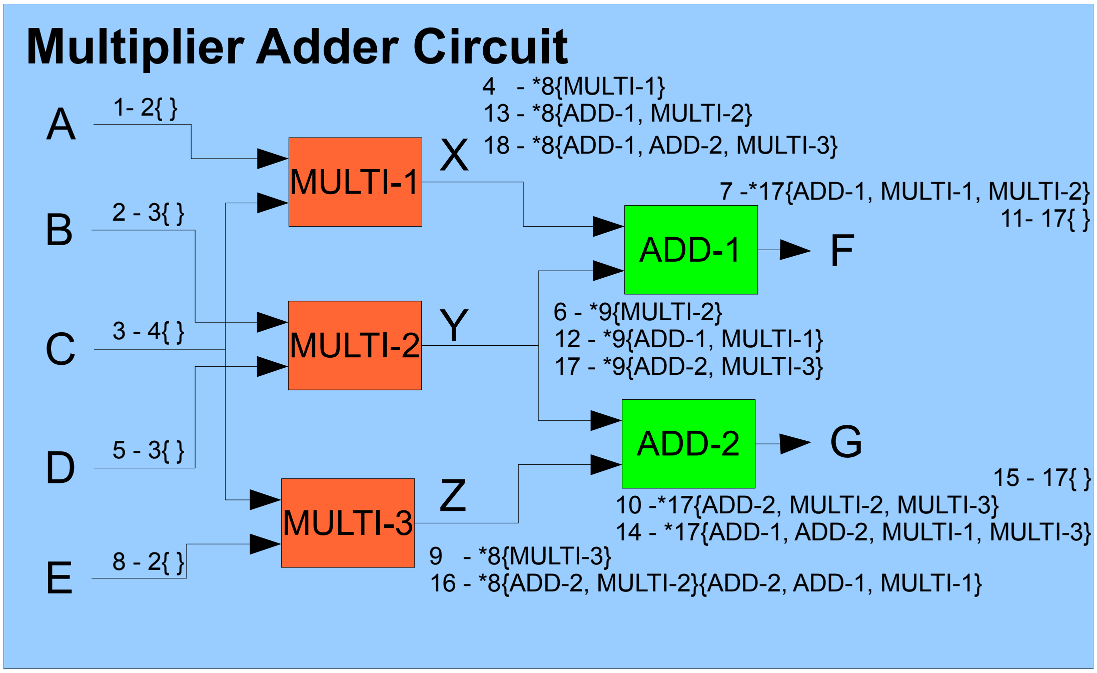


Figure 4.3: Consistent Multiplier Adder Circuit

The steps based on these observations are as follows:

1. The value 2 is given to node A, A=2. No propagations can occur as more information is required to make any assumptions.
   - No Conflicts Exist ⇒ No Diagnoses
2. The value 3 is given to node B, B=3. No propagations can occur as more information is required to make any assumptions.
   - No Conflicts Exist ⇒ No Diagnoses
3. The value 4 is given to node C, C=4, and is propagated.
   - No Conflicts Exist ⇒ No Diagnoses
4. Propagation of A and C result in assumption at point X. (2 x 4 = 8 assuming MULTI-1 is functioning correctly)
   - No Conflicts Exist ⇒ No Diagnoses
5. The value 3 is given to node D, D=3, and is propagated.
   - No Conflicts Exist ⇒ No Diagnoses
6. Propagation of B and D result in assumption at point Y. (3 x 3 = 9 assuming MULTI-2 is functioning correctly)
   - No Conflicts Exist ⇒ No Diagnoses

7. Propagation of X and Y result in assumption at point F. ((2 x 4) + (3 x 3) = 17 assuming ADD-1, MULTI-1 and MULTI-2 are functioning correctly)
   - No Conflicts Exist ⇒ No Diagnoses
8. The value 2 is given to node E, E=2, and is propagated.
   - No Conflicts Exist ⇒ No Diagnoses
9. Propagation of C and E result in assumption at point Z. (2 x 4 = 8 assuming MULTI-3 is functioning correctly)
   - No Conflicts Exist ⇒ No Diagnoses
10. Propagation of Y and Z result in assumption at point G. ((3 x 3) + (4 x 2) = 17 assuming ADD-2, MULTI-2 and MULTI-3 are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
11. The value 17 is given to node F, F=17, and is propagated.
    - No Conflicts Exist ⇒ No Diagnoses
12. Propagation of F and X result in assumption at point Y. (17 - (2 x 4) = 9 assuming ADD-1, MULTI-1 are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
13. Propagation of F and Y result in assumption at point X. (17 - (3 x 3) = 8 assuming ADD-1 and MULTI-2 are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
14. Propagation of X, F, Y and Z result in assumption at point G. ((17 - (2 x 4)) + (2 x 4) = 17 assuming ADD-1, ADD-2, MULTI-1 and MULTI-3 are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
15. The value 17 is given to node G, G=17, and is propagated.
    - No Conflicts Exist ⇒ No Diagnoses
16. Propagation of G and Y result in assumption at point Z. (17 - (3 x 3) = 8 assuming ADD-2 and MULTI-2, or ADD-2, ADD-1 and MULTI-1, are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
17. Propagation of G and Z result in assumption at point Y (17 - (2 x 4) = 9 assuming ADD-2 and MULTI-3 are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
18. Propagation of F, Y, G and Z result in assumption at point X. (17 - (17 - (2 x 4)) = 8 assuming ADD-1, ADD-2 and MULTI-3 are functioning correctly)

- No Conflicts Exist ⇒ No Diagnoses

As all observations from A to Z are consistent there are no associated diagnoses for this example and all models have operated as expected.

Faulty Multiplier Adder Example

The example shown in Figure 4.4 presents an ordered list of observations and assumptions as they propagate through the multiplier adder circuit. The observations inserted into the model are: A=2, B=3, C=4, D=5, E=2, F=17, G=19.

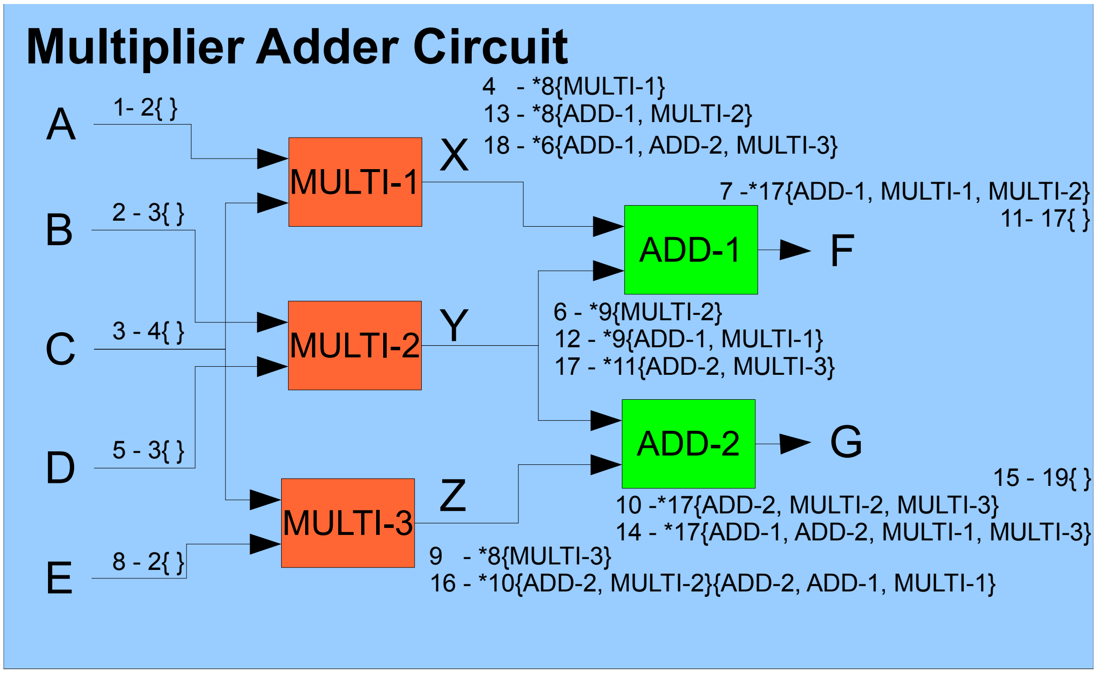


Figure 4.4: Faulty Multiplier Adder Circuit

The steps based on these observations are as follows:

1. The value 2 is given to node A, A=2. No propagations can occur as more information is required to make any assumptions.
   - No Conflicts Exist ⇒ No Diagnoses
2. The value 3 is given to node B, B=3. No propagations can occur as more information is required to make any assumptions.
   - No Conflicts Exist ⇒ No Diagnoses
3. The value 4 is given to node C, C=4, and is propagated.
   - No Conflicts Exist ⇒ No Diagnoses

4. Propagation of A and C result in assumption at point X. (2 x 4 = 8 assuming MULTI-1 is functioning correctly)
   - No Conflicts Exist ⇒ No Diagnoses
5. The value 3 is given to node D, D=3, and is propagated.
   - No Conflicts Exist ⇒ No Diagnoses
6. Propagation of B and D result in assumption at point Y. (3 x 3 = 9 assuming MULTI-2 is functioning correctly)
   - No Conflicts Exist ⇒ No Diagnoses
7. Propagation of X and Y result in assumption at point F. ((2 x 4) + (3 x 3) = 17 assuming ADD-1, MULTI-1 and MULTI-2 are functioning correctly)
   - No Conflicts Exist ⇒ No Diagnoses
8. The value 2 is given to node E, E=2, and is propagated.
   - No Conflicts Exist ⇒ No Diagnoses
9. Propagation of C and E result in assumption at point Z. (2 x 4 = 8 assuming MULTI-3 is functioning correctly)
   - No Conflicts Exist ⇒ No Diagnoses
10. Propagation of Y and Z result in assumption at point G. ((3 x 3) + (4 x 2) = 17 assuming ADD-2, MULTI-2 and MULTI-3 are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
11. The value 17 is given to node F, F=17, and is propagated.
    - No Conflicts Exist ⇒ No Diagnoses
12. Propagation of F and X result in assumption at point Y. (17 - (2 x 4) = 9 assuming ADD-1, MULTI-1 are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
13. Propagation of F and Y result in assumption at point X. (17 - (3 x 3) = 8 assuming ADD-1 and MULTI-2 are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
14. Propagation of X, F, Y and Z result in assumption at point G. ((17 - (2 x 4)) + (2 x 4) = 17 assuming ADD-1, ADD-2, MULTI-1 and MULTI-3 are functioning correctly)
    - No Conflicts Exist ⇒ No Diagnoses
15. The value 19 is given to node G, G=19, and is propagated.
    - Conflicts Exist:

    - {ADD-2, MULTI-2, MULTI-3}
    - {ADD-1, ADD-2, MULTI-1, MULTI-3}
  - ADD-2, MULTI-2 or MULTI-3 maloperated and ADD-1, ADD-2, MULTI-1 or MULTI-3 maloperated

16. Propagation of G and Y result in assumption at point Z. (19 - (3 x 3) = 10 assuming ADD-2 and MULTI-2, or ADD-2, ADD-1 and MULTI-1, are functioning correctly)
    - Conflicts Exist:
      - {ADD-2, MULTI-2, MULTI-3}
      - {ADD-1, ADD-2, MULTI-1, MULTI-3}
      - {ADD-2, MULTI-2}
      - {ADD-2, ADD-1, MULTI-1}
    - ADD-2, MULTI-2 or MULTI-3 maloperated and ADD-1, ADD-2, MULTI-1 or MULTI-3 maloperated and ADD-2 or MULTI-3 maloperated and ADD-2, ADD-1 or MULTI-1 maloperated

17. Propagation of G and Z result in assumption at point Y (19 - (2 x 4) = 11 assuming ADD-2 and MULTI-3 are functioning correctly)
    - Conflicts Exist:
      - {ADD-2, MULTI-2, MULTI-3}
      - {ADD-1, ADD-2, MULTI-1, MULTI-3}
      - {ADD-2, MULTI-2}
      - {ADD-2, ADD-1, MULTI-1}
      - {ADD-1, ADD-2, MULTI-3}
    - ADD-2, MULTI-2 or MULTI-3 maloperated and ADD-1, ADD-2, MULTI-1 or MULTI-3 maloperated and ADD-2 or MULTI-3 maloperated and ADD-2, ADD-1 or MULTI-1 maloperated and ADD-1, ADD-2 or MULTI-3 maloperated

18. Propagation of F, Y, G and Z result in assumption at point X. (17 - (19 - (2 x 4)) = 6 assuming ADD-1, ADD-2 and MULTI-3 are functioning correctly)
    - Conflicts Exist:
      - {ADD-2, MULTI-2, MULTI-3}
      - {ADD-1, ADD-2, MULTI-1, MULTI-3}
      - {ADD-2, MULTI-2}
      - {ADD-2, ADD-1, MULTI-1}

  - {ADD-1, ADD-2, MULTI-3}
- ADD-2, MULTI-2 or MULTI-3 maloperated and ADD-1, ADD-2, MULTI-1 or MULTI-3 maloperated and ADD-2 or MULTI-3 maloperated and ADD-2, ADD-1 or MULTI-1 maloperated and ADD-1, ADD-2 or MULTI-3 maloperated

With the final observations available, the conflict sets due to G's mismatched observation and propagation are:

- {ADD-2, MULTI-2, MULTI-3}
- {ADD-1, ADD-2, MULTI-1, MULTI-3}

Note this is cut down from during diagnosis because the removed conflicts were caused by multiple assumptions rather than actual observations.

This effectively means that either ADD-2, MULTI-2 or MULTI-3 maloperated and either ADD-1, ADD-2, MULTI-1 or MULTI-3 maloperated.

The following steps are then taken to generate a set of hypotheses or diagnoses ("fault candidate set") from the set of conflicts.

Step 1: Remove Duplicates
Conflict sets:

- {ADD-2, MULTI-2, MULTI-3}
- {ADD-1, ADD-2, MULTI-1, MULTI-3}

Step 2: Remove Super-sets
Conflict sets:

- {ADD-2, MULTI-3}{MULTI-2}
- {ADD-2, MULTI-3}{ADD-1, MULTI-1}

Step 3: Remove Common Conflicts and Multiply
Extract:

- {ADD-2}
- {MULTI-3}

Multiply:

- {MULTI-2}
- {ADD-1, MULTI-1}

Diagnoses:

- {ADD-2}
- {MULTI-3}

- {MULTI-2, ADD-1}
- {MULTI-2, MULTI-1}

Therefore the minimal diagnoses are: {ADD-1} and {MULTI-3}, translating as the most likely fault is associated with either ADD-1 or MULTI-3, meaning the failure of either of these components alone could lead to the conflicts observed; otherwise to replicate these conflicts both MULTI-2 and ADD-1 or both MULTI-2 and MULTI-1 would have to have failed.

This demonstrates the GDE function on a non fully observable system, X, Y and Z values are never received. When systems are fully observable diagnoses is generally able to be traced directly to the fault component. The next example presents the same problem but with actual values X, Y and Z observable.

The observations inserted into the model, Figure 4.5 are: A=2, B=3, C=4, D=5, E=2, F=17, G=19.

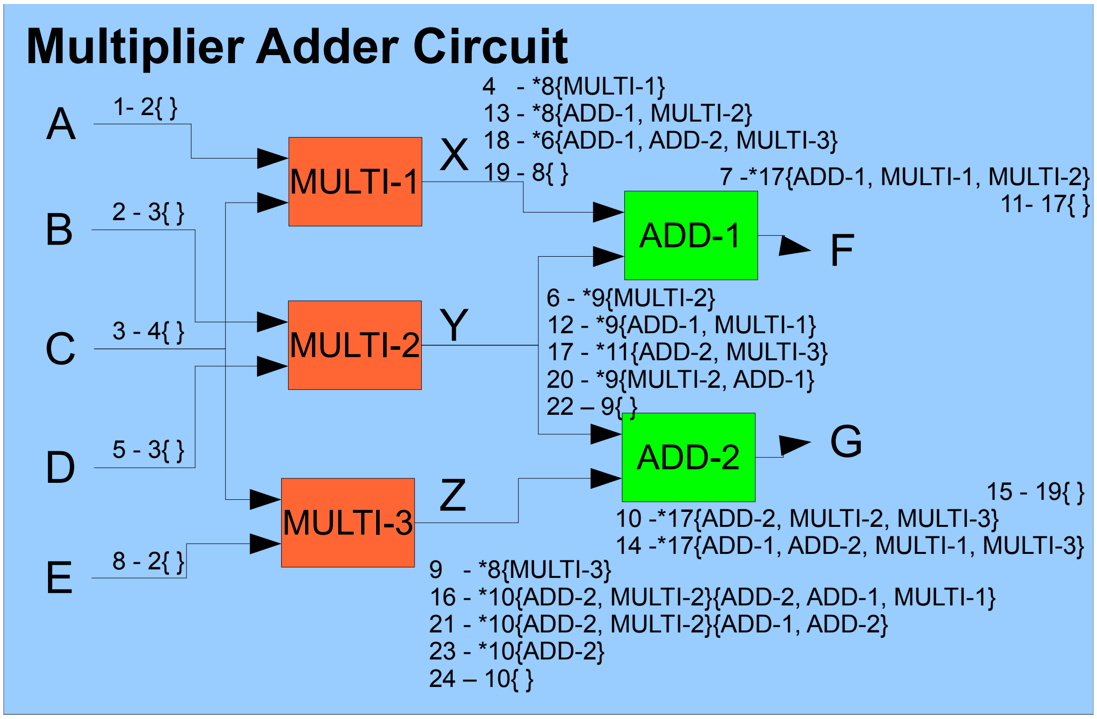


Figure 4.5: Faulty Fully Observable Multiplier Adder Circuit

With the final observations available, the conflict sets can be seen at:

- X where conflict lies in ADD-1, ADD-2 or MULTI-3
- Y where conflict lies in ADD-2 or MULTI-3
- Z where conflict lies in MULTI-3
- G where conflict lies in ADD-2, MULTI-2, MULTI-3 or ADD-1, ADD-2, MULTI-1, MULTI-3

It can therefore be seen that the common factor in all of these conflicts is:

- MULTI-3

Given a single common conflict set exists the diagnosis is therefore: MULTI-3 maloperated. It can also be seen that given the observation, Z=10, this is not the result of 4 x 2, and can be seen to cause the propagation of error to the other devices.

4.3.5 GDE Implementation

The GDE implementation in Lisp by Forbus and de Kleer [Forbus and de Kleer, 1993] is implemented to directly map to the definition of GDE. This implementation has been utilised for PSDiagnosis and diagnosis results ported into Java. It therefore consists of the following modules:

- ATMS (Assumption Based Truth Maintenance System): this module carries out the diagnostic functions. Thus meaning it caches all incoming data, and all of the inferences the data has produced. The ATMS also carries out conflict detection, and candidate generation and refinement.
- ATCON (Assumption Based Constraint Language): this module provides extensions to the Lisp language, it is divided into nine functions [Forbus and de Kleer, 1993]:
  1. “Definitions & Initialisation”- procedures for building and loading data structures.
  2. “Defining Prototypes”- procedures for reading the constraints applied to specific models.
  3. “Creating Constraints”- procedures for compiling the constraints applied to specific models.
  4. “Accessors”- procedures for accessing the constraints and their elements.
  5. “Equality Systems”- Implementing the comparison procedure for identifying conflicts.
  6. “Setting and Accessing Cell Values”- implementing the injection procedures for new data or inferences.
  7. “Rule Execution”- Controls the scheduling or ATCON rule execution.
  8. “Constructing Solutions”- Interfaces with the ATMS for the construction of diagnoses.
  9. “Interrogatives”- procedures for accessing and printing names and values from the system.

A full description of this software and functionality is defined in [Forbus and de Kleer, 1993]. The following sections discuss the steps required to apply this technique to the post-fault analysis problem.

4.4 Applying the GDE to Protection Validation

4.4.1 Protection System Behaviour under Failures

To explore the behaviour of the protection system under failure, the following examples demonstrate the behaviour of the system based on the failure of a single component in various sections of the network and on the failure of a device's communication or non operation of a device. This exploration provides a basis for the understanding required to construct models of "normal" behaviour and therefore model the protection system. The example network for this description can be seen in Figure 4.6.

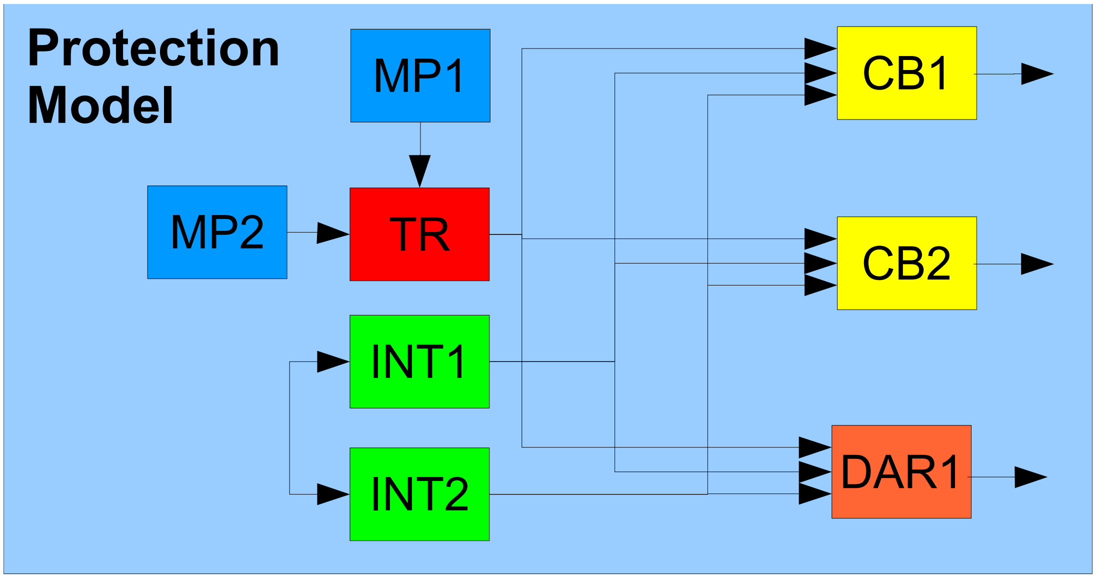


Figure 4.6: Example Network

Failure of a single component (untimely operation)

This section discusses the problems involved in a single device operating in an incorrect manner. All devices have specific operation times involved, for example: a trip relay introduces a 10ms delay, a circuit breaker a 40ms delay and an intertrip 30ms between each trip. It is important to make sure that all devices are operating within a certain tolerance of their designed time delay. Any additional delay on the operation of a device can lead to increased damage to expensive equipment or increased duration of potentially dangerous fault conditions. As shown in the example network (Figure 4.6) the devices are interconnected in such a way that, for example, a delay incurred by the trip relay would be passed on to each circuit breaker and the DAR. A more concerning and difficult to predict malfunction is caused when a device fails to operate completely or the alarm identifying its operation goes missing. The behaviours when this kind of failure occurs are described in the next section. It should be noted that for this type of analysis the time delays are all calculated based on the earliest alarm time-stamp, therefore the trigger for the incident would have an alarm time of 0ms.

Missing Alarms and Non-operation

Although the protection network under review is fully observable there are cases where alarms can fail to be received by the control or protection engineers. This can be caused by a range of problems from the device itself failing to communicate its state change to communications failures at RTUs (Remote Telemetry Units) and control rooms. The causes of these problems are outside the scope of this thesis however the problems caused by their malfunction must be understood. The following sections describe the effect on the network between a “missing” alarm situation and a non-operation and the importance of being able to detect and diagnose the differences between them.

Failure of Trip Relay Trip relays are always expected to operate under fault conditions. Their non-operation can have a devastating effect on the equipment where it could be inflicted with high fault currents causing large amounts of wear or even fatal damage of circuit breakers. It is therefore important to distinguish between the occurrences of genuine non-operation of trip relays and the occurrence of missing alarms. The next two case studies show the effect on incoming data of firstly the non-operation of a trip relay and secondly the non-arrival of the trip relay signal. It should be noted that although multiple trip coils are usually available, within the network operators SCADA data there is no ability to determine which coil has resulted in the alarms production.

Table 4.1: SCADA Alarms - Failure of Trip Relay

SCADA Alarms (Trip Relay (non-operation))		
Index	**Alarm Name & Text**	**Alarm Time(ms)**
A	1ST MAIN PROT OPTD ON	0
B	2ND MAIN PROT OPTD ON	0
D	INT 1 REC ON	40
E	INT 2 REC ON	70

Table 4.2: SCADA Alarms - Failure of Trip Relay (Missing Alarm)

SCADA Alarms (Trip Relay (missing alarm))		
Index	**Alarm Name & Text**	**Alarm Time(ms)**
A	1ST MAIN PROT OPTD ON	0
B	2ND MAIN PROT OPTD ON	0
D	INT 1 REC ON	40
E	INT 2 REC ON	70
F	CB1 OPEN	50
G	CB1 CLOSED	20010
H	CB2 OPEN	50

As shown the differences in the alarms received for a missing and non-operational trip relay are extensive. If an alarm is missing all devices which rely on its operation will operate as expected whereas if a device fails to operate any devices relying on its operation will also fail to operate.

Failure of Intertrip 1 Intertrips are always expected to operate under fault conditions however their modelling is challenging. Often intertrips do not initiate the operation of a circuit breaker with the main protections and trip relay carrying out the operation. In this case it is difficult to determine the difference between the non-reception of an alarm and a non-operation of an intertrip. This can be seen in the following two examples one of which the intertrip has not arrived and one of which the intertrip itself failed.

Table 4.3: SCADA Alarms - Failure of Intertrip

SCADA Alarms (Intertrip (non-operation))		
Index	**Alarm Name & Text**	**Alarm Time(ms)**
A	1ST MAIN PROT OPTD ON	0
B	2ND MAIN PROT OPTD ON	0
C	TRIP RELAYS TO BE RESET-E	10
E	INT 2 REC ON	70
F	CB1 OPEN	50
G	CB1 CLOSED	20010
H	CB2 OPEN	50

Table 4.4: SCADA Alarms - Failure of Intertrip (Missing Alarm)

SCADA Alarms (Intertrip (missing alarm))		
Index	**Alarm Name & Text**	**Alarm Time(ms)**
A	1ST MAIN PROT OPTD ON	0
B	2ND MAIN PROT OPTD ON	0
C	TRIP RELAYS TO BE RESET-E	10
E	INT 2 REC ON	70
F	CB1 OPEN	50
G	CB1 CLOSED	20010
H	CB2 OPEN	50

Both tables demonstrate that the sequence of events are identical where the intertrip is not directly responsible for the tripping of the circuit breakers. Where the operation of an intertrip device occurs after that of the trip relays, the non-arrival of its alarm is indistinguishable from its non-operation.

Where the intertrip provides the signal to the circuit breaker to open it is far easier to distinguish between a fault and a missing alarm, as shown in the next two tables.

Table 4.5: SCADA Alarms - Failure of Intertrip (No Circuit Breaker Operation)

SCADA Alarms (Intertrip (non-operation))		
Index	**Alarm Name & Text**	**Alarm Time(ms)**
E	INT 2 REC ON	70
F	CB1 OPEN	110
G	CB1 CLOSED	20010
H	CB2 OPEN	110

Table 4.5 presents the sequence of events which would occur if INT 1 failed to operate but should have triggered the operation of both circuit breakers i.e. should have operated at 10ms tripping the circuit breakers at 50ms.

Table 4.6: SCADA Alarms - Failure of Intertrip (Missing Alarm & No Circuit Breaker Operation)

SCADA Alarms (Intertrip (missing alarm))		
Index	**Alarm Name & Text**	**Alarm Time(ms)**
E	INT 2 REC ON	70
F	CB1 OPEN	50
G	CB1 CLOSED	20010
H	CB2 OPEN	50

Table 4.6 presents the sequence of events which would occur if INT 1 operated but its alarm failed to arrive. Therefore the circuit breakers response at 50ms is correct, even though its occurrence appears before the arrival of INT2 operation.

From both of these tables it can be seen that the difference between a missing alarm and a device failure are significant and therefore would be able to be diagnosed as such.

Failure of Circuit Breaker Circuit breakers are always expected to open under fault conditions. Their non-operation can have a devastating effect on the equipment where it could be inflicted with high fault currents causing large amounts of wear or even fatal damage to both itself and other equipment on the circuit. It is therefore important to distinguish between the occurrences of genuine non-operation of circuit breakers and the occurrence of missing alarms. The next two case studies show the effect on incoming data of firstly the non-operation of a circuit breaker and secondly the non-arrival of the circuit breaker signal.

Table 4.7: SCADA Alarms - Failure of Circuit Breaker

SCADA Alarms (Circuit Breaker 1 (non-operation))		
Index	**Alarm Name & Text**	**Alarm Time(ms)**
A	1ST MAIN PROT OPTD ON	0
B	2ND MAIN PROT OPTD ON	0
C	TRIP RELAYS TO BE RESET-E	10
D	INT 1 REC ON	40
E	INT 2 REC ON	70
H	CB2 OPEN	50

Table 4.8: SCADA Alarms - Failure of Circuit Breaker (Missing Alarm)

SCADA Alarms (Circuit Breaker 1 (missing alarm))		
Index	**Alarm Name & Text**	**Alarm Time(ms)**
A	1ST MAIN PROT OPTD ON	0
B	2ND MAIN PROT OPTD ON	0
C	TRIP RELAYS TO BE RESET-E	10
D	INT 1 REC ON	40
E	INT 2 REC ON	70
G	CB1 CLOSED	20010
H	CB2 OPEN	50

As shown the differences in the alarms received for a missing and non-operational circuit breaker are minimal. If an alarm is missing the DAR can be seen to operate, whereas without the DAR cannot be seen. For example CB 2 doesn't have a DAR and in this situation the alarms would be identical on both the non-operation of the circuit breaker and on the circuit breaker alarm being missing. This means that circuit breakers, as with intertrips, can be difficult to diagnose. On this occurrence further investigations would be required to identify whether other surrounding circuits tripped to isolate the fault and observe the occurrence of any breaker failure alarms which although not analysed would be associated with the incident. It should be noted that it is rare that a circuit breaker would not have a DAR, however this can be seen in order to prevent any reclosure onto a persitant fault within a T-shaped circuit. In this case the circuit breaker without DAR would only be manually reclosed after the other circuit breakers confirmed that the fault has been cleared. This can be implemented as a means to additionally protect expensive plant from fault current.

4.4.2 Behavioural Models

Model-based diagnosis requires some form of model to represent the system of interest. In the case of a fully observably system, such as the SCADA alarm processing problem models can be produced of each individual element in the overall system and then new facts can both be inserted and propagated at each connection between the devices. A simple example of the propagation of both facts (alarms in this case) and assumptions can be seen in Figure 4.7.

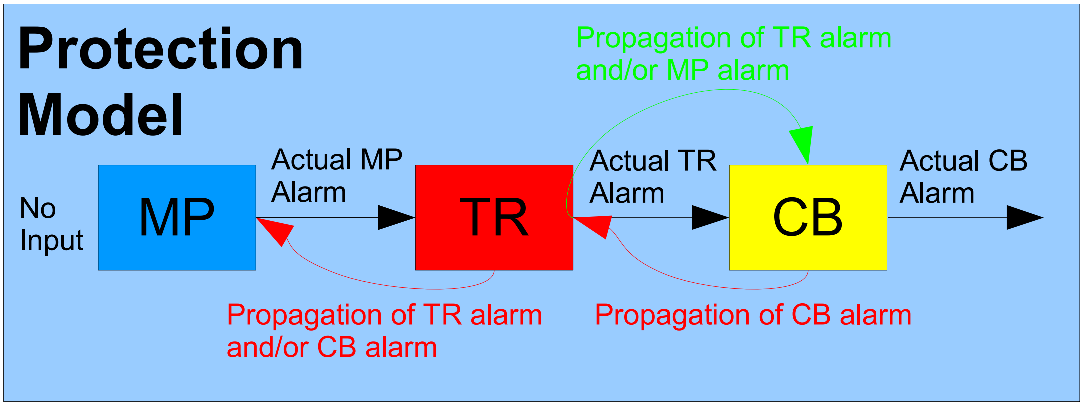


Figure 4.7: Alarm insertion and propagation

In the case of SCADA-based model-based diagnosis the models themselves can be logical representations of the expected behaviour of devices given either actual observations or the known operation of the devices around it. In order to effectively model the behaviour of each device within the protection system, the models need to be configured to predict behaviours both forward from inputs to outputs and backwards from outputs to inputs. The protection system can be seen as fully observable as during correct SCADA system operation, each device operation would in turn be expected to trigger an alarm. In particular for the validation of protection performance, the time delays associated with the logical propagation of outputs to other devices is vital. A component library was produced consisting of the following device types:

- Main Protection (MP)
- Trip Relay (TR)
- Circuit Breaker (CB)
- Delated Automatic Recloser (DAR)
- Intertrip (INT) (these devices due to a current lack of synchronisation between sites in the test data can only be validated within the substation i.e when would each intertrip be expected to arrive and what effect would this have on the scheme)
- Circuit Breaker Manual Recloser (MR- abbreviated for diagram purposes only.)

The component library has been constructed to provide multiple settings for each device i.e. trip relays connected to multiple or single main protection devices and delayed automatic reclosers can be configured for different automatic reclosure settings. Utilising both the protection database, as discussed in section 2.4.4, and the component library allows the behavioural models to be configured at run time for each individual circuit on the network.

In order to effectively model the behaviour of each device within the protection system the models need to be configured to predict outputs from inputs and inputs from outputs. The behavioural models rely on the specific timings for each of the inputs and outputs. The following describes the functions for all devices within the protection system:

Trip Relay

The trip relay's function on the protection system is to send the trip command to a circuit breaker when it receives notification of a fault. Although each circuit breaker is connected to a single trip relay with multiple trip coils, within the SCADA communications alarms they are indecipherable and where multiple circuit breakers are present at a single circuit end a single "master" trip relay must be modelled. This then assumes that whenever the trip-relay operation alarm is received all circuit breakers would be expected to operate regardless of the specific relay connection. Although ΔTR is utilised in the algorithm the system has a general "default" TR operating delay of 10ms. This is determined as a static operational time due to the time-tagging of SCADA data, for these examples a 20ms operation could signify an operation up to 29ms and therefore a deviation of as little as 10ms, to 20ms, are classed as problematic operations.

The following tables provide the algorithms required to model the different types of trip relays within the protection network (where T_A denotes the value at node A):

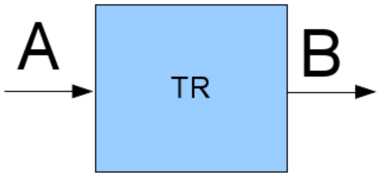


Figure 4.8: Trip Relay One Input, One Output

Table 4.9: Trip Relay One Input, One Output

Trip Relay (One Input, One Output) (Figure 4.8.)	
Constraint 1 - A is Known	**Constraint 2 - B is Known**
$T_B = T_A + \Delta TR$	$T_A = T_B - \Delta TR$

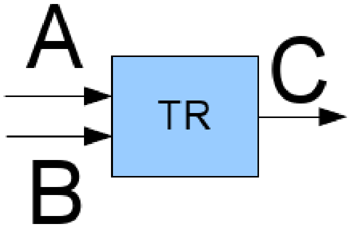


Figure 4.9: Trip Relay Two Input, One Output

Table 4.10: Trip Relay Two Input, One Output

Trip Relay (Two Input, One Output) (Figure 4.9.)	
Constraint 1 - A & B Known	**Constraint 2 - A & C Known**
$if(T_A < T_B)$ $T_C = T_A + \Delta TR$ $else$ $T_C = T_B + \Delta TR$	$if(T_C - T_A) < \Delta TR$ $T_B = T_C - \Delta TR$ $else$ $T_B = Unknown$
Constraint 3 - B & C Known	
$if(T_C - T_B) < \Delta TR$ $T_A = T_C - \Delta TR$ $else$ $T_A = Unknown$	

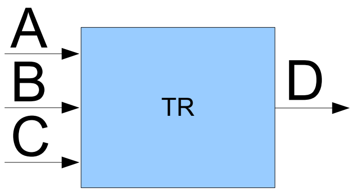


Figure 4.10: Trip Relay Three Input, One Output

Table 4.11: Trip Relay Three Input, One Output

Trip Relay (Three Input, One Output) (Figure 4.10.)	
Constraint 1 - A, B & C Known	**Constraint 2 - A, B & D Known**
$if(T_A < T_B) and (T_A < T_C)$ $T_D = T_A + \Delta TR$ $elseif(T_B < T_C)$ $T_D = T_B + \Delta TR$ $else$ $T_D = T_C + \Delta TR$	$if(T_A < T_B)$ $if(T_D - T_A) < \Delta TR$ $T_C = T_D - \Delta TR$ $else$ $T_C = Unknown$ $elseif(T_D - T_B) < \Delta TR$ $T_C = T_D - \Delta TR$ $else$ $T_C = Unknown$
Constraint 3 - A, C & D Known	**Constraint 4 - B, C & D Known**
$if(T_A < T_C)$ $if(T_D - T_A) < \Delta TR$ $T_B = T_D - \Delta TR$ $else$ $T_B = Unknown$ $elseif(T_D - T_C) < \Delta TR$ $T_B = T_D - \Delta TR$ $else$ $T_B = Unknown$	$if(T_B < T_C)$ $if(T_D - T_B) < \Delta TR$ $T_A = T_D - \Delta TR$ $else$ $T_A = Unknown$ $elseif(T_D - T_C) < \Delta TR$ $T_A = T_D - \Delta TR$ $else$ $T_A = Unknown$

Intertrip

The purpose of an intertrip is to allow trip signals to be communicated between both ends of a feeder. Due to the lack of GPS synchronisation of SCADA alarm timestamps the functionality of the intertrip itself cannot be modelled. However the behaviour of intertrips in respect to each other can be modelled. Where two intertrips are present in a feeder circuit the intertrips would be expected to arrive Δ INT ms apart. The following tables provide the algorithms required to model an intertrip within the protection network:

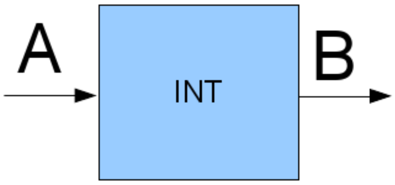


Figure 4.11: Intertrip

Table 4.12: Intertrip

Intertrip (One Input, One Output) (Figure 4.11.)	
Constraint 1 - A is Known	**Constraint 2 - B is Known**
$T_B = T_A - \Delta INT$ *or* $T_B = T_A + \Delta INT$	$T_A = T_B - \Delta INT$ *or* $T_A = T_B + \Delta INT$

Circuit Breaker

The purpose of a circuit breaker is to open and interrupt fault current during fault conditions. It is important that a circuit breaker responds in a timely manner during a fault to limit the damage to itself and other equipment on the network. Although ΔCB is utilised in the algorithm the system has a general "default" TR operating delay of 40ms. The following tables provide the algorithms required to model different circuit breaker configurations within the protection network:

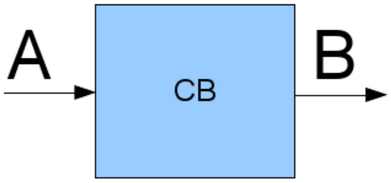


Figure 4.12: Circuit Breaker One Input, One Output

Table 4.13: Circuit Breaker One Input, One Output

Circuit Breaker (One Input, One Output) (Figure 4.12.)	
Constraint 1 - A is Known	**Constraint 2 - B is Known**
$T_B = T_A + \Delta CB$	$T_A = T_B - \Delta CB$

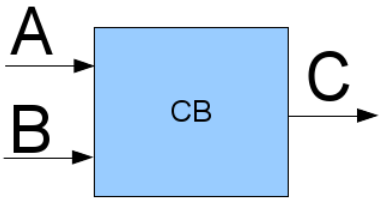


Figure 4.13: Circuit Breaker Two Input, One Output

Table 4.14: Circuit Breaker Two Input, One Output

Circuit Breaker (Two Input, One Output) (Figure 4.13.)	
Constraint 1 - A & B Known	**Constraint 2 - A & C Known**
$if(T_A < T_B)$ $\quad T_C = T_A + \Delta CB$ $else$ $\quad T_C = T_B + \Delta CB$	$if(T_C - T_A) < \Delta CB$ $\quad T_B = T_C - \Delta CB$ $else$ $\quad T_B = Unknown$
Constraint 3 - B & C Known	
$if(T_C - T_B) < \Delta CB$ $\quad T_A = T_C - 40$ $else$ $\quad T_A = Unknown$	

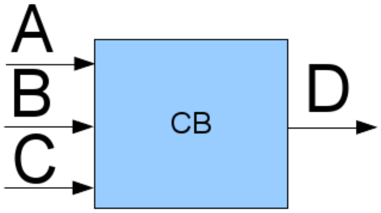


Figure 4.14: Circuit Breaker Three Input, One Output

Table 4.15: Circuit Breaker Three Input, One Output

Circuit Breaker (Three Input, One Output) (Figure 4.14.)	
Constraint 1 - A, B & C Known	**Constraint 2 - A, B & D Known**
$if(T_A < T_B)and(T_A < T_C)$ $T_D = T_A + \Delta CB$ $elseif(T_B < T_C)$ $T_D = T_B + \Delta CB$ $else$ $T_D = T_C + \Delta CB$	$if(T_A < T_B)$ $if(T_D - T_A) < \Delta CB$ $T_C = T_D - \Delta CB$ $else$ $T_C = Unknown$ $elseif(T_D - T_B) < \Delta CB$ $T_C = T_D - \Delta CB$ $else$ $T_C = Unknown$
Constraint 3 - A, C & D Known	**Constraint 4 - B, C & D Known**
$if(T_A < T_C)$ $if(T_D - T_A) < \Delta CB$ $T_B = T_D - \Delta CB$ $else$ $T_B = Unknown$ $elseif(T_D - T_C) < \Delta CB$ $T_B = T_D - \Delta CB$ $else$ $T_B = Unknown$	$if(T_B < T_C)$ $if(T_D - T_B) < \Delta CB$ $T_A = T_D - \Delta CB$ $else$ $T_A = Unknown$ $elseif(T_D - T_C) < \Delta CB$ $T_A = T_D - \Delta CB$ $else$ $T_A = Unknown$

Delayed Automatic Recloser (DAR)

The purpose of a DAR is to re-close a circuit breaker after a delay, ΔDAR. The delay is set to allow ample time for fault clearance while ensuring line down time is kept to a minimum. Due to the larger time frames associated with DAR devices their models must implement a tolerance of one second or 1000ms onto the model delay requirements, meaning that a DAR can operate up to 1 second early without inserting errors during backwards propagation to trip relays etc. For a single input DAR model this effect means that although inputs are able to be propagated forwards, the errors associated with attempting to propagate the operaation backwards could be too great and therefore is not performed.

The following tables provide the algorithms required to model different DAR configurations within the protection network:

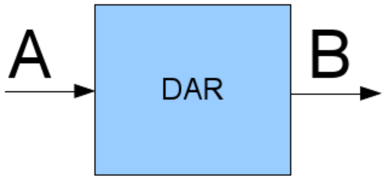


Figure 4.15: DAR One Input, One Output

Table 4.16: DAR One Input, One Output

DAR (One Input, One Output) (Figure 4.15.)	
Constraint 1 - A is Known	**Constraint 2 - B is Known**
$T_B = T_A + \Delta DAR$	$T_A = Unknown$

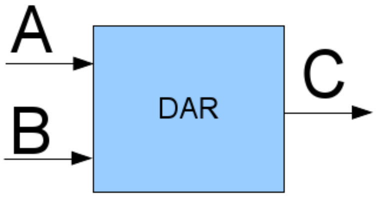


Figure 4.16: DAR Two Input, One Output

Table 4.17: DAR Two Input, One Output

DAR (Two Input, One Output) (Figure 4.16.)	
Constraint 1 - A & B Known	**Constraint 2 - A & C Known**
$if(T_A < T_B)$ $\quad T_C = T_A + \Delta DAR$ $else$ $\quad T_C = T_B + \Delta DAR$	$if(T_C - T_A) < (\Delta DAR - 1000)$ $\quad T_B = T_C - \Delta DAR$ $else$ $\quad T_B = Unknown$
Constraint 3 - B & C Known	
$if(T_C - T_B) < (\Delta DAR - 1000)$ $\quad T_A = T_C - \Delta DAR$ $else$ $\quad T_A = Unknown$	

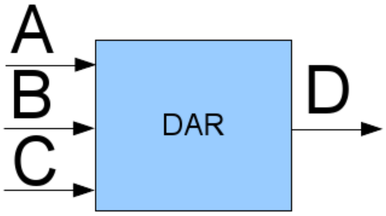


Figure 4.17: DAR Three Input, One Output

Table 4.18: DAR Three Input, One Output

DAR (Three Input, One Output) (Figure 4.17.)	
Constraint 1 - A, B & C Known	**Constraint 2 - A, B & D Known**
$if(T_A < T_B)and(T_A < T_C)$ $\quad T_D = T_A + \Delta DAR$ $elseif(T_B < T_C)$ $\quad T_D = T_B + \Delta DAR$ $else$ $\quad T_D = T_C + \Delta DAR$	$if(T_A < T_B)$ $\quad if(T_D - T_A) < (\Delta DAR - 1000)$ $\qquad T_C = T_D - \Delta DAR$ $\quad else$ $\qquad T_C = Unknown$ $elseif(T_D - T_B) < (\Delta DAR - 1000)$ $\quad T_C = T_D - \Delta DAR$ $else$ $\quad T_C = Unknown$
Constraint 3 - A, C & D Known	**Constraint 4 - B, C & D Known**
$if(T_A < T_C)$ $\quad if(T_D - T_A) < (\Delta DAR - 1000)$ $\qquad T_B = T_D - \Delta DAR$ $\quad else$ $\qquad T_B = Unknown$ $elseif(T_D - T_C) < (\Delta DAR - 1000)$ $\quad T_B = T_D - \Delta DAR$ $else$ $\quad T_B = Unknown$	$if(T_B < T_C)$ $\quad if(T_D - T_B) < (\Delta DAR - 1000)$ $\qquad T_A = T_D - \Delta DAR$ $\quad else$ $\qquad T_A = Unknown$ $elseif(T_D - T_C) < (\Delta DAR - 1000)$ $\quad T_A = T_D - \Delta DAR$ $else$ $\quad T_A = Unknown$

4.4.3 System Structure

Protection schemes are made up of a collection of these devices all interconnected to protect a specific section of the power network. Figure 4.18 shows the connectivity of devices for an example protection network. As soon as the devices present on a circuit are known the connectivity of devices can be inferred due to a standard of connectivity for protection devices, i.e. trip relays are always connected to a circuit breaker. The combination of behavioural models and system structure forms the GDE model for the system. The protection system is fully observable and therefore is is assumed that all inputs and outputs are also fully observable, with exception to the main protection devices as the fault inception time they operate on is unavailable within SCADA.

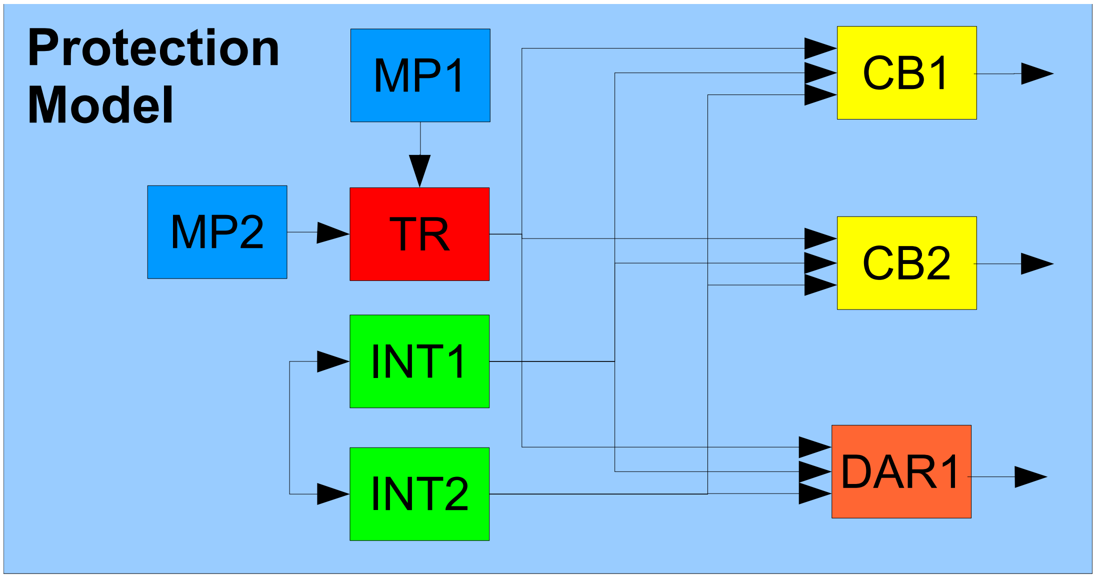


Figure 4.18: Protection Model

4.4.4 Conflict Detection

As discussed earlier appropriate conflict detection methods are vital to identify the points at which the observed behaviour of the protection system have deviated from the expected behaviour to the point that they have maloperated.

4.4.5 Fault Candidate Generation

Due to the "generic" features of the GDE the exploitation of the theory of diagnosis is able to remain unchanged for the post-fault protection validation to occur. After conflicts have been generated the theory of diagnosis must be applied.

Minimal Diagnosis

The notion of a minimal diagnosis candidate set is defined as a set of components which when assumed to be faulty and all other components assumed to be operating correctly can explain the behaviour being observed. The process of reducing the "candidate sets" into a minimal diagnosis utilises the theory of set covering, and is therefore generic and able to remain unchanged for this system.

4.5 Modifying the GDE approach for protection validation assessment

This section discusses the complexities involved in the application of GDE for both normal and special cases, including the modifications required for this application.

4.5.1 Operational Tolerances

Unfortunately as with all systems both the measurements taken of the system and the models for protection validation can have uncertainties and inaccuracies associated with them. This means that the system must account for a tolerance on measured and inferred outcomes to prevent false conclusions. Different devices also require different tolerances for example DAR devices operate over a matter of seconds rather than milliseconds and therefore requires a larger tolerance of ±1 second. The SCADA system itself has a maximum resolution of 10 milliseconds, this means that in regard to the devices expected to operate within a matter of milliseconds of each other an appropriate tolerance cannot be set. Not only are inaccuracies and uncertainties present, as with many electronic devices, protection system devices in reality have an acceptable range of timing behaviours. Therefore to account for this "normal" behaviour, an optional setting is available for the system which permits specific device types or settings to provide a range of acceptable operation timings. This modification of GDE allows the analysis system to take into account not only measurement errors but also permit "acceptable" operational times for devices.

4.5.2 Missing Alarm Identification

For the power system protection network the state changes of all devices are assumed to be observable. Unfortunately the communications network associated with SCADA alarms can cause alarms to be significantly delayed or even missing. Therefore it is desirable to account for this by distinguishing between the occurrence of a missing alarm and a non-operational device. The GDE is designed to handle the missing alarm problem by permitting the empty set (non-reception of alarm) to not cause a conflict. However for this case the non-reception of an alarm must be flagged, allowing engineers to be aware of potential communications problems. The GDE required modification to allow for this missing alarm identification therefore allowing SCADA alarms to be analysed effectively.

"SCADA" Model

The "SCADA" model has been defined to allow the identification of missing alarms. At the end of an incident any devices deemed to have not operated will be injected with a SCADA alarm, if the device operated correctly and the alarm was simply missing a conflict will be produced identifying a missing alarm. If the device failed to operate, the propagation of the SCADA alarm will result in a conflict being created which would identify the maloperation of the device. SCADA models are one input, one output components which are configured to only allow forward propagation of a “missing” alarm forward, thus meaning that the inclusion of their models cannot result in the incorrect prediction of a “missing” alarm during normal analysis.

To demonstrate the operation of the modified GDE system a simplified example is shown in this section. This example is first provided without a means of identifying missing alarms and the second part demonstrates the use of missing alarm identification models. The example scheme has two main protections, two intertrips, a trip relay, 2 circuit breakers and a delayed automatic recloser (DAR). The expected alarm propagation times are: TR - +10ms, INT1/INT2 - ±30ms, CB - +40ms, DAR - +20,000ms. The structure for this scheme and steps involved in the analysis process can be seen in Figures 4.19, 4.20 and 4.21 with the alarms presented within Table 4.19.

Table 4.19: SCADA Alarms for Missing Alarm Example

SCADA Alarms		
Index	**Alarm Name & Text**	**Δ T (ms)**
MP1	1ST MAIN PROT OPTD ON	0
MP2	2ND MAIN PROT OPTD ON	0
INT1	INT 1 REC ON	40
INT2	INT 2 REC ON	70
CB1	CB1 OPEN	50
CB2	CB2 OPEN	50
DAR	CB2 CLOSED	20010

If the alarms arrive in the order provided in Table 4.19 then the following steps are carried out when no missing alarm detection is operating:

1. MP1 alarm arrives but cannot be propagated forwards until MP2 has arrived. (This is because if MP2 actually arrived before 0ms then any following devices, e.g. TR or CB, would be expected to operate at a different time.)
    - No Conflicts Exist ⇒ No Diagnoses
2. MP2 alarm arrives and is propagated.
    - No Conflicts Exist ⇒ No Diagnoses
3. INT1 alarm arrives, is propagated to INT2 which then results in propagation to CB1, CB2 and DAR.
    - No Conflicts Exist ⇒ No Diagnoses
4. INT2 alarm arrives, confirms the propagation of INT1.
5. CB1 alarm arrives and confirms the prediction for CB1 and backs up the prediction for TR.

- No Conflicts Exist ⇒ No Diagnoses

6. CB2 alarm arrives and confirms the prediction for CB2 and backs up the prediction for TR.
   - No Conflicts Exist ⇒ No Diagnoses
7. DAR alarm arrives and confirms the prediction for DAR.
   - No Conflicts Exist ⇒ No Diagnoses

Due to the non-arrival of alarm TR the final diagnosis is that nothing incorrect has occurred however missing alarms should be identified.

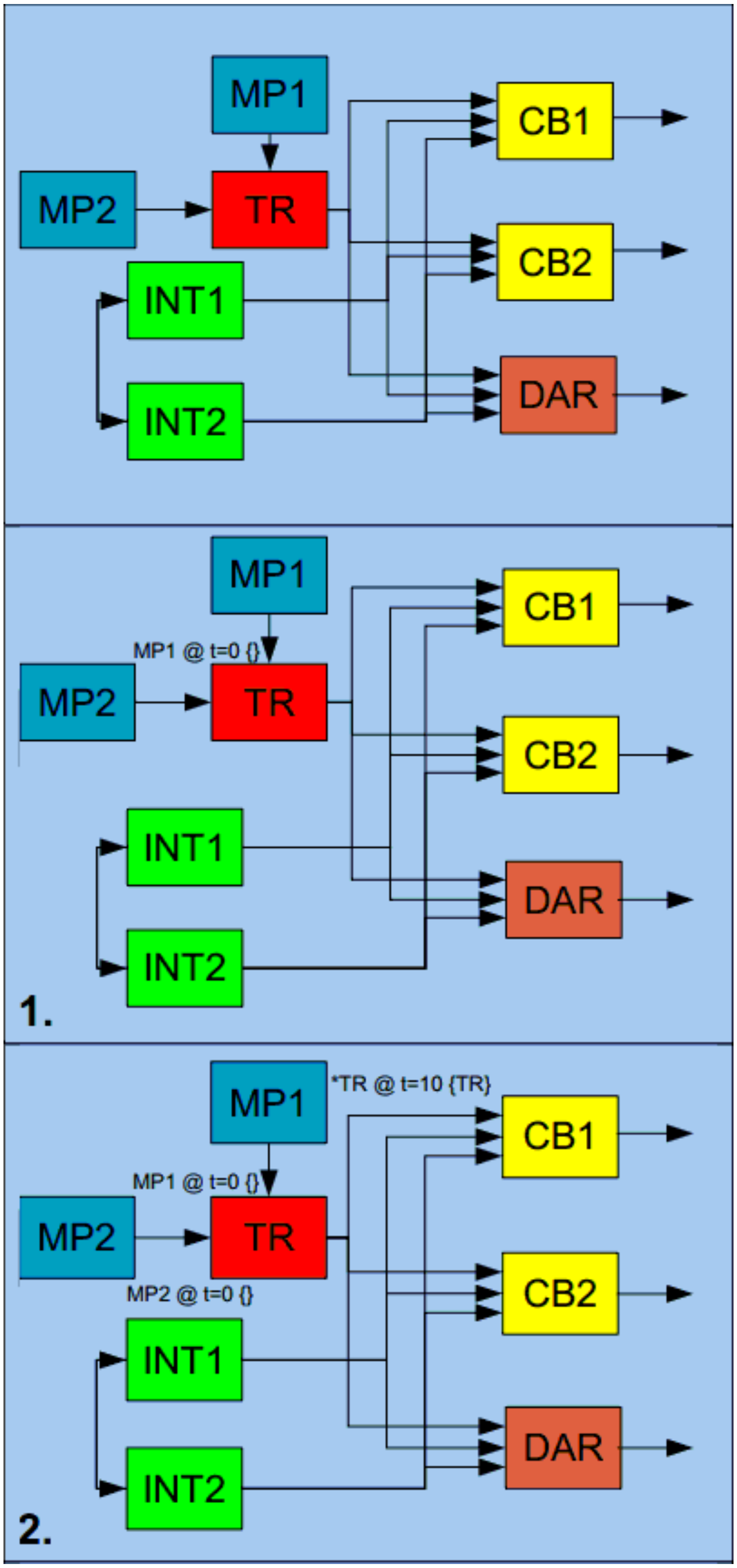


Figure 4.19: Protection Model - Steps 1 & 2

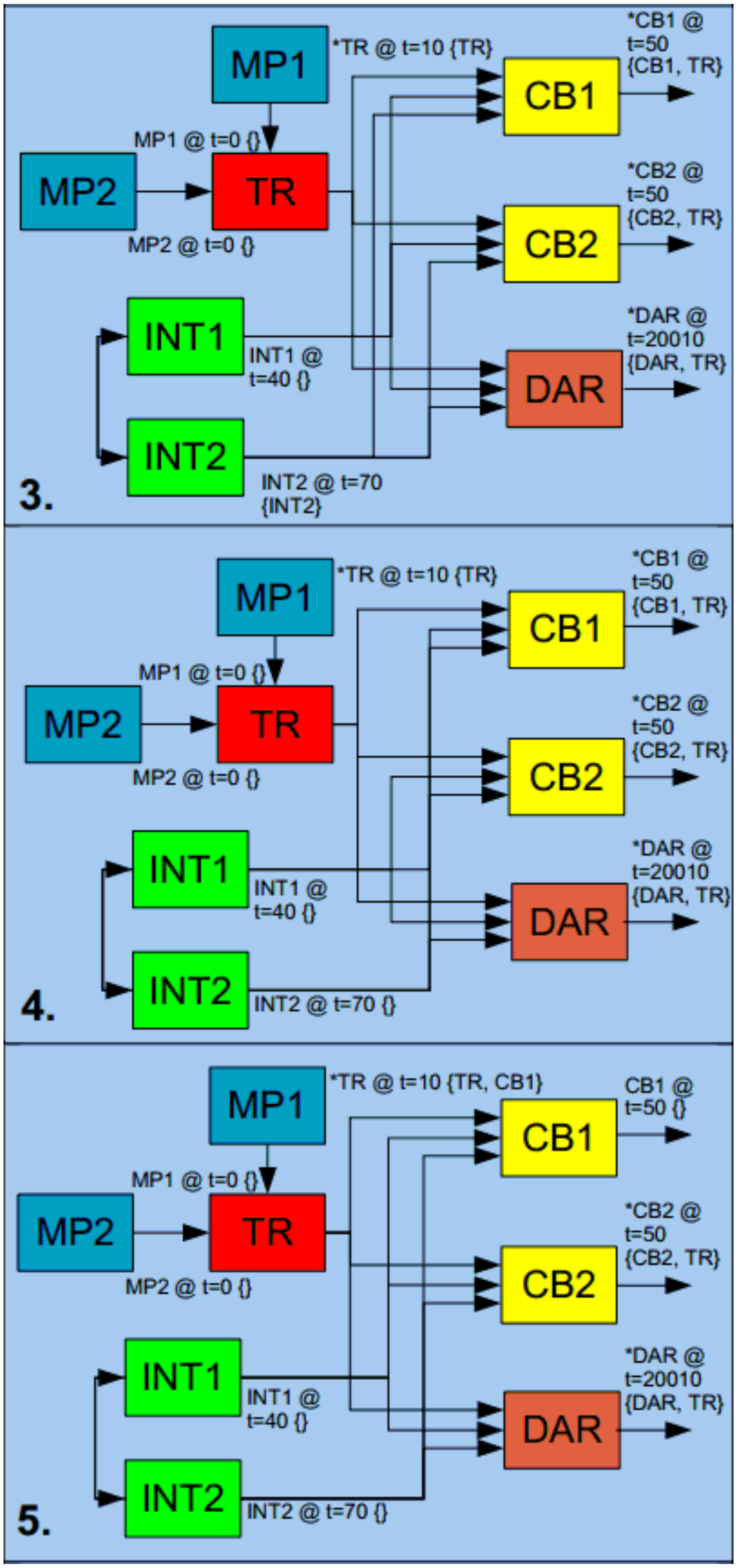


Figure 4.20: Protection Model - Steps 3, 4 & 5

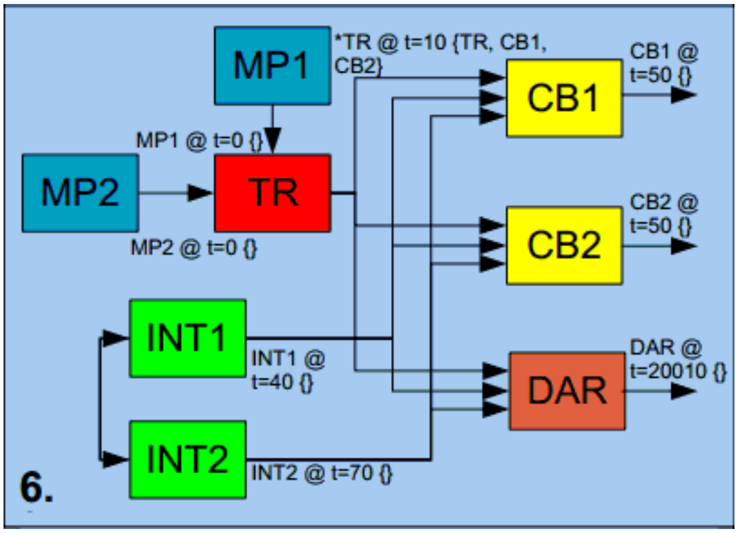


Figure 4.21: Protection Model - Step 6

The same example but using the missing alarm models, shown in Figure 4.22 is now demonstrated. With step 7 representing the insertion of a “missing alarm” into the SCADATR model, therefore presenting the propagation of a “missing alarm” and the consequences of its arrival on the TR output.

1. Same as previous example
2. Same as previous example
3. Same as previous example
4. Same as previous example
5. Same as previous example
6. Same as previous example
7. Missing alarm “TRIP RELAY OPTD ON” at T=9999999ms is inserted resulting in conflict.
    - Conflicts Exist {SCADATR},{TR, CB1, CB2} (Either the TR alarm was missing or TR, CB1 and CB2 all maloperated)
      ⇒ Minimal Diagnoses: {SCADATR}

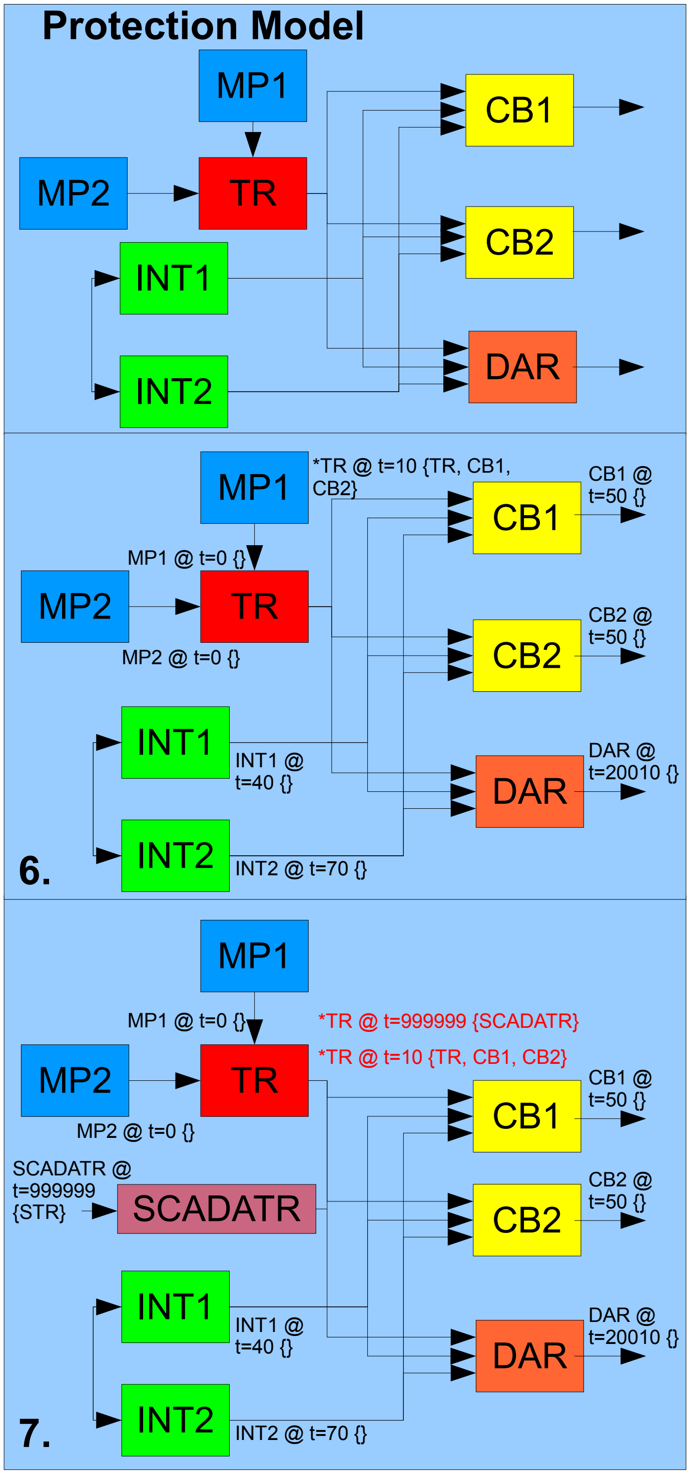


Figure 4.22: Protection Model for Missing Alarm Example - Detection

This method is able to clearly identify that an alarm has gone missing without introducing any false identification of maloperation. This novel advance within the GDE method, providing the identification of missing alarms, allows common communication problems to be accounted for and therefore improves the overall value of the SCADA analysis system.

4.6 System Architecture

Central to the design of the PSDiagnosis system is the importance of ease of maintenance and future proofing of the system. This section discusses how the MBD system has been configured to ensure maintainability and flexibility in the future. This flexibility is a benefit gained through the model-based approach and augmented GDE method which have been designed. The novel method allows for ease of maintenance by allowing the configuration of models, protection schemes and SCADA syntax to be carried out without impacting the rest of the system or requiring understanding of the underlying diagnostic algorithm. The high level architecture of the system can be seen in Figure 4.23.

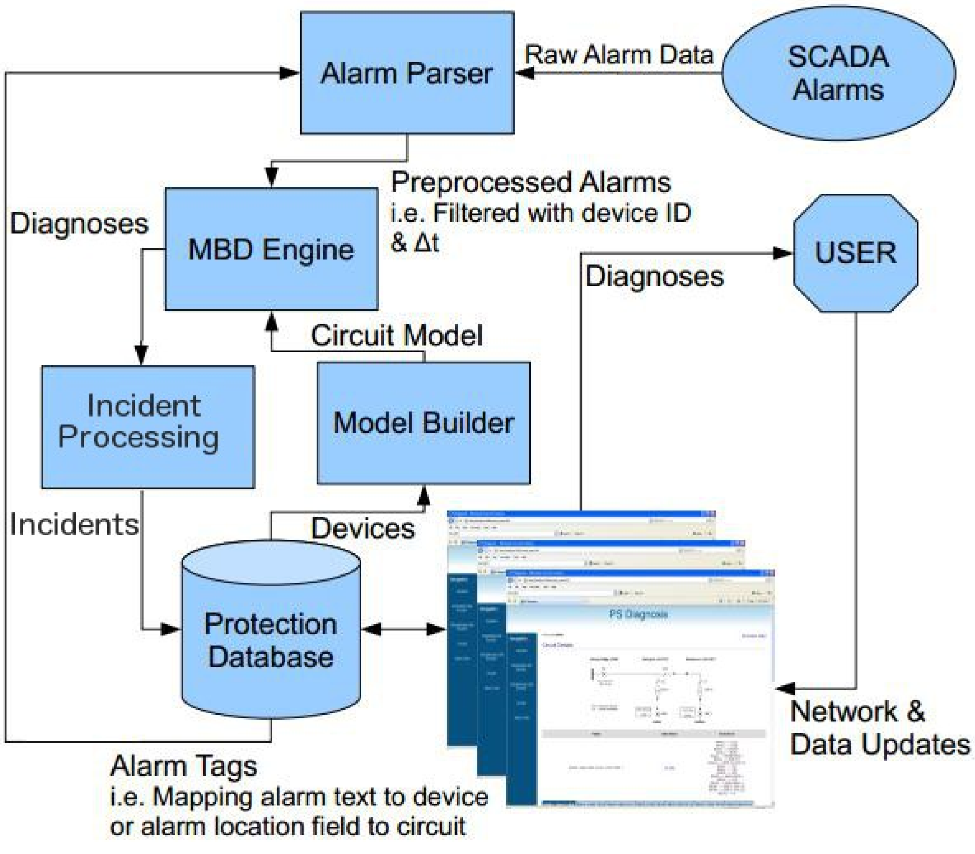


Figure 4.23: PSDiagnosis Architecture

4.6.1 System Modules

This section presents a break down of each module in the PSDiagnosis system and its function in the complete post-fault analysis cycle.

Protection Database

Central to the design of the PSDiagnosis system is the importance of ease of maintenance and future proofing of the system. A transmission system operator within the United Kingdom provided the information required to create a relational database linking the circuits and devices on their transmission network to the SCADA identifiers and alarm text provided by the SCADA system. These links are then harnessed during run-time to associate incidents to particular circuits and utilising this knowledge to dynamically build and configure the models for each device on the relevant circuit. The protection database is utilised by the alarm parser and model builder, with post-fault analysis data being uploaded when a diagnosis has been reached.

Alarm Parser

The Alarm Parser is responsible for filtering irrelevant alarms, i.e. removal of non operational alarms such as communications alerts, and the grouping of relevant alarms based on location. This system utilises the protection database to access the alarm tags which associate alarms to particular circuits and therefore tie alarms to devices. The models have been designed to utilise a simplified representation of an alarm, made up of a device id and Δ t. The alarm parser is able to strip unnecessary information from alarms allowing the MBD engine to insert alarms into the correct models. This simplified alarm format means that as long as these two values can be extracted the alarm itself could come from any source, for example IEC 61850 MMS or digital signals from DFRs.

Model Builder

Protection schemes are made up of a collection of interconnected devices, used to protect a specific section of the power network. When the devices present on a circuit are known the connectivity of devices can be inferred due to a relatively standard means of connectivity between protection devices. For example, main protection devices are connected to the trip relays for each circuit breaker on a circuit end. This combination of behavioural models and system structure forms the GDE model for the system. The model builder generates this structure at run time depending on the devices present on each circuit. This structure is identified using a rule based expert system, accessing the protection database, which allows not only system structure but also device specific settings to be identified. For example: specific relay types or circuit breakers may have abnormal operating times due to their age or model, this can be configured based on the identification of the device type within the protection database.

Incident Processing

This system module operates as an additional layer of analysis and formatting for diagnoses. The module utilises a set of rules which allow each incident to be assigned a priority/severity, a high level summary and an association with availble DFR records or adjoining incidents. These rules can be seen in Appendix C.3.

Web Front End

The transmission operator has been provided with a web front end within the corporate intranet which allows users of different privileges to view, amend and add new circuits or information to the database. The viewer provides a means for multiple user groups to not only keep up to date with the network's current settings but also calculate statistics, and the fault diagnosis inserted during incidents. This web front end allows batch updates of the network via CSV (Comma Separated Value) files or via a simple "form" within the software. These utilities mean that during routine maintenance of substations as well as the creation of new substation and equipment replacement any changes can easily be applied to the database. This in turn allows the Model Builder to access up-to-date network information during incidents.

4.6.2 Case Study of Interaction Between Modules

This case study is intended to provide an overview of how the PSDiagnosis system processes alarms and provides information to front-end users. This process can be split into 4 stages of analysis:

1. Alarm Collation and Filtering
2. Model Building and Configuration
3. Alarm Insertion and Diagnosis Extraction
4. Formatting of Diagnosis and Incident Summarisation

Alarm Collation and Filtering

Alarm collation and filtering is the first step carried out by the alarm parser, this module listens for any new alarms appearing in the SCADA database and polls any alarms associated with incidents i.e. All "operation" alarms. When an alarm is identified as relevant it is then utilised to access the protection database and retrieve all SCADA identifiers for the particular circuit end showing activity. As alarms are received any associated alarms are grouped, creating a sequence of SCADA events for analysis. When 2 minutes has elapsed, taken using SCADA alarm time rather than local due to differing propagation delays at different substations, an incident would be deemed to have closed and any alarms for that site arriving after 2 minutes will roll into the next incident. An example sequence of events given a trigger alarm of 1ST MAIN PROT OPTD ON at 25/07/2012 06:59:51.250 can be seen in Table. 4.20.

Table 4.20: SCADA Alarms for System Operation Example

SCADA Alarms			
Site Identifier	**Index**	**Alarm Name & Text**	**∆ T (ms)**
SUBA –> SUBB	MP1	1ST MAIN PROT OPTD ON	0
SUBA –> SUBB	MP2	2ND MAIN PROT OPTD ON	0
SUBA –> SUBB	TR	TRIP RELAY OPTD ON	20
SUBA –> CB1	CB1	CB1 OPEN	60
SUBA –> CB2	CB2	CB2 OPEN	60
SUBA –> SUBB	MP1	1ST MAIN PROT OPTD OFF	90
SUBA –> SUBB	MP2	2ND MAIN PROT OPTD OFF	120
SUBA –> SUBB	TR	TRIP RELAY OPTD OFF	11410
SUBA –> CB1	DAR1	CB1 CLOSED	20020
SUBA –> CB2	DAR2	CB2 CLOSED	22020

Model Building and Configuration

Once the circuit end of interest has been identified by the alarm parser, the model of that circuit can be constructed and configured. Utilising a Rule-Based approach, rule base provided in Appendix B, circuit ends are able to be extracted from the protection database, schema provided in Appendix C, and the relevant models selected, configured and connected. This process involved the following steps and interactions:

1. Request list of devices present on circuit end and any non "default" operation times. For example:
   - MP1 - NONE => Default arrival verification only
   - MP2 - NONE => Default arrival verification only
   - TR - NONE => Default 10ms delay from input
   - CB1 - NONE => Default 40ms delay from input
   - DAR1 - 20s => Delay 20,000ms from input
   - CB2 - NONE => Default 40ms delay from input
   - DAR2 - 22s => Delay 22,000ms from input
2. Identify connectivity between devices and build skeleton for model - programmatic representation of Figure 4.24.
3. Request generic lisp models for each device on the circuit end, plus SCADA model.

4. Modify generic lisp models to contain correct timing modifications for non "default" device times. Utilising circuit information assign inputs/outputs the correct index for incoming alarms and attach SCADA models to allow the detection of missing alarms to be identified. (The effect of SCADA models can be seen in Figure 4.25) N.B. From this point MP models are replaced with SCADAMP models. This is due to fault inception information being unavailable and therefore the validation of the operation of MPs based on their inputs is also unavailable.

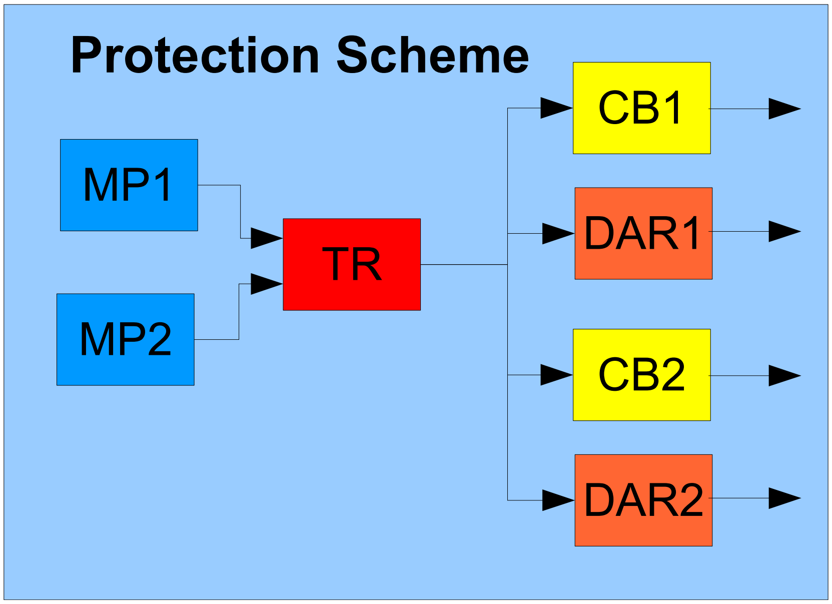


Figure 4.24: Integration Example

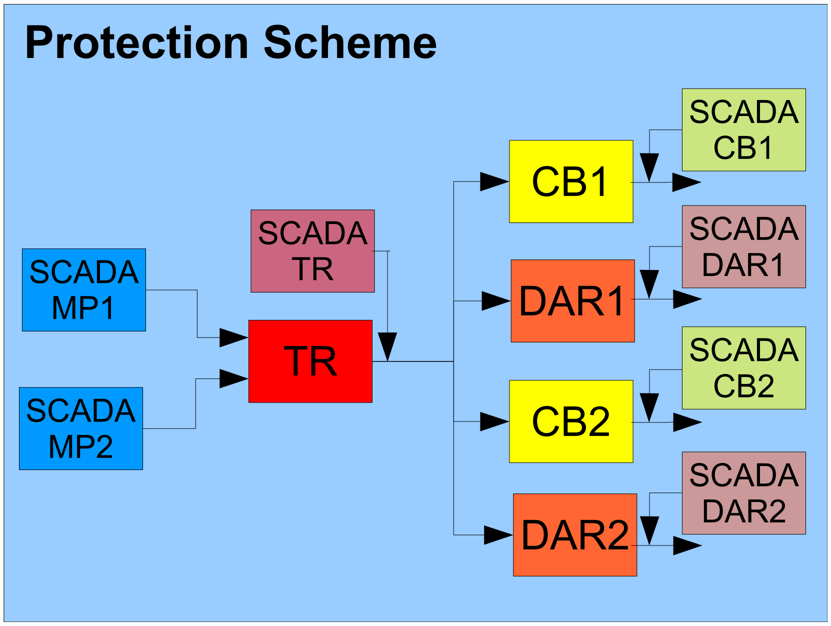


Figure 4.25: Integration Example with SCADA Alarms

Alarm Insertion and Protection Validation

Immediately following the construction of the protection scheme model, SCADA alarms are able to be inserted into the completed model.

With the arrival of alarms shown in table 4.20, the following steps are performed. The propagated values and assumptions at each stage are shown in Figure 4.26.

1. MP1 alarm arrives but cannot be propagated forwards until MP2 has arrived (This is because if MP2 actually arrived before 0ms then any following devices, e.g. TR or CB, would be expected to operate at a different time.)
    - No Conflicts Exist ⇒ No Diagnoses
2. MP2 alarm arrives and allows propagation.
    - No Conflicts Exist ⇒ No Diagnoses
3. Propagation of MP1/MP2 results in expected alarm at TR.
4. TR alarm arrives and creates conflict. (TR actual arrival 20ms, TR expected arrival 10ms)
    - Conflicts Exist {TR} (TR maloperated)
      ⇒ Minimal Diagnoses: {TR}
5. Propagation of TR results in expected alarms at CB1, DAR1, CB2 and DAR2.
    - Conflicts Exist {TR} (TR maloperated)
      ⇒ Minimal Diagnoses: {TR}
6. CB1 alarm arrives and confirms the propagation of the observed TR (at 20ms).
    - Conflicts Exist {TR} (TR maloperated)
      ⇒ Minimal Diagnoses: {TR}
7. CB2 alarm arrives and confirms the propagation of the observed TR (at 20ms).
    - Conflicts Exist {TR} (TR maloperated)
      ⇒ Minimal Diagnoses: {TR}
8. DAR1 alarm arrives and confirms the propagation of the of the observed TR (at 20ms)
    - Conflicts Exist {TR} (TR maloperated)
      ⇒ Minimal Diagnoses: {TR}
9. DAR2 alarm arrives and confirms the propagation of the of the observed TR (at 20ms)
    - Conflicts Exist {TR} (TR maloperated)
      ⇒ Minimal Diagnoses: {TR}

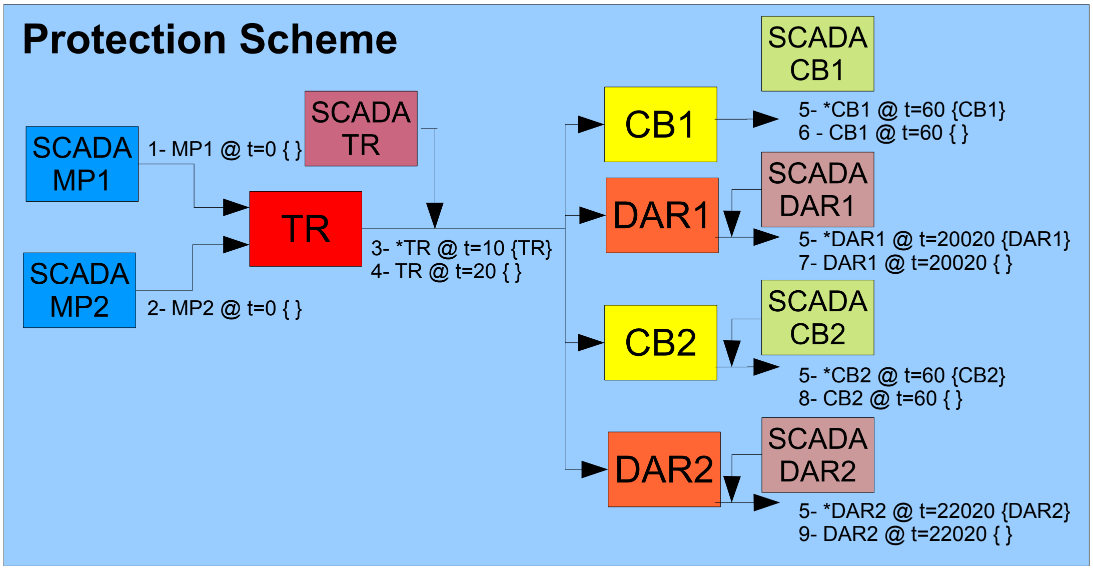


Figure 4.26: Protection Connectivity & Alarm Propagation for Integration Example

Therefore the final diagnosis is that the trip relay failed to operate as expected. In terms of what this diagnosis means, the trip relay failed to operate in a timely manner based on the inputs it received by the main protection devices. This potential diagnosis is then available for formatting and presentation to engineers.

If this incident had finished with a non-operating device an additional stage in this system would be the identification of missing alarms, where missing alarms are inserted into the appropriate "SCADA" model.

Incident Processing

At this point in the analysis process the following information is now able to be identified:

- *Disturbance Location* - Disturbance Occurred at 25/07/2012 06:59:51.250 on the line between SUBA and SUBB.
- *Incident Alarms* - Time ordered list of alarms both in real-time, e.g. 06:59:51.250 and Δt from trigger alarm.
- *Missing Alarms* - All expected alarms arrived.
- *Protection Validation* - The TR failed to operate as expected.

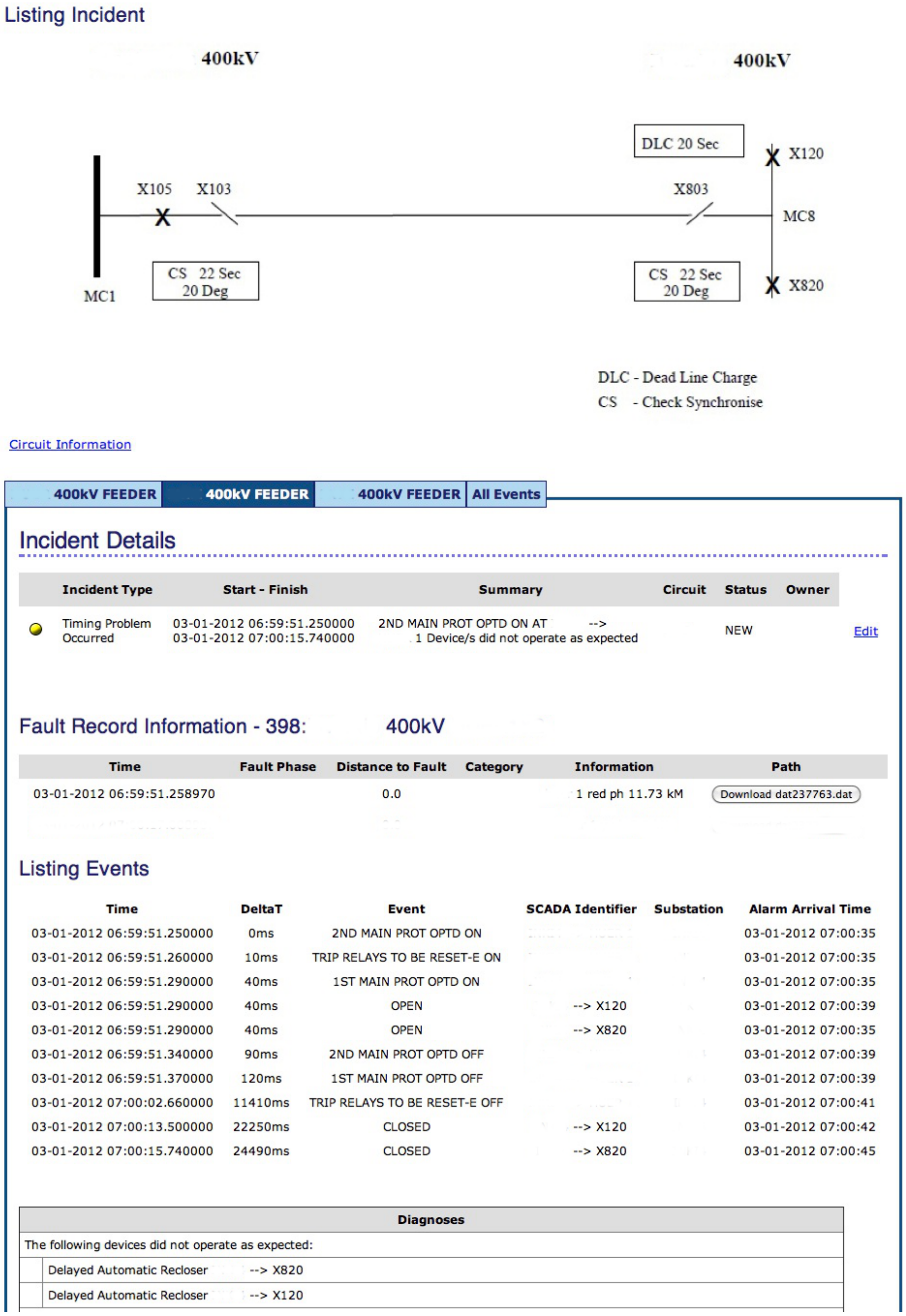
Utilising the rules within Appendix C.3 the following additional information is able to be extracted:

- *Incident Severity Summary* - Rule C.14 - Amber: Timing problems occurred.
- *Associated DFR?* - Rule C.15 - If one exists a link allowing users to open the record is created.

- *Incident Group* - Rule C.16 - If incidents have occurred simultaneously on surrounding circuits, combine incidents into an incident group for display to users.

This information including the circuit information associated with the faulted circuit, is then presented to users via the web front end.

An example demonstrating a potential fault sequence output can be seen in Figure 4.27.



Incident Details

	Incident Type	Start - Finish	Summary	Circuit	Status	Owner	
●	Timing Problem Occurred	03-01-2012 06:59:51.250000 03-01-2012 07:00:15.740000	2ND MAIN PROT OPTD ON AT --> 1 Device/s did not operate as expected		NEW		Edit

Fault Record Information - 398: 400kV

Time	Fault Phase	Distance to Fault	Category	Information	Path
03-01-2012 06:59:51.258970		0.0		1 red ph 11.73 kM	Download dat237763.dat

Listing Events

Time	DeltaT	Event	SCADA Identifier	Substation	Alarm Arrival Time
03-01-2012 06:59:51.250000	0ms	2ND MAIN PROT OPTD ON			03-01-2012 07:00:35
03-01-2012 06:59:51.260000	10ms	TRIP RELAYS TO BE RESET-E ON			03-01-2012 07:00:35
03-01-2012 06:59:51.290000	40ms	1ST MAIN PROT OPTD ON			03-01-2012 07:00:35
03-01-2012 06:59:51.290000	40ms	OPEN	--> X120		03-01-2012 07:00:39
03-01-2012 06:59:51.290000	40ms	OPEN	--> X820		03-01-2012 07:00:35
03-01-2012 06:59:51.340000	90ms	2ND MAIN PROT OPTD OFF			03-01-2012 07:00:39
03-01-2012 06:59:51.370000	120ms	1ST MAIN PROT OPTD OFF			03-01-2012 07:00:39
03-01-2012 07:00:02.660000	11410ms	TRIP RELAYS TO BE RESET-E OFF			03-01-2012 07:00:41
03-01-2012 07:00:13.500000	22250ms	CLOSED	--> X120		03-01-2012 07:00:42
03-01-2012 07:00:15.740000	24490ms	CLOSED	--> X820		03-01-2012 07:00:45

Diagnoses	
The following devices did not operate as expected:	
	Delayed Automatic Recloser --> X820
	Delayed Automatic Recloser --> X120

Figure 4.27: Web front End Screen Shot

4.7 Conclusion

This chapter has discussed the GDE method of diagnosis and the system PSDiagnosis and its functionality as a post-fault analysis system. The GDE has been demonstrated to provide the means of diagnosing multiple faults and through the extraction of diagnosis from the modelling system the ability to utilise interchangeable and configurable models. The PSDiagnosis system has demonstrated that the use of "generic" models combined with a maintainable database environment and the GDE can fulfill the diagnostic requirements of protection engineers. It can provide engineers with a close to real-time perspective of the network as well as more in-depth information regarding the operation of protection devices and the identification of further sources of information (DFR and TWL records). This provision has been demonstrated through case studies demonstrating its ability to correctly diagnose faults and its functionality from alarm collation to diagnosis. Further deployment case studies and results are discussed within Chapter 5.

Chapter 5

Deployment Case-Studies & Results

5.1 Introduction

The system discussed in this thesis has been deployed as a prototype system at ScottishPower Energy Networks. This chapter presents the findings from industrial deployment of the system, including case studies taken from actual incidents and faults occurring on the ScottishPower network. Throughout the requirements, design and deployment of the PSDiagnosis system, reliability as an accurate diagnosis system and flexibility for configuration have been key. This chapter discusses how well these needs were met during testing and deployment of the system.

5.2 Deployment Findings

As a method of testing the maintainability of the software, the database/web front end modules of PSDiagnosis were installed first. Installation was carried out in July 2011, with 6 months of consultation and refinement ensuring that the maintenance required for accurate analysis on system changes was able to be carried out easily and within the normal operations and procedures of engineers using the system. The system was demonstrated to be stable and user-friendly, able to maintain up-to-date network information and be accurate for the alarm processing system.

The alarm processing system has been installed as a prototype system, which carried out analysis on major historical incidents before going live. This included the analysis of over 150,000 SCADA alarms generated between August and September 2009, and high rate storm data from February 2010 (over 10,000 alarms and 30 circuit faults within 24 hours) and January 2012 (over 32,000 alarms and 70 circuit faults in 24 hours). Examination of the analysis results has been carried out with protection engineering experts at ScottishPower Energy Networks. The results from this examination demonstrate an incident identification rate of 100%, with device diagnosis positively identifying all major protection maloperations, with a minimal amount of false positives caused by inability to validate distance protection operation zones, 4% of 100 incidents but included the correct identification of a faulty distance protection relay. Benchmark analysis was carried out using January 2012 as a case study, this study presented a rate of 55 filtered alarms per minute and the identification of all 68 incidents over 24 hours in less than 45 minutes. The next section details industrial case studies carried out on data ranging from 2010 to 2012 and the results achieved.

5.3 Industrial Case-Studies

The following case studies present multi-ended incidents and their groupings during post-fault analysis. The following case-studies were utilised to validate and refine the analysis of incidents.

5.3.1 Case Study 1 - DAR Lockout

Case Study 1 involves an incident which affected 2 circuits ends and within 2 minutes caused DAR lockout. Lockout was not caused by a permanent fault but due to multiple faults occurring on the line within a short period of time, the circuit breakers were able to stay closed for over 2 seconds before the second fault was identified by main protection devices.

End 1 - DAR Failure

The relevant event alarms observed from End 1 in this case study can be seen in Table 5.1, with the network topology automatically configured for this circuit end in Figure 5.1.

Table 5.1: SCADA Alarms for Case Study 1 - End 1

SCADA Alarms			
Site Identifier	**Index**	**Alarm Name & Text**	**∆ T (ms)**
SUBA –> SUBB	MP1	1ST MAIN PROT OPTD ON	0
SUBA –> SUBB	TR	TRIP RELAY OPTD ON	10
SUBA –> CB1	CB1	CB1 OPEN	30

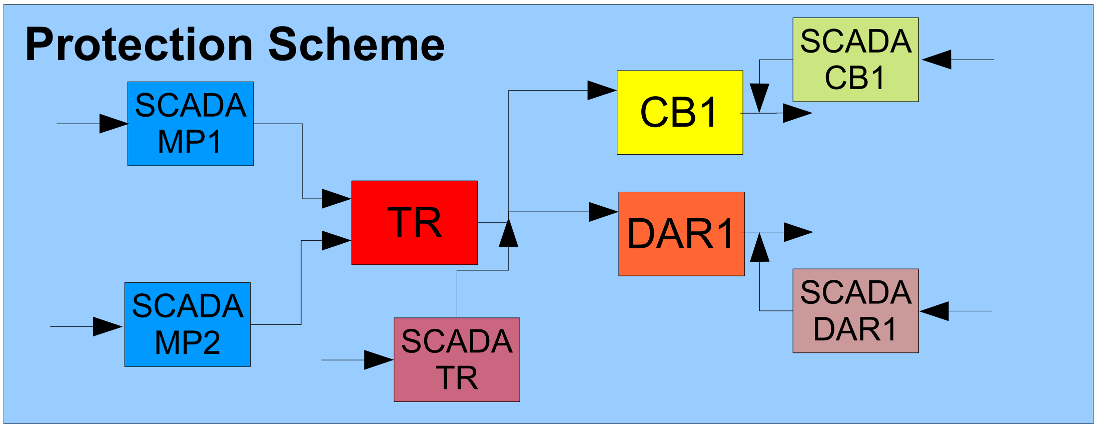


Figure 5.1: Protection Connectivity for Case Study 1 - End 1
N.B. Expected Alarm Propagation Times: TR - +10ms, CB - +40ms, DAR - +22,000ms

The propagated values and assumptions at each stage are shown in Figure 5.2. It should be noted that for the input to trip relays, a delay is created to ensure either both main protections have arrived or have been identified as missing. This ensures that an incorrect assumption is not propagated. For example if MP1 operates at 0ms, the trip relay would be expected at 10ms, however if MP2 actually operated at -10ms and has been delayed slightly then the trip relay would be expected at 0ms. Waiting for both alarms arrival or determined non-arrival prevents the potential propagation of incorrect values. With the arrival of alarms shown in table 5.1, the following steps are performed.

1. MP1 alarm arrives but cannot be propagated forwards until MP2 has arrived. (This is because if MP2 actually arrived before 0ms then any following devices, e.g. TR or CB, would be expected to operate at a different time.)
   - No Conflicts Exist ⇒ No Diagnoses
2. TR alarm arrives and is propagated.
   - No Conflicts Exist ⇒ No Diagnoses
3. Propagation of TR results in expected alarms at CB1 and DAR1.
4. CB1 alarm arrives and creates conflict.
   - Conflicts Exist {CB1} (CB1 maloperated)
     ⇒ Minimal Diagnoses: {CB1}
5. Propagation of CB1 results in expected alarm at TR confirming conflict.
6. Missing Alarm identified and inserted at SCADA MP2 and is propagated to MP2.
7. As MP2's non arrival is irrelevant to the operation of TR no further propagation is carried out.
8. Missing Alarm identified and inserted at SCADADAR1 and is propagated to DAR1.

9. As DAR1's arrival is expected this results in further conflict.

   - Conflicts Exist:
     - {CB1}
     - {DAR1, SCADADAR1}
   - CB1 maloperated and either DAR1 maloperated or DAR alarm missing
   - Minimal Diagnoses:
     - {CB1, DAR1}
     - {CB1, SCADADAR1}

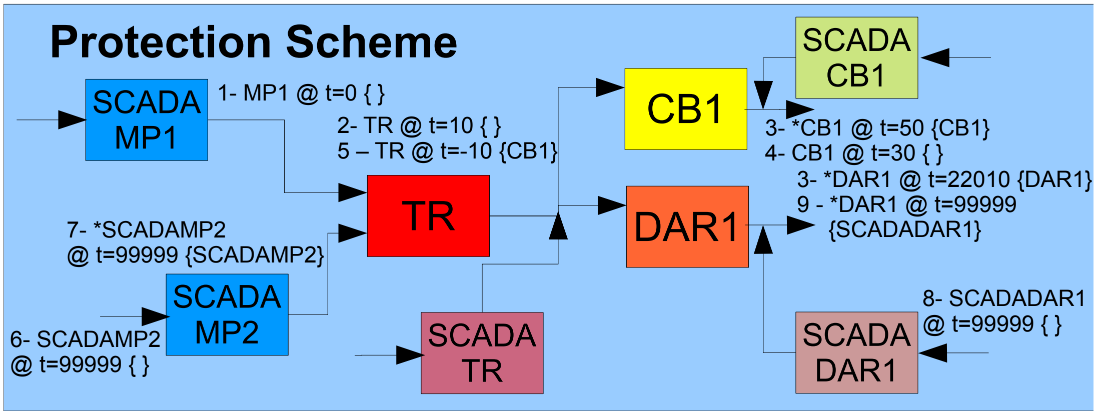
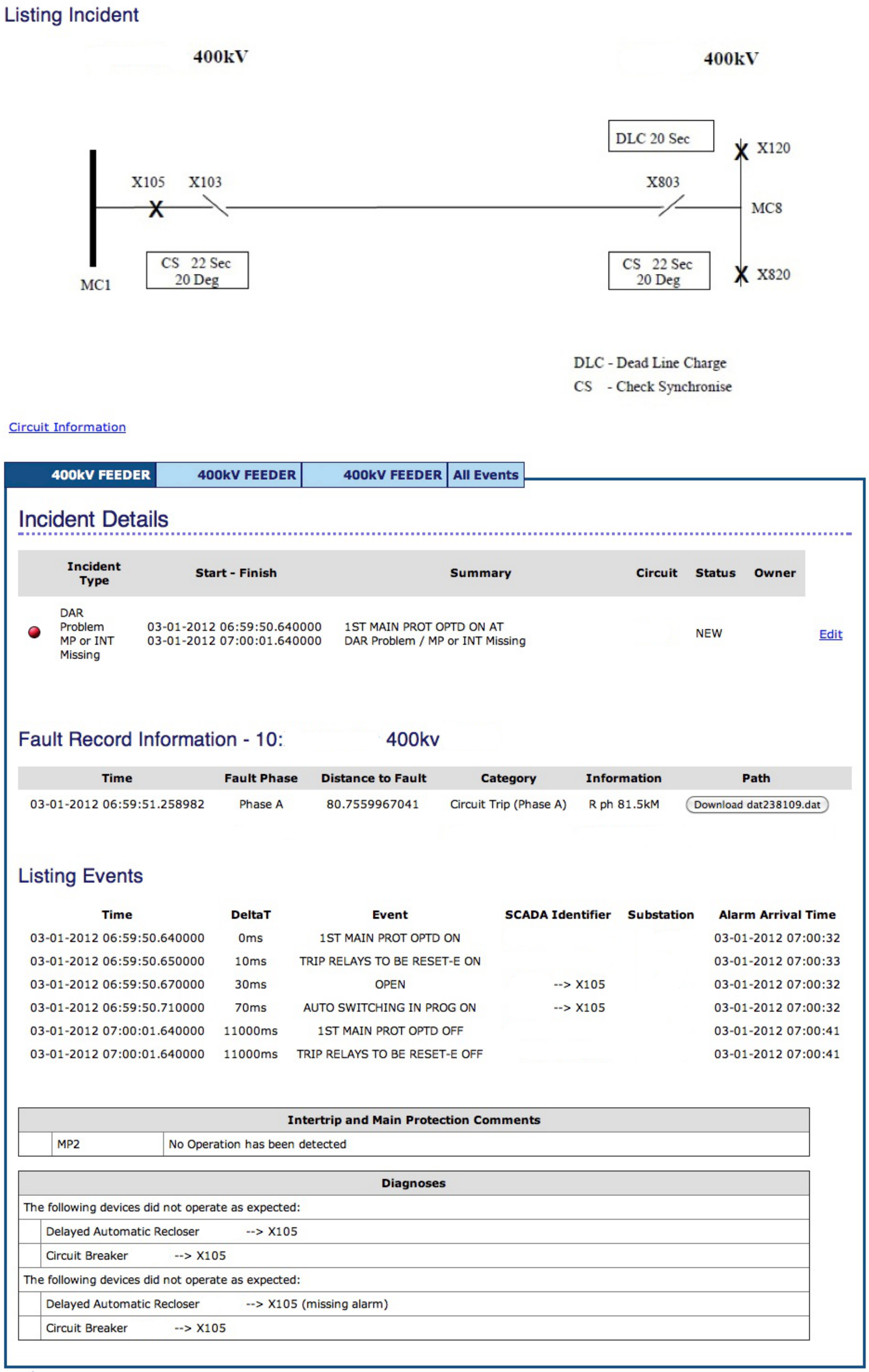


Figure 5.2: Protection Connectivity & Alarm Propagation for Case Study 1 - End 1
N.B. Expected Alarm Propagation Times: TR - +10ms, CB - +40ms, DAR - +22,000ms

Given this incident and final diagnoses the view presented to engineers for this circuit end can be seen in Figure 5.3. A set of rules is used to identify where a missing alarm did not cause observed problems but needs highlighted, in this situation MP2 did not operate as it is a distance protection and the fault was closer to the remote end of the line. Circuit breaker 1 is identified as inhibiting faulty behaviour due to its quick operation, although not an issue in terms of the protection of the network, this would be investigated as the circuit breaker should not physically have been able to operate this quickly.



Incident Details

	Incident Type	Start - Finish	Summary	Circuit	Status	Owner	
●	DAR Problem MP or INT Missing	03-01-2012 06:59:50.640000 03-01-2012 07:00:01.640000	1ST MAIN PROT OPTD ON AT DAR Problem / MP or INT Missing		NEW		Edit

Fault Record Information - 10: 400kv

Time	Fault Phase	Distance to Fault	Category	Information	Path
03-01-2012 06:59:51.258982	Phase A	80.7559967041	Circuit Trip (Phase A)	R ph 81.5kM	Download dat238109.dat

Listing Events

Time	DeltaT	Event	SCADA Identifier	Substation	Alarm Arrival Time
03-01-2012 06:59:50.640000	0ms	1ST MAIN PROT OPTD ON			03-01-2012 07:00:32
03-01-2012 06:59:50.650000	10ms	TRIP RELAYS TO BE RESET-E ON			03-01-2012 07:00:33
03-01-2012 06:59:50.670000	30ms	OPEN	--> X105		03-01-2012 07:00:32
03-01-2012 06:59:50.710000	70ms	AUTO SWITCHING IN PROG ON	--> X105		03-01-2012 07:00:32
03-01-2012 07:00:01.640000	11000ms	1ST MAIN PROT OPTD OFF			03-01-2012 07:00:41
03-01-2012 07:00:01.640000	11000ms	TRIP RELAYS TO BE RESET-E OFF			03-01-2012 07:00:41

Intertrip and Main Protection Comments		
	MP2	No Operation has been detected

Diagnoses	
The following devices did not operate as expected:	
	Delayed Automatic Recloser --> X105
	Circuit Breaker --> X105
The following devices did not operate as expected:	
	Delayed Automatic Recloser --> X105 (missing alarm)
	Circuit Breaker --> X105

Figure 5.3: Front End View for Case Study 1 - End 1 (DAR Lockout)

End 2 - DAR Lockout

The relevant event alarms observed from End 2 in this case study can be seen in Tables 5.2 and 5.3, with the network topology automatically configured for this circuit end in Figure 5.4. Due to the separation of events at this circuit end by the operation of both DARs the analysis of this incident is divided into two stages.

Table 5.2: SCADA Alarms for Case Study 1 - End 2 Pt 1

SCADA Alarms			
Site Identifier	**Index**	**Alarm Name & Text**	**Δ T (ms)**
SUBA –> SUBB	MP2	2ND MAIN PROT OPTD ON	0
SUBA –> SUBB	TR	TRIP RELAY OPTD ON	10
SUBA –> SUBB	MP1	1ST MAIN PROT OPTD ON	40
SUBA –> CB1	CB1	CB1 OPEN	40
SUBA –> CB2	CB2	CB2 OPEN	40
SUBA –> SUBB	MP1	1ST MAIN PROT OPTD OFF	90
SUBA –> SUBB	MP2	2ND MAIN PROT OPTD OFF	120
SUBA –> SUBB	TR	TRIP RELAY OPTD OFF	11410
SUBA –> CB1	DAR1	CB1 CLOSED	22250
SUBA –> CB2	DAR2	CB2 CLOSED	24490

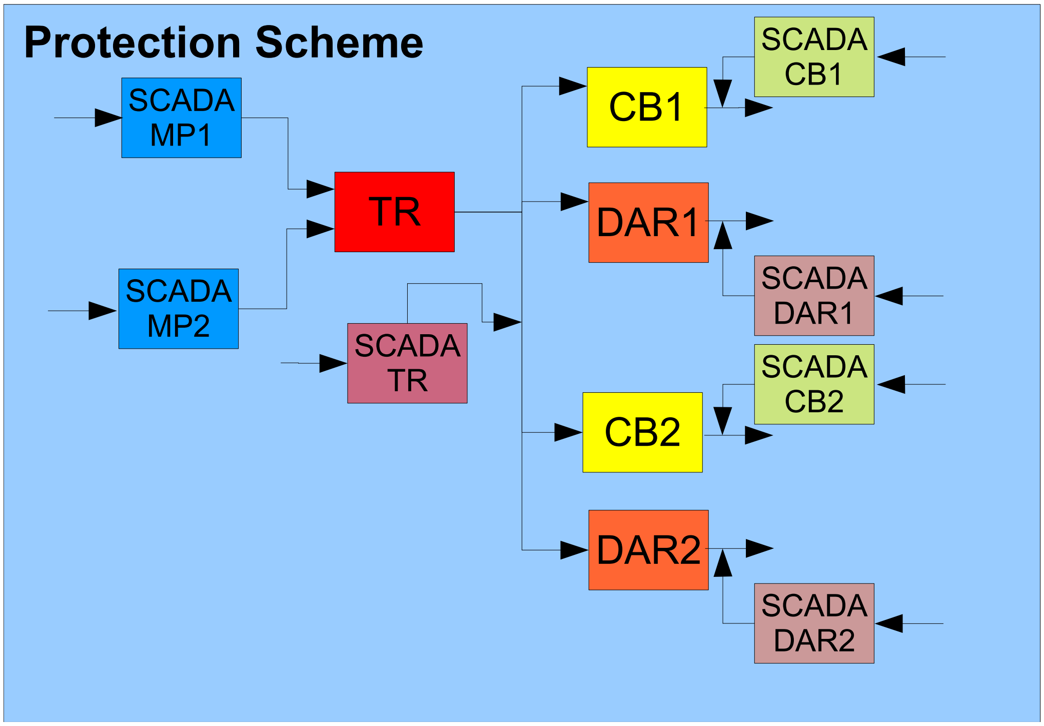


Figure 5.4: Protection Connectivity for Case Study 1 - End 2
N.B. Expected Alarm Propagation Times: TR - +0-10ms, CBs - +30ms, DAR1 (X120) - +20,000ms, DAR2 (X820) - +22,000ms

Incident 1 With the arrival of alarms shown in table 5.2, the following steps are performed. The propagated values and assumptions at each stage are shown in Figure 5.5

1. MP2 alarm arrives but cannot be propagated forwards until MP1 has arrived. (This is because if MP1 actually arrived before 0ms then the trip relay would be expected to operate at a different time)
   - No Conflicts Exist ⇒ No Diagnoses
2. TR alarm arrives and is propagated.
   - No Conflicts Exist ⇒ No Diagnoses
3. Propagation of TR results in expected alarms at CB1, CB2, DAR1 and DAR2.
4. MP1 alarm arrives and confirms TR operation.
   - No Conflicts Exist ⇒ No Diagnoses
5. CB1 alarm arrives.
6. CB1 is propagated back to TR confirming correct operation from current observations.
   - No Conflicts Exist ⇒ No Diagnoses
7. CB2 alarm arrives.

8. CB2 is propagated back to TR confirming correct operation from current observations.
   - No Conflicts Exist ⇒ No Diagnoses
9. DAR1 alarm arrives and creates conflict
   - Conflicts Exist {DAR1} (DAR1 maloperated)
   - Minimal Diagnoses: {DAR1}
10. DAR2 alarm arrives and creates conflict
    - Conflicts Exist:
      - {DAR1}
      - {DAR2}
    - DAR1 and DAR2 maloperated
    - Minimal Diagnoses: {DAR1, DAR2}

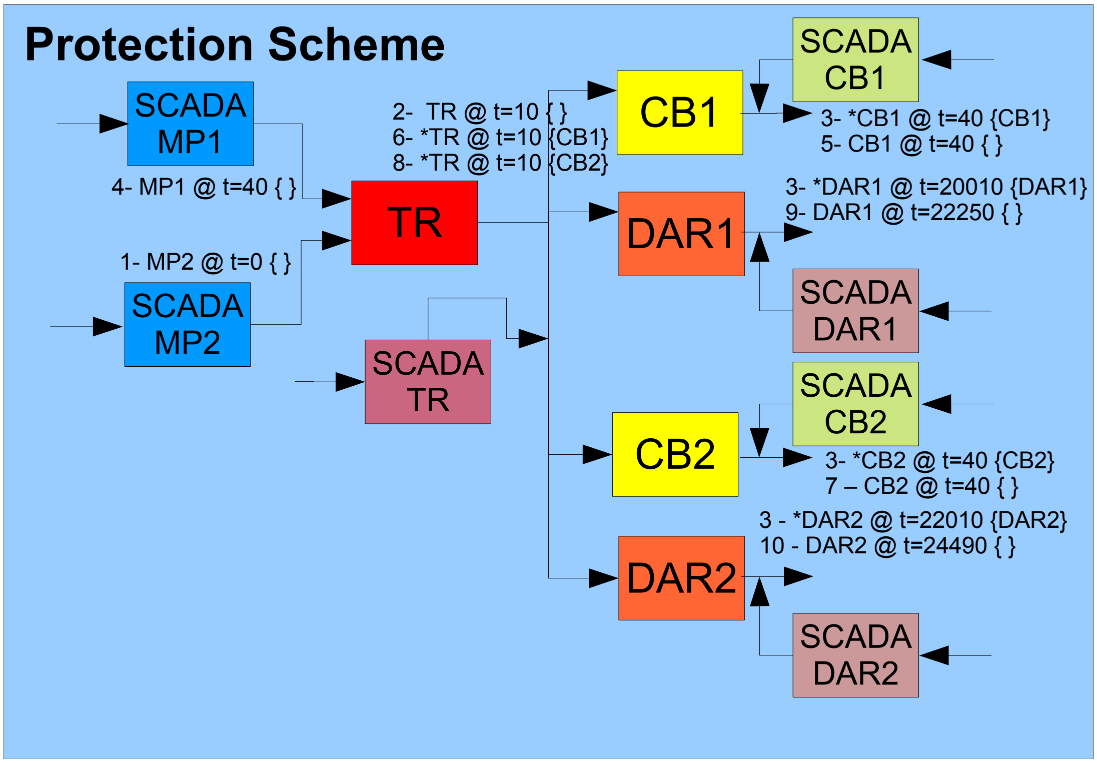
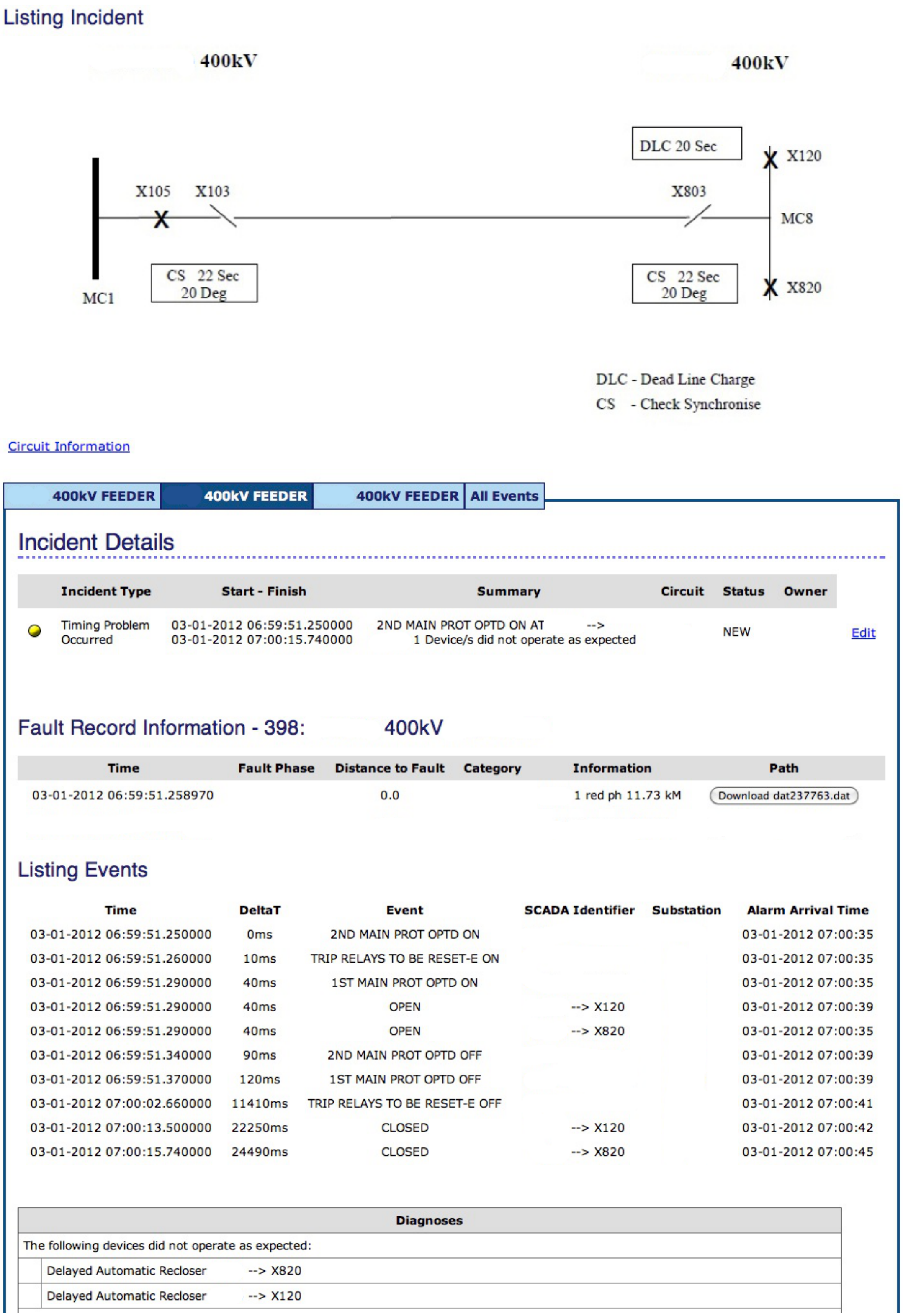


Figure 5.5: Protection Connectivity & Alarm Propagation for Case Study 1 - End 2 Pt 1
N.B. Expected Alarm Propagation Times: TR - +0-10ms, CBs - +30ms, DAR1 (X120) - +20,000ms, DAR2 (X820) - +22,000ms

Given this incident and final diagnoses the view presented to engineers for this circuit end can be seen in Figure 5.6.



Listing Incident

Circuit Information

400kV FEEDER | 400kV FEEDER | 400kV FEEDER | All Events

Incident Details

	Incident Type	Start - Finish	Summary	Circuit	Status	Owner	
●	Timing Problem Occurred	03-01-2012 06:59:51.250000 03-01-2012 07:00:15.740000	2ND MAIN PROT OPTD ON AT --> 1 Device/s did not operate as expected		NEW		Edit

Fault Record Information - 398: 400kV

Time	Fault Phase	Distance to Fault	Category	Information	Path
03-01-2012 06:59:51.258970		0.0		1 red ph 11.73 kM	Download dat237763.dat

Listing Events

Time	DeltaT	Event	SCADA Identifier	Substation	Alarm Arrival Time
03-01-2012 06:59:51.250000	0ms	2ND MAIN PROT OPTD ON			03-01-2012 07:00:35
03-01-2012 06:59:51.260000	10ms	TRIP RELAYS TO BE RESET-E ON			03-01-2012 07:00:35
03-01-2012 06:59:51.290000	40ms	1ST MAIN PROT OPTD ON			03-01-2012 07:00:35
03-01-2012 06:59:51.290000	40ms	OPEN	--> X120		03-01-2012 07:00:39
03-01-2012 06:59:51.290000	40ms	OPEN	--> X820		03-01-2012 07:00:35
03-01-2012 06:59:51.340000	90ms	2ND MAIN PROT OPTD OFF			03-01-2012 07:00:39
03-01-2012 06:59:51.370000	120ms	1ST MAIN PROT OPTD OFF			03-01-2012 07:00:39
03-01-2012 07:00:02.660000	11410ms	TRIP RELAYS TO BE RESET-E OFF			03-01-2012 07:00:41
03-01-2012 07:00:13.500000	22250ms	CLOSED	--> X120		03-01-2012 07:00:42
03-01-2012 07:00:15.740000	24490ms	CLOSED	--> X820		03-01-2012 07:00:45

Diagnoses	
The following devices did not operate as expected:	
Delayed Automatic Recloser	--> X820
Delayed Automatic Recloser	--> X120

Figure 5.6: Front End View for Case Study 1 - End 2 Pt 1 (Slow DAR operation)

Table 5.3: SCADA Alarms for Case Study 1 - End 2 Pt 2

SCADA Alarms			
Site Identifier	**Index**	**Alarm Name & Text**	**∆ T (ms)**
SUBA –> SUBB	TR	TRIP RELAY OPTD ON	0
SUBA –> SUBB	MP2	2ND MAIN PROT OPTD ON	0
SUBA –> CB2	CB2	CB2 OPEN	30
SUBA –> CB1	CB1	CB1 OPEN	30
SUBA –> SUBB	MP1	1ST MAIN PROT OPTD ON	30
SUBA –> SUBB	MP2	2ND MAIN PROT OPTD OFF	80
SUBA –> SUBB	MP1	1ST MAIN PROT OPTD OFF	100
SUBA –> CB2	DAR2	CB2 CLOSE LOCKOUT ON	230
SUBA –> CB1	DAR1	CB1 CLOSE LOCKOUT ON	240
SUBA –> SUBB	TR	TRIP RELAY OPTD OFF	11510
SUBA –> CB1	DAR1	CB1 CLOSE LOCKOUT OFF	93100
SUBA –> CB2	DAR2	CB2 CLOSE LOCKOUT OFF	100780

Incident 2 With the arrival of alarms shown in table 5.3, the following steps are performed. It should be noted that the order of arrival of TR and MP2 although physically impossible as MP2 triggered TR, due to the order of insertion into the database they are inserted into the analysis system in this order. The propagated values and assumptions at each stage are shown in Figure 5.7.

1. TR alarm arrives and is propagated.
2. Propagation of TR results in expected alarms at CB1, CB2, DAR1 and DAR2.
3. MP2 alarm arrives but cannot be propagated forwards until MP1 has arrived. (This is because if MP1 actually arrived before 0ms then the trip relay would be expected to operate at a different time)
   - No Conflicts Exist ⇒ No Diagnoses
4. CB1 alarm arrives.
5. CB1 is propagated back to TR confirming correct operation from current observations.
   - No Conflicts Exist ⇒ No Diagnoses
6. CB2 alarm arrives.
7. CB2 is propagated back to TR confirming correct operation from current observations.
   - No Conflicts Exist ⇒ No Diagnoses
8. MP1 alarm arrives and confirms TR operation on MP2.
   - No Conflicts Exist ⇒ No Diagnoses

9. Missing Alarm identified and inserted at SCADADAR1 and is propagated to DAR1.

10. As DAR1's arrival is expected this results in conflict.

    - Conflicts Exist {DAR1, SCADADAR1} (Either DAR1 or SCADADAR1 (i.e. DAR alarm missing) maloperated)

    - Minimal Diagnoses: {DAR1}{SCADADAR1}

11. Missing Alarm identified and inserted at SCADADAR2 and is propagated to DAR2.

12. As DAR2's arrival is expected this results in conflict.

    - Conflicts Exist:

      – {DAR1, SCADADAR1}

      – {DAR2, SCADADAR2}

    - Either DAR1 or SCADADAR1 (i.e. DAR alarm missing) and either DAR2 or SCADADAR2 maloperated

    - Minimal Diagnoses:

      – {DAR1, DAR2}

      – {SCADADAR1, SCADADAR2}

      – {DAR1, SCADADAR2}

      – {SCADADAR1, DAR2}

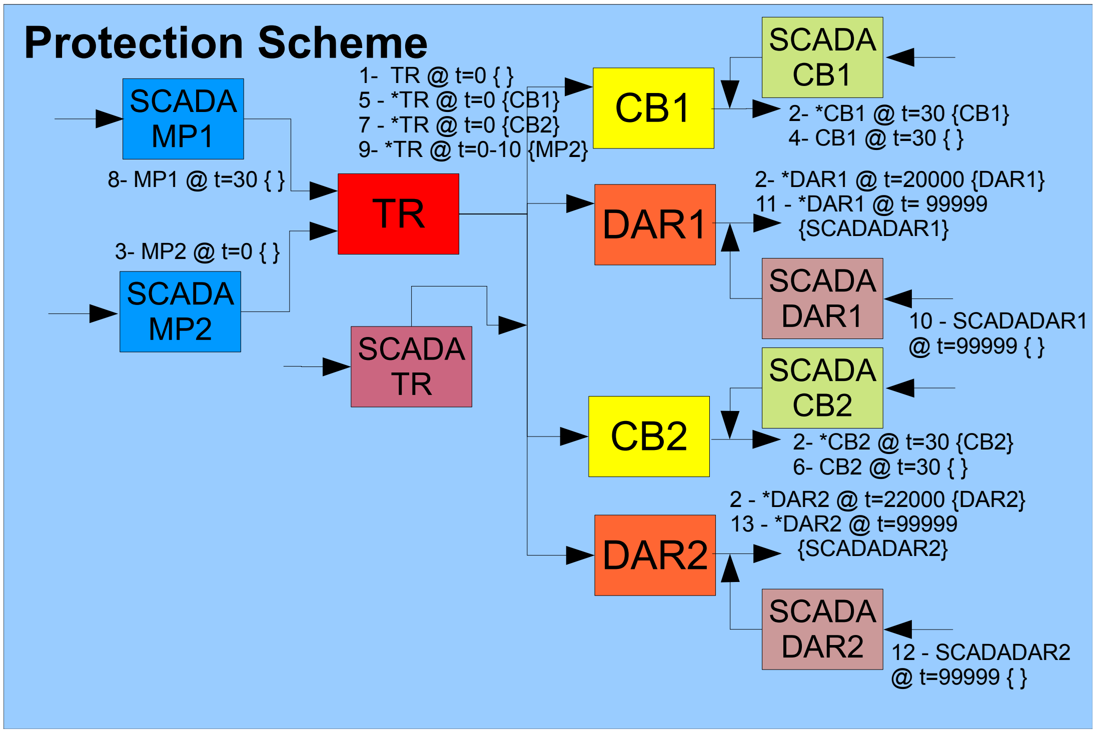
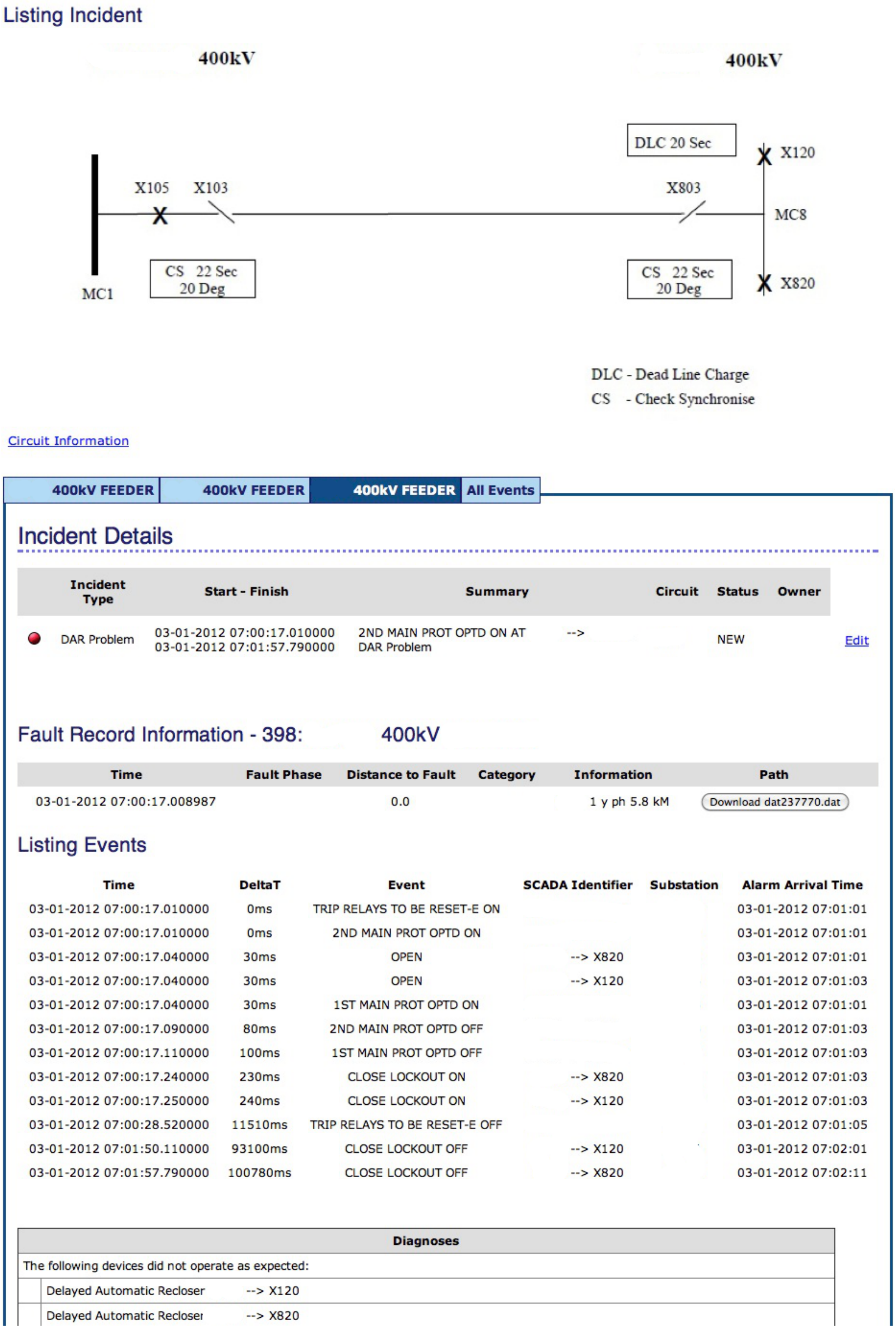


Figure 5.7: Protection Connectivity & Alarm Propagation for Case Study 1 - End 2 Pt 2
N.B. Expected Alarm Propagation Times: TR - +0-10ms, CBs - +30ms, DAR1 (X120) - +20,000ms, DAR2 (X820) - +22,000ms

Given this incident and final diagnoses the view presented to engineers for this circuit end can be seen in Figure 5.8. Lockout alarms although not analysed are grouped with this incident allowing engineers to identify that although lockout occurred, the lockout was reset after 93 and 100 seconds of incident start and therefore the potential for manual reclosure is available.



Listing Incident

Incident Details

	Incident Type	Start - Finish	Summary	Circuit	Status	Owner	
●	DAR Problem	03-01-2012 07:00:17.010000 03-01-2012 07:01:57.790000	2ND MAIN PROT OPTD ON AT DAR Problem	-->	NEW		Edit

Fault Record Information - 398: 400kV

Time	Fault Phase	Distance to Fault	Category	Information	Path
03-01-2012 07:00:17.008987		0.0		1 y ph 5.8 kM	Download dat237770.dat

Listing Events

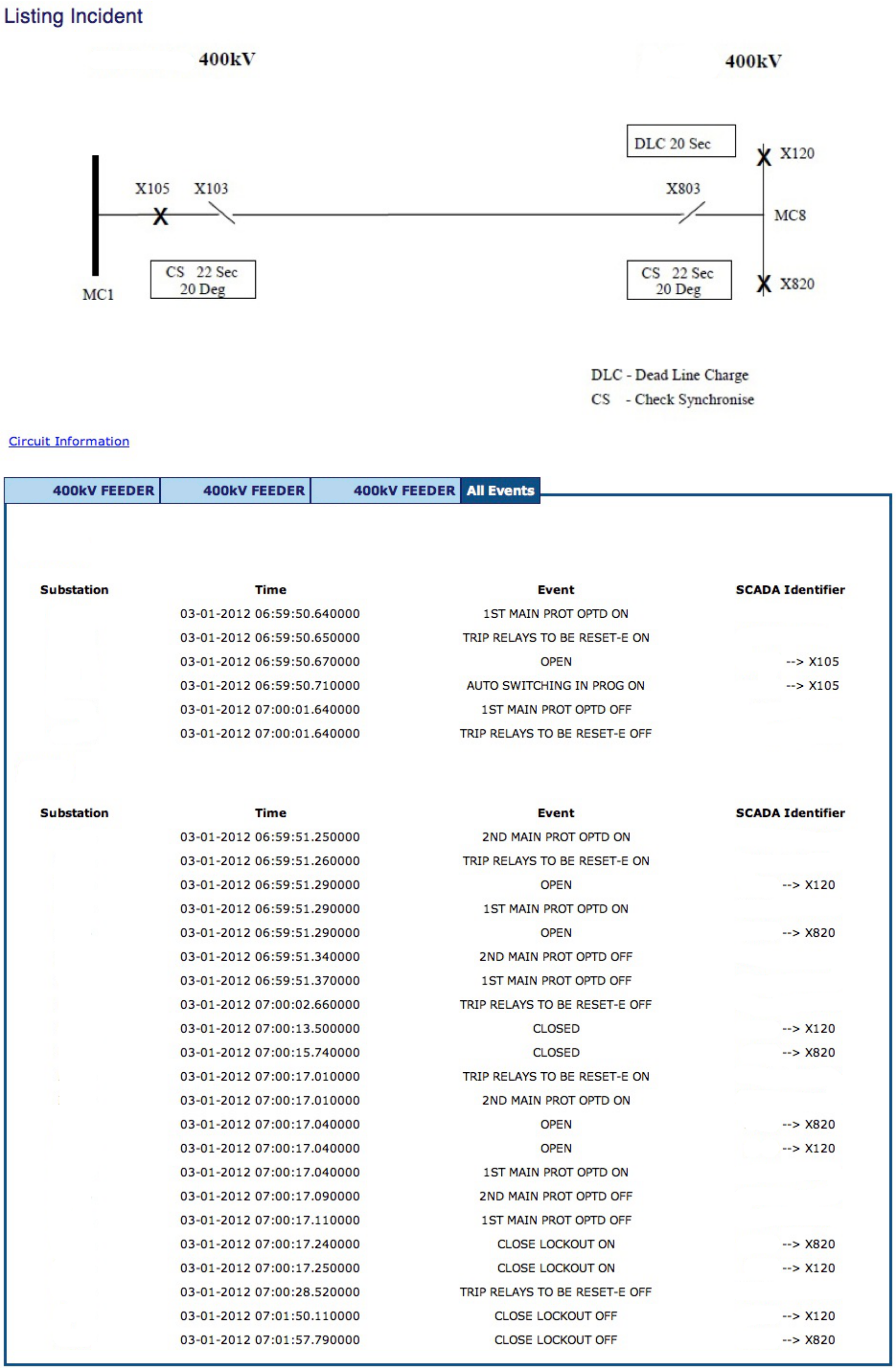
Time	DeltaT	Event	SCADA Identifier	Substation	Alarm Arrival Time
03-01-2012 07:00:17.010000	0ms	TRIP RELAYS TO BE RESET-E ON			03-01-2012 07:01:01
03-01-2012 07:00:17.010000	0ms	2ND MAIN PROT OPTD ON			03-01-2012 07:01:01
03-01-2012 07:00:17.040000	30ms	OPEN	--> X820		03-01-2012 07:01:01
03-01-2012 07:00:17.040000	30ms	OPEN	--> X120		03-01-2012 07:01:03
03-01-2012 07:00:17.040000	30ms	1ST MAIN PROT OPTD ON			03-01-2012 07:01:01
03-01-2012 07:00:17.090000	80ms	2ND MAIN PROT OPTD OFF			03-01-2012 07:01:03
03-01-2012 07:00:17.110000	100ms	1ST MAIN PROT OPTD OFF			03-01-2012 07:01:03
03-01-2012 07:00:17.240000	230ms	CLOSE LOCKOUT ON	--> X820		03-01-2012 07:01:03
03-01-2012 07:00:17.250000	240ms	CLOSE LOCKOUT ON	--> X120		03-01-2012 07:01:03
03-01-2012 07:00:28.520000	11510ms	TRIP RELAYS TO BE RESET-E OFF			03-01-2012 07:01:05
03-01-2012 07:01:50.110000	93100ms	CLOSE LOCKOUT OFF	--> X120		03-01-2012 07:02:01
03-01-2012 07:01:57.790000	100780ms	CLOSE LOCKOUT OFF	--> X820		03-01-2012 07:02:11

Diagnoses	
The following devices did not operate as expected:	
Delayed Automatic Recloser	--> X120
Delayed Automatic Recloser	--> X820

Figure 5.8: Front End View for Case Study 1 - End 1 Pt 2(DAR Lockout)

All Alarms

The system is also able to present engineers with a chronological list of alarms which have been grouped into an incident. For this case study the "All Events" page is shown in Figure 5.9.



Substation	Time	Event	SCADA Identifier
	03-01-2012 06:59:50.640000	1ST MAIN PROT OPTD ON	
	03-01-2012 06:59:50.650000	TRIP RELAYS TO BE RESET-E ON	
	03-01-2012 06:59:50.670000	OPEN	--> X105
	03-01-2012 06:59:50.710000	AUTO SWITCHING IN PROG ON	--> X105
	03-01-2012 07:00:01.640000	1ST MAIN PROT OPTD OFF	
	03-01-2012 07:00:01.640000	TRIP RELAYS TO BE RESET-E OFF	

Substation	Time	Event	SCADA Identifier
	03-01-2012 06:59:51.250000	2ND MAIN PROT OPTD ON	
	03-01-2012 06:59:51.260000	TRIP RELAYS TO BE RESET-E ON	
	03-01-2012 06:59:51.290000	OPEN	--> X120
	03-01-2012 06:59:51.290000	1ST MAIN PROT OPTD ON	
	03-01-2012 06:59:51.290000	OPEN	--> X820
	03-01-2012 06:59:51.340000	2ND MAIN PROT OPTD OFF	
	03-01-2012 06:59:51.370000	1ST MAIN PROT OPTD OFF	
	03-01-2012 07:00:02.660000	TRIP RELAYS TO BE RESET-E OFF	
	03-01-2012 07:00:13.500000	CLOSED	--> X120
	03-01-2012 07:00:15.740000	CLOSED	--> X820
	03-01-2012 07:00:17.010000	TRIP RELAYS TO BE RESET-E ON	
	03-01-2012 07:00:17.010000	2ND MAIN PROT OPTD ON	
	03-01-2012 07:00:17.040000	OPEN	--> X820
	03-01-2012 07:00:17.040000	OPEN	--> X120
	03-01-2012 07:00:17.040000	1ST MAIN PROT OPTD ON	
	03-01-2012 07:00:17.090000	2ND MAIN PROT OPTD OFF	
	03-01-2012 07:00:17.110000	1ST MAIN PROT OPTD OFF	
	03-01-2012 07:00:17.240000	CLOSE LOCKOUT ON	--> X820
	03-01-2012 07:00:17.250000	CLOSE LOCKOUT ON	--> X120
	03-01-2012 07:00:28.520000	TRIP RELAYS TO BE RESET-E OFF	
	03-01-2012 07:01:50.110000	CLOSE LOCKOUT OFF	--> X120
	03-01-2012 07:01:57.790000	CLOSE LOCKOUT OFF	--> X820

Figure 5.9: SCADA Alarms for Case Study 1

5.3.2 Case Study 2 - Protection Problems

Case Study 2 involves a incident which affected 2 circuits ends but involved a complicated communications problem. This case study presents one side of line where an intertrip is interrupted due to communications problems and the circuit breaker fails to operate as quickly as expected.

The relevant event alarms observed in this case study can be seen in Table 5.4, with the network topology automatically configured for this circuit end in Figure 5.10.

Table 5.4: SCADA Alarms for Case Study 2

SCADA Alarms			
Site Identifier	**Index**	**Alarm Name & Text**	**Δ T (ms)**
SUBA –> SUBB	MP1	1ST MAIN PROT OPTD ON	0
SUBA –> SUBB	TR	TRIP RELAY OPTD ON	10
SUBA –> CB1	CB1	CB1 OPEN	70
SUBA –> SUBB	MP1	1ST MAIN PROT OPTD OFF	100
SUBA –> SUBB	DAR1	AUTOSWITCHING IN PROG ON	140
SUBA –> SUBB	TR	TRIP RELAY OPTD OFF	10060
SUBA –> CB1	DAR1	CB1 CLOSED	22250
SUBA –> SUBB	DAR1	AUTOSWITCHING COMPLETE ON	29420
SUBA –> SUBB	DAR1	AUTOSWITCHING IN PROG OFF	29420
SUBA –> SUBB	DAR1	AUTOSWITCHING COMPLETE OFF	31420

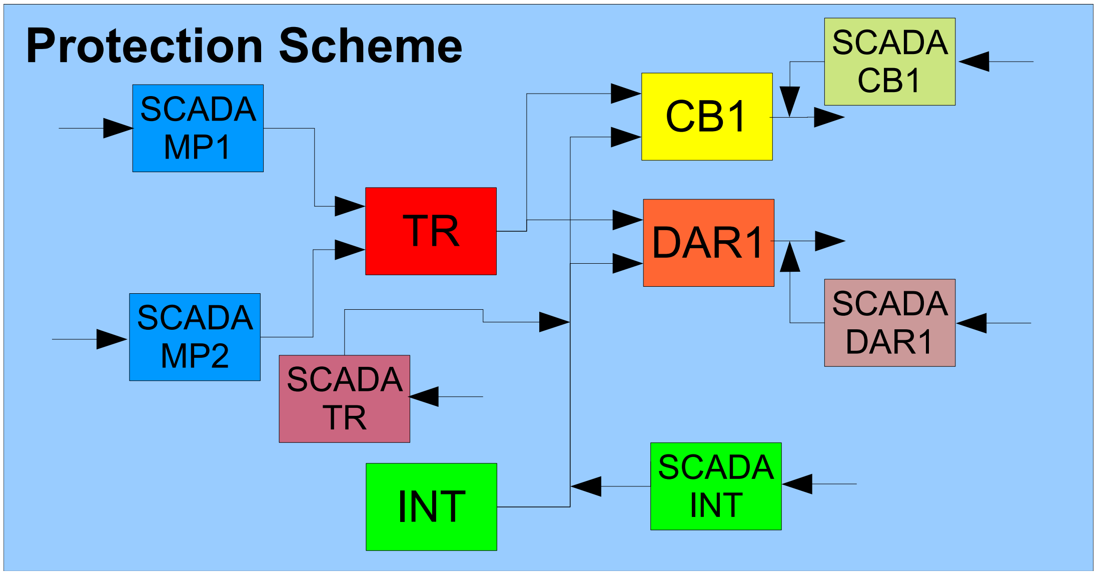


Figure 5.10: Protection Connectivity for Case Study 2
N.B. Expected Alarm Propagation Times: TR - +10ms, CB - +40ms, DAR - +27,000ms

With the arrival of alarms shown in table 5.4, the following steps are performed. The propagated values and assumptions at each stage are shown in Figure 5.11

1. MP1 alarm arrives but cannot be propagated forwards until MP2 has arrived. (This is because if MP2 actually arrived before 0ms then the trip relay would be expected to operate at a different time)
   - No Conflicts Exist ⇒ No Diagnoses
2. TR alarm arrives but cannot be propagated forwards until INT1 has arrived. (This is because if INT1 actually arrived before 10ms then the circuit breaker would be expected to operate at a different time)
   - No Conflicts Exist ⇒ No Diagnoses
3. CB1 alarm arrives but cannot be propagated/validated until INT1 has arrived. (This is because if INT1 actually arrived before TR then the circuit breaker would be expected to operate at a different time)
   - No Conflicts Exist ⇒ No Diagnoses
4. DAR1 alarm arrives but cannot be propagated/validated until INT1 has arrived. (This is because if INT1 actually arrived before TR then the DAR would be expected to operate at a different time)
   - No Conflicts Exist ⇒ No Diagnoses
5. Missing Alarm identified and inserted at SCADA MP2 and is propagated to MP2.
6. As MP2's non arrival is irrelevant to the operation of TR no further propagation is carried out.
7. Missing Alarm identified and inserted at SCADAINT1 and is propagated to INT1 then to CB1, DAR1 and TR respectively.
8. As INT1's arrival is expected and identified conflict, the following is deduced.
   - Conflicts Exist {CB1, SCADAINT1} (Either CB1 maloperated or SCADAINT1 (i.e. INT1 alarm missing) occurred)
   - Minimal Diagnoses: {CB1}{SCADAINT1}

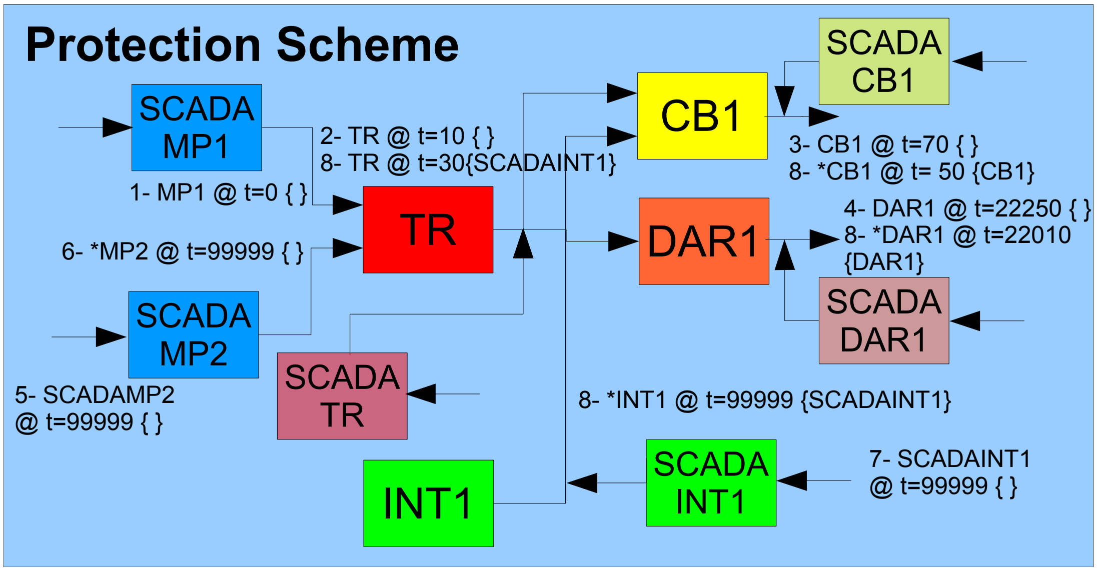
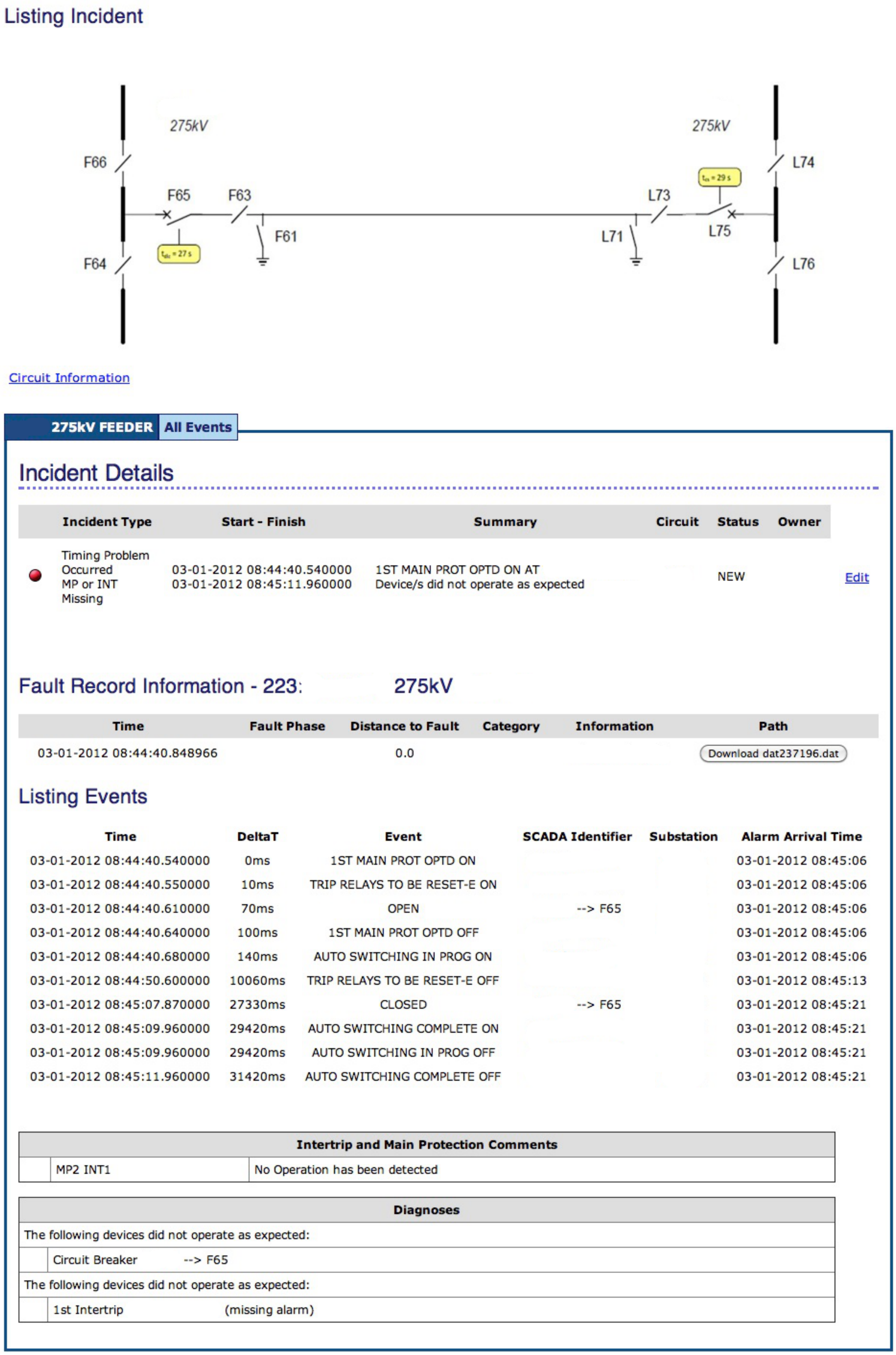


Figure 5.11: Protection Connectivity & Alarm Propagation for Case Study 2
N.B. Expected Alarm Propagation Times: TR - +10ms, CB - +40ms, DAR - +21,000-23,000ms

Given this incident and final diagnoses the view presented to engineers for this circuit end can be seen in Figure 5.12. A set of rules is used to identify where a missing alarm did not cause observed problems but needs highlighted, in this situation MP2 did not operate as it is a distance protection and the fault was closer to the remote end of the line.



Listing Incident

Circuit Information

275kV FEEDER | All Events

Incident Details

	Incident Type	Start - Finish	Summary	Circuit	Status	Owner	
●	Timing Problem Occurred MP or INT Missing	03-01-2012 08:44:40.540000 03-01-2012 08:45:11.960000	1ST MAIN PROT OPTD ON AT Device/s did not operate as expected		NEW		Edit

Fault Record Information - 223: 275kV

Time	Fault Phase	Distance to Fault	Category	Information	Path
03-01-2012 08:44:40.848966		0.0			Download dat237196.dat

Listing Events

Time	DeltaT	Event	SCADA Identifier	Substation	Alarm Arrival Time
03-01-2012 08:44:40.540000	0ms	1ST MAIN PROT OPTD ON			03-01-2012 08:45:06
03-01-2012 08:44:40.550000	10ms	TRIP RELAYS TO BE RESET-E ON			03-01-2012 08:45:06
03-01-2012 08:44:40.610000	70ms	OPEN	--> F65		03-01-2012 08:45:06
03-01-2012 08:44:40.640000	100ms	1ST MAIN PROT OPTD OFF			03-01-2012 08:45:06
03-01-2012 08:44:40.680000	140ms	AUTO SWITCHING IN PROG ON			03-01-2012 08:45:06
03-01-2012 08:44:50.600000	10060ms	TRIP RELAYS TO BE RESET-E OFF			03-01-2012 08:45:13
03-01-2012 08:45:07.870000	27330ms	CLOSED	--> F65		03-01-2012 08:45:21
03-01-2012 08:45:09.960000	29420ms	AUTO SWITCHING COMPLETE ON			03-01-2012 08:45:21
03-01-2012 08:45:09.960000	29420ms	AUTO SWITCHING IN PROG OFF			03-01-2012 08:45:21
03-01-2012 08:45:11.960000	31420ms	AUTO SWITCHING COMPLETE OFF			03-01-2012 08:45:21

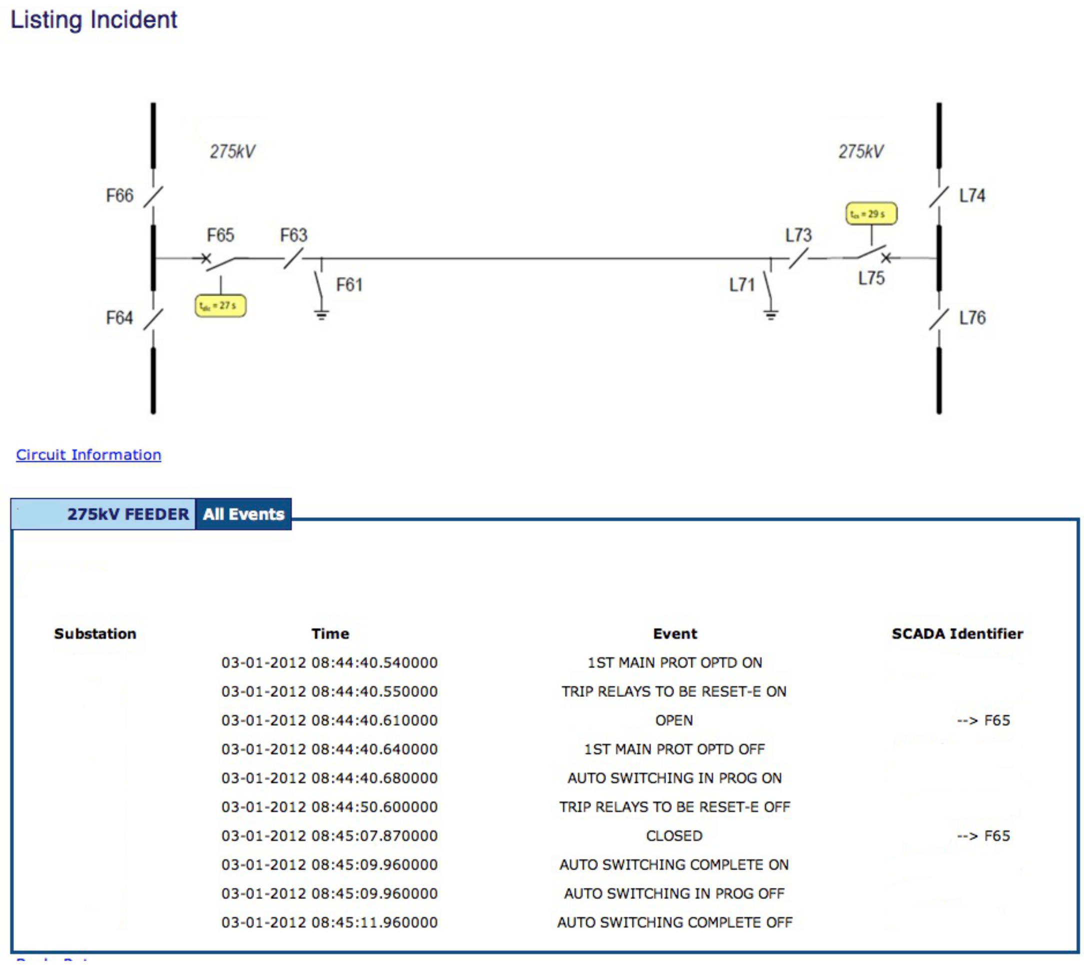
Intertrip and Main Protection Comments		
	MP2 INT1	No Operation has been detected

Diagnoses	
The following devices did not operate as expected:	
	Circuit Breaker --> F65
The following devices did not operate as expected:	
	1st Intertrip (missing alarm)

Figure 5.12: Front End View for Case Study 2 (Timing Problem)

All Alarms

The system is also able to present engineers with a chronological list of alarms which have been grouped into an incident. For this case study the “All Events” page is shown in Figure 5.13.



Substation	Time	Event	SCADA Identifier
	03-01-2012 08:44:40.540000	1ST MAIN PROT OPTD ON	
	03-01-2012 08:44:40.550000	TRIP RELAYS TO BE RESET-E ON	
	03-01-2012 08:44:40.610000	OPEN	--> F65
	03-01-2012 08:44:40.640000	1ST MAIN PROT OPTD OFF	
	03-01-2012 08:44:40.680000	AUTO SWITCHING IN PROG ON	
	03-01-2012 08:44:50.600000	TRIP RELAYS TO BE RESET-E OFF	
	03-01-2012 08:45:07.870000	CLOSED	--> F65
	03-01-2012 08:45:09.960000	AUTO SWITCHING COMPLETE ON	
	03-01-2012 08:45:09.960000	AUTO SWITCHING IN PROG OFF	
	03-01-2012 08:45:11.960000	AUTO SWITCHING COMPLETE OFF	

Figure 5.13: SCADA Alarms for Case Study 2

5.3.3 Case Study 3 - Successful Operation

Case Study 3 involves an incident which affected 2 circuits ends, but due to communication problems the incidents appeared separated by such a time as to be grouped into different high level incidents. This case study presents one side of a line where the protection operates successfully and is able, regardless of communications issues, to reclose the circuit breaker and return the line to service within 16 seconds.

The relevant event alarms observed in this case study can be seen in Table 5.5, with the network topology automatically configured for this circuit end in Figure 5.14.

Table 5.5: SCADA Alarms for Case Study 3

SCADA Alarms			
Site Identifier	**Index**	**Alarm Name & Text**	**Δ T (ms)**
SUBA –> SUBB	INT1	INTERTRIP1 RECEIVE OPTD ON	0
SUBA –> SUBB	TR	TRIP RELAY OPTD ON	0
SUBA –> SUBB	INT2	INTERTRIP2 RECEIVE OPTD ON	0
SUBA –> CB1	CB1	CB1 OPEN	50
SUBA –> SUBB	DAR1	AUTOSWITCHING IN PROG ON	140
SUBA –> SUBB	INT1	INTERTRIP1 RECEIVE OPTD OFF	10300
SUBA –> SUBB	INT2	INTERTRIP2 RECEIVE OPTD OFF	10480
SUBA –> SUBB	TR	TRIP RELAY OPTD OFF	10460
SUBA –> SUBB	DAR1	AUTOSWITCHING IN PROG OFF	10480
SUBA –> SUBB	DAR1	AUTOSWITCHING COMPLETE ON	15310
SUBA –> CB1	DAR1	CB1 CLOSED	15350
SUBA –> SUBB	DAR1	AUTOSWITCHING COMPLETE OFF	15490

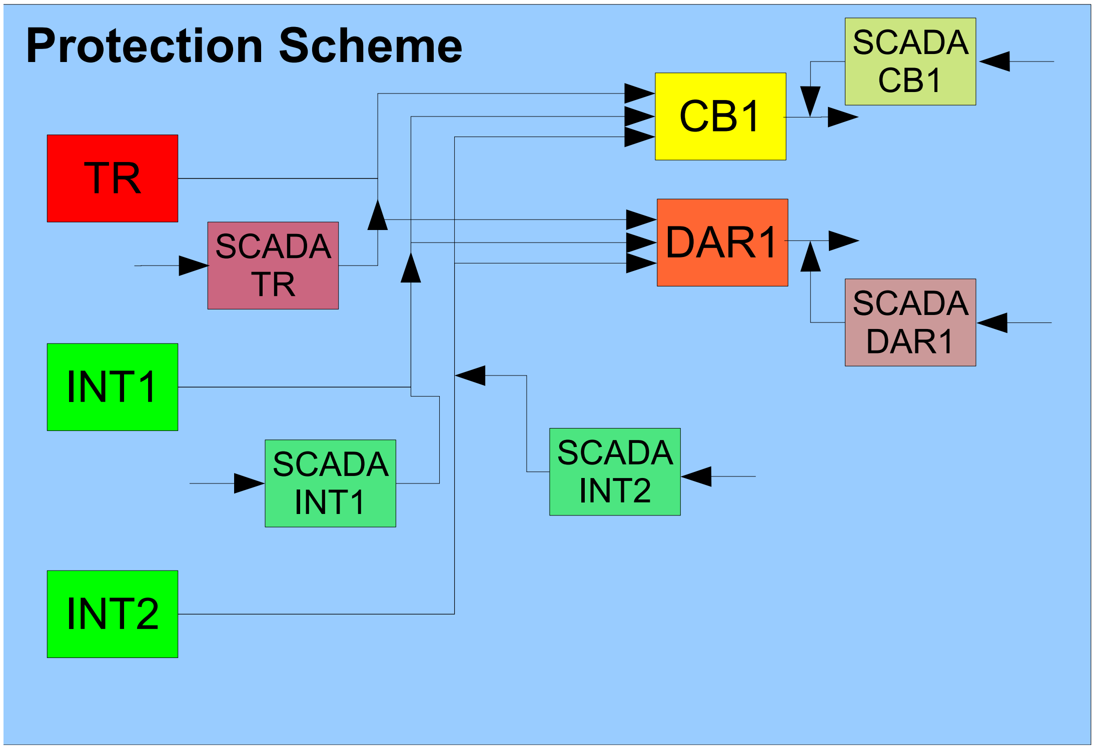


Figure 5.14: Protection Connectivity for Case Study 3
N.B. Expected Alarm Propagation Times: TR - +10ms, CB - +50ms, INT1-2 +-30ms DAR - +15,000ms

With the arrival of alarms shown in table 5.5, the following steps are performed. The propagated values and assumptions at each stage are shown in Figure 5.15

1. INT1 alarm arrives but cannot be propagated towards CB1 until TR and INT2 have arrived. (This is because if TR or INT2 actually arrived before 0ms then the circuit breaker would be expected to operate at a different time)
2. INT1 can be propagated to INT2
    - No Conflicts Exist ⇒ No Diagnoses
3. INT2 alarm arrives but cannot be propagated towards CB1 until TR has arrived. (This is because if TR actually arrived before 0ms then the circuit breaker would be expected to operate at a different time)
4. INT2 can be propagated to INT1
    - No Conflicts Exist ⇒ No Diagnoses
5. TR alarm arrives.
6. With the arrival of TR, alarms can now be propagated to CB1 and DAR1.
    - No Conflicts Exist ⇒ No Diagnoses
7. CB1 alarm arrives and confirms correct operation.
    - No Conflicts Exist ⇒ No Diagnoses
8. DAR1 alarm arrives and confirms correct operation.
    - No Conflicts Exist ⇒ No Diagnoses

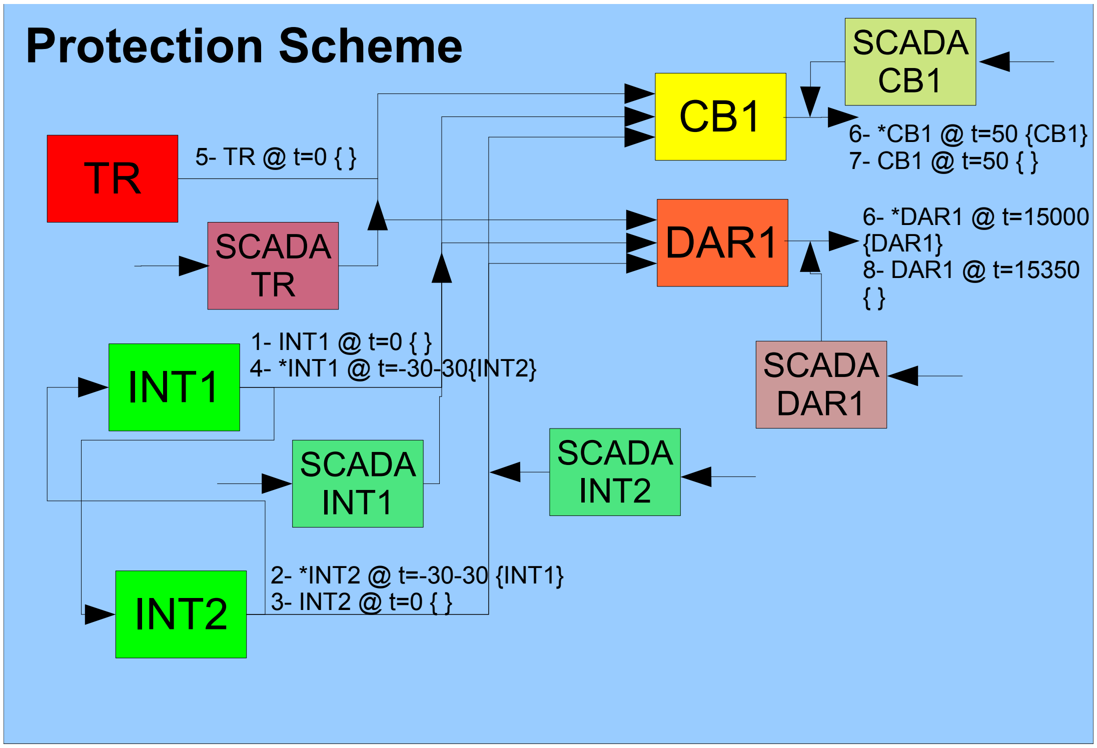
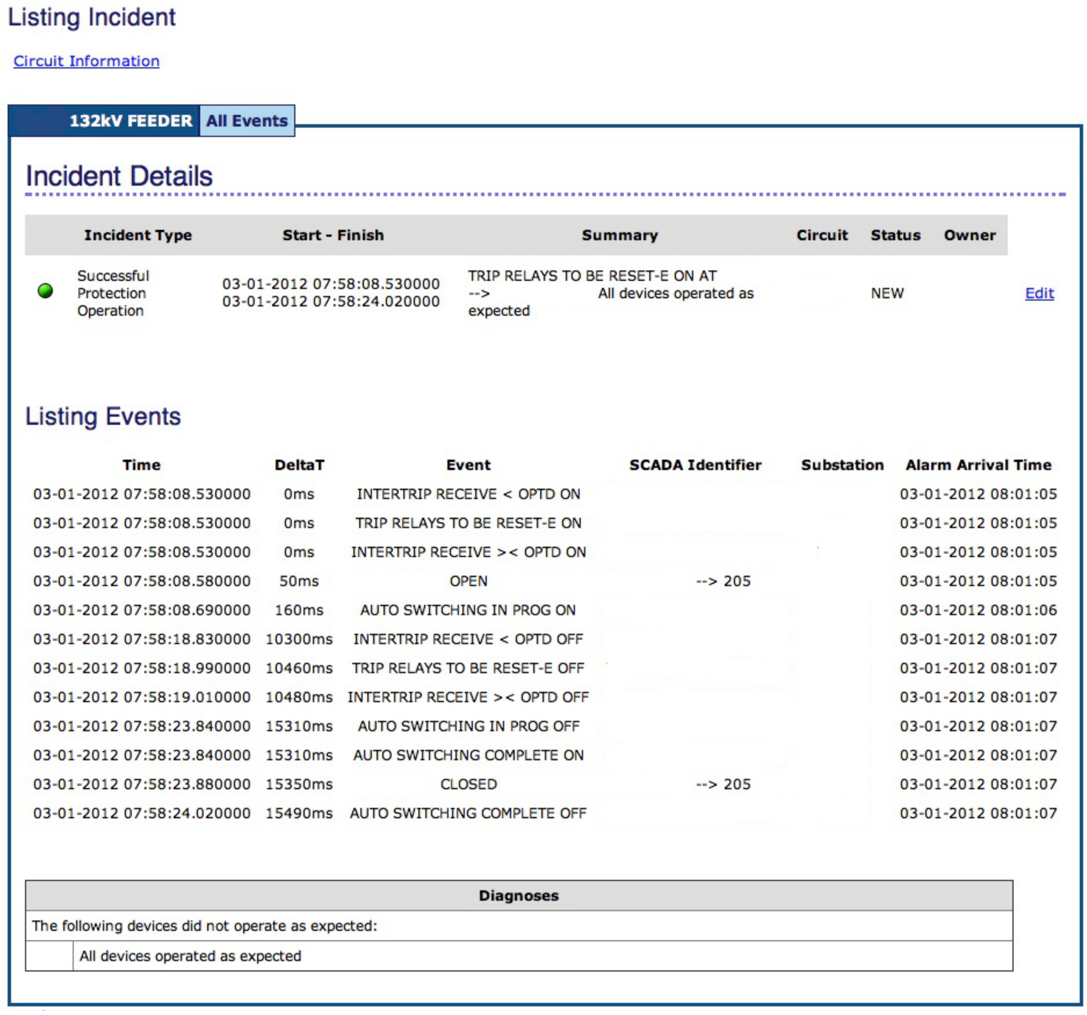


Figure 5.15: Protection Connectivity & Alarm Propagation for Case Study 3
N.B. Expected Alarm Propagation Times: TR - +10ms, CB - +50ms, INT1-2 +-30ms DAR - +15,000ms

Given this incident and final diagnoses the view presented to engineers for this circuit end can be seen in Figure 5.16.



Listing Incident

Circuit Information

Incident Details

	Incident Type	Start - Finish	Summary	Circuit	Status	Owner	
●	Successful Protection Operation	03-01-2012 07:58:08.530000 03-01-2012 07:58:24.020000	TRIP RELAYS TO BE RESET-E ON AT --> All devices operated as expected		NEW		Edit

Listing Events

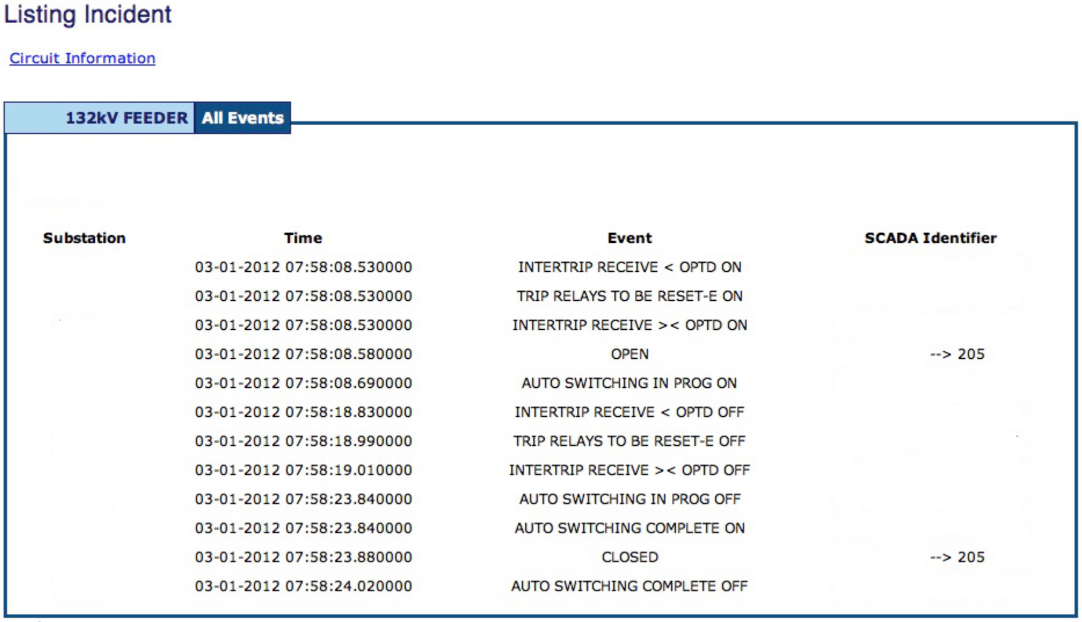
Time	DeltaT	Event	SCADA Identifier	Substation	Alarm Arrival Time
03-01-2012 07:58:08.530000	0ms	INTERTRIP RECEIVE < OPTD ON			03-01-2012 08:01:05
03-01-2012 07:58:08.530000	0ms	TRIP RELAYS TO BE RESET-E ON			03-01-2012 08:01:05
03-01-2012 07:58:08.530000	0ms	INTERTRIP RECEIVE >< OPTD ON			03-01-2012 08:01:05
03-01-2012 07:58:08.580000	50ms	OPEN	--> 205		03-01-2012 08:01:05
03-01-2012 07:58:08.690000	160ms	AUTO SWITCHING IN PROG ON			03-01-2012 08:01:06
03-01-2012 07:58:18.830000	10300ms	INTERTRIP RECEIVE < OPTD OFF			03-01-2012 08:01:07
03-01-2012 07:58:18.990000	10460ms	TRIP RELAYS TO BE RESET-E OFF			03-01-2012 08:01:07
03-01-2012 07:58:19.010000	10480ms	INTERTRIP RECEIVE >< OPTD OFF			03-01-2012 08:01:07
03-01-2012 07:58:23.840000	15310ms	AUTO SWITCHING IN PROG OFF			03-01-2012 08:01:07
03-01-2012 07:58:23.840000	15310ms	AUTO SWITCHING COMPLETE ON			03-01-2012 08:01:07
03-01-2012 07:58:23.880000	15350ms	CLOSED	--> 205		03-01-2012 08:01:07
03-01-2012 07:58:24.020000	15490ms	AUTO SWITCHING COMPLETE OFF			03-01-2012 08:01:07

Diagnoses	
The following devices did not operate as expected:	
	All devices operated as expected

Figure 5.16: Front End View for Case Study 3 (Operated as Expected)

All Alarms

The system is also able to present engineers with a chronological list of alarms which have been grouped into an incident. For this case study the “All Events” page is shown in Figure 5.17.



Listing Incident

Circuit Information

Substation	Time	Event	SCADA Identifier
	03-01-2012 07:58:08.530000	INTERTRIP RECEIVE < OPTD ON	
	03-01-2012 07:58:08.530000	TRIP RELAYS TO BE RESET-E ON	
	03-01-2012 07:58:08.530000	INTERTRIP RECEIVE >< OPTD ON	
	03-01-2012 07:58:08.580000	OPEN	--> 205
	03-01-2012 07:58:08.690000	AUTO SWITCHING IN PROG ON	
	03-01-2012 07:58:18.830000	INTERTRIP RECEIVE < OPTD OFF	
	03-01-2012 07:58:18.990000	TRIP RELAYS TO BE RESET-E OFF	
	03-01-2012 07:58:19.010000	INTERTRIP RECEIVE >< OPTD OFF	
	03-01-2012 07:58:23.840000	AUTO SWITCHING IN PROG OFF	
	03-01-2012 07:58:23.840000	AUTO SWITCHING COMPLETE ON	
	03-01-2012 07:58:23.880000	CLOSED	--> 205
	03-01-2012 07:58:24.020000	AUTO SWITCHING COMPLETE OFF	

Figure 5.17: SCADA Alarms for Case Study 3

Chapter 6

Conclusions & Further Work

6.1 Conclusions

The provision of accurate and timely post-fault incident identification and protection validation can both reduce the pressures on engineers and save money for utilities as a whole. This automated analysis system, utilising a model-based approach, provides engineers with the information available from the analysis of SCADA alarms. This information allows informed decisions to be made during fault conditions which can both reduce CML and protect expensive equipment from damage. The information also allows prioritisation of events allowing engineers to tackle incidents in an efficient manner. During the research involved in this thesis the tasks for an intelligent post-fault analysis system were as follows:

- *Determine disturbance location*
- *Analyse "out of order" alarms*
- *Filter irrelevant alarms*
- *Group alarms into incidents*
- *Identify missing alarms*
- *Validate the performance of protection equipment*
- *Minimise Maintenance Overhead*
- *Universal Access*

As demonstrated in chapter 3, the problem of post-fault analysis and SCADA alarm processing has been widely documented and a wide range of approaches considered. However the following gaps were identified with the provision for alarm processing systems:

- Inability to identify unknown or multiple faults occurring in the protection network

- Inability to adapt to network updates and different configurations
- Carry out analysis purely on slower to arrive but higher fidelity information

The system presented in this thesis is able to meet the requirements provided by engineers by utilising the GDE method of diagnosis to carry out accurate analysis and backed by a network database and intranet based front end to provide a low maintenance solution. By utilising the GDE method the following advantages are available over the use of other knowledge and model based approaches:

- *Diagnose Multiple Faults* - able to identify potential occurrences of both single and multiple faults on a protection scheme.
- *No Fault Models Required* - able to identify the occurrence of novel or previously unseen faults without the need for fault models, while providing the ability to add fault models for additional diagnostic information.
- *Easily Configurable Models* - able to utilise the separation of models and diagnosis to create a configurable library of protection device models.
- *No Requirement for Ordered Alarms* - able to handle alarms without the need to wait to achieve chronological order.
- *Respond to Missing Alarms* - able to identify potential missing alarms without interfering with genuine protection fault diagnosis.

Chapters 4 and 5 detail a number of utility case-studies to highlight the performance of the post-fault analysis system both based on accuracy, timely performance and its industrial use. It was therefore proven that the system is capable of carrying out accurate analysis on a number of protection scheme configurations and malfunctions (missing alarm/s, untimely operation, failed operation) and have both the stability and ease of maintenance required for industrial deployment.

The contribution to knowledge in the field of post-fault incident identification and protection validation can therefore be summarised as:

- Adapting the GDE to handle temporal discrete events on dynamically generated models and identify and diagnose missing alarms and temporal tolerances.
- The research, design, development and implementation of a low maintenance post-fault incident identification and protection validation solution, which is able to accurately and within a timely manner support engineers in the management of protection equipment and the power network during faults.
- The translation of expert knowledge and protection information into a protection database allowing central access and updating capabilities to engineers. Thus maintaining a flexible link between SCADA data and protection scheme configurations, including the potential for IEC 61850 integration.

- The examination of industrially deployed system performance based on incident identification, classification, faulted component selection and timely operation.

The system discussed in this thesis is an industrially working system providing engineers with a first pass diagnosis on SCADA based data and collating further information in the form of DFR data whenever it becomes available. The industrial importance of this work can be summarised into the following:

- Accurate protection validation system providing diagnostic information within seconds of network events.
- Provides a single point of access to retrieve both SCADA information associated with a network incident which will provide access to DFR data when it becomes available. Therefore allowing the system to be enhanced when DFR data arrival times reduce.
- Provides engineers a simple location to keep and maintain network topology information in an easily accessable and configurable manner.

6.2 Further Work

The following sections identify potential extensions or enhancements to the PSDiagnosis system. Although the system is currently in use there is still a wide range of further research options available to complement this work in the future.

6.2.1 Analysis and Integration of Further Data Sources

By utilising the protection database along with the model-based SCADA alarm processor the analysis of further data sources can be completed. The present system although capable of identifying available DFR and TWL data, currently cannot carry out automated analysis of these data sources. However, many of the functions available within the current PSDiagnosis system would be able to be directly linked to the analysis of DFR records, for example the digital operations stored within the records. Therefore, allowing reuse of the PSDiagnosis models to provide more accurate protection diagnosis and include models for main protection devices as fault inception timings and distance to fault information will be available.

6.2.2 Further Diagnostics from DFR Data

With the integration of further data sources, the ability to extract more detailed analysis could become available. Some of these details are suggested in this section.

Circuit Breaker Condition Monitoring

An avenue for further research is in the extraction of circuit breaker condition from DFR data. Trip coil analysis and calculations such as I^2t could be extracted and accumulated for each circuit breaker connected to a DFR. This information could allow the condition of circuit breakers to both be monitored over time and provide better maintenance schedules automatically tuned for each breaker. By carrying out this type of mainteance, analysis is able to be done utilising currently available data therefore providing a means of "free" condition monitoring of circuit breakers. This type of analysis is open for futher development especially in the area of condition based maitenance which at present are in the early stages of development, particularly with newer types of switchgear becoming more commonly used and the load implications associated with interconnected series compensation. By combining the number of operations, currently available within PSDiagnosis, with the type of operations and collating this information with device failures could save significant volumes of money both on maintenance but also on the necessary isolations during maintenance.

VT Diagnostics

Through previous research and development within the Institute of Energy and Environment, University of Strathclyde, Glasgow [Davidson et al., 2008] it was found that through the analysis of DFR data potential problems with VT (Voltage Transformers) can be identified. With further research this identification may be able to be refined to identify particular fault signatures which can then be presented to engineers during automated DFR analysis.

6.2.3 IEC 61850 Integration

Although IEC 61850 devices are still in the minority compared to the older legacy network devices, the ability to merge these devices into the analysis system would be ideal for a future proof system. Therefore further work would involve the use of a IEC 61850 case study section of the network which could be integrated into the system and tested in conjunction with the legacy devices.

Appendix A

Default Lisp Models

This appendix details the raw Lisp based models built for the model based system. The default values inserted are used where devices expected timings are unavailable.

A.1 Trip Relay

A.1.1 One Input

```
(constraint one-tr-component ((a cell) (b cell) (ok assumption))
 (formulae (b (a ok) (if (alarm-p a)
                         (setq b (make-alarm :device "TR ON" :time-ms (+ (alarm-time-ms a) 10)))
                         :dismiss))
           (a (b ok) (if (alarm-p b)
                         (setq a (make-alarm :device "MAIN PROT OPTD" :time-ms (- (alarm-time-ms b) 10)))
                         :dismiss))))
```

A.1.2 Two Input

```
(constraint two-tr-component ((a cell) (b cell) (c cell) (ok assumption))
 (formulae (c (a b ok) (if (and (alarm-p a) (alarm-p b))
                           (if (< (alarm-time-ms a) (alarm-time-ms b))
                               (setq c (make-alarm :device "TR ON" :time-ms (+ (alarm-time-ms a) 10)))
                               (setq c (make-alarm :device "TR ON" :time-ms (+ (alarm-time-ms b) 10))))
                       :dismiss))
           (a (b c ok) (if (and (alarm-p b) (alarm-p c))
                           (if (< (- (alarm-time-ms c) (alarm-time-ms b)) 0)
                               (setq a (make-alarm :device "MAIN PROT OPTD" :time-ms (- (alarm-time-ms c) 10)))
                               :dismiss)
                           :dismiss))

          (b (a c ok) (if (and (alarm-p a) (alarm-p c))
                           (if (< (- (alarm-time-ms c) (alarm-time-ms a)) 0)
                               (setq b (make-alarm :device "MAIN PROT OPTD" :time-ms (- (alarm-time-ms c) 10)))
                               :dismiss)
                           :dismiss))))
```

A.1.3 Three Input

```
(constraint three-tr-component ((a cell) (b cell) (c cell) (d cell) (ok assumption))
 (formulae (d (a b c ok) (if (and (alarm-p a) (alarm-p b) (alarm-p c))
                           (if (< (alarm-time-ms a) (alarm-time-ms b))
                               (if (< (alarm-time-ms a) (alarm-time-ms c))
                                   (setq d (make-alarm :device "TR ON" :time-ms (+ (alarm-time-ms a) 10)))
                                   (setq d (make-alarm :device "TR ON" :time-ms (+ (alarm-time-ms c) 10))))
                               (if (< (alarm-time-ms b) (alarm-time-ms c))
                                   (setq d (make-alarm :device "TR ON" :time-ms (+ (alarm-time-ms b) 10)))
                                   (setq d (make-alarm :device "TR ON" :time-ms (+ (alarm-time-ms c) 10)))))
                         :dismiss))
           (c (a b d ok) (if (and (alarm-p a) (alarm-p b) (alarm-p d))
                           (if (< (alarm-time-ms a) (alarm-time-ms b))
                               (if (< (- (alarm-time-ms d) (alarm-time-ms a)) 0)
                                   (setq c (make-alarm :device "MAIN PROT OPTD" :time-ms (- (alarm-time-ms d) 10)))
                                   :dismiss)
                               (if (< (- (alarm-time-ms d) (alarm-time-ms b)) 0)
                                   (setq c (make-alarm :device "MAIN PROT OPTD" :time-ms (- (alarm-time-ms d) 10)))
                                   :dismiss))
                           :dismiss))
          (b (a c d ok) (if (and (alarm-p a) (alarm-p c) (alarm-p d))
                           (if (< (alarm-time-ms a) (alarm-time-ms c))
                               (if (< (- (alarm-time-ms d) (alarm-time-ms a)) 0)
                                   (setq b (make-alarm :device "MAIN PROT OPTD" :time-ms (- (alarm-time-ms d) 10)))
                                   :dismiss)
                               (if (< (- (alarm-time-ms d) (alarm-time-ms c)) 0)
                                   (setq b (make-alarm :device "MAIN PROT OPTD" :time-ms (- (alarm-time-ms d) 10)))
                                   :dismiss))
```

```
                 :dismiss))
(a (b c d ok) (if (and (alarm-p b) (alarm-p c) (alarm-p d))
                 (if (< (alarm-time-ms c) (alarm-time-ms b))
                     (if (< (- (alarm-time-ms d) (alarm-time-ms c)) 0)
                         (setq a (make-alarm :device "MAIN PROT OPTD" :time-ms (- (alarm-time-ms d) 10)))
                         :dismiss)
                     (if (< (- (alarm-time-ms d) (alarm-time-ms b)) 0)
                         (setq a (make-alarm :device "MAIN PROT OPTD" :time-ms (- (alarm-time-ms d) 10)))
                         :dismiss))
                 :dismiss))
 ))
```

A.2 Intertrip

```
(constraint one-int-component ((a cell) (b cell) (ok assumption))
 (formulae (b (a ok) (if (alarm-p a)
                         (setq b (make-alarm :device "INT x REC ON" :time-ms (+ (alarm-time-ms a) 30)))
                         :dismiss))
           (a (b ok) (if (alarm-p b)
                         (setq b (make-alarm :device "INT x REC ON" :time-ms (+ (alarm-time-ms b) 30))) :dismiss))
            ))
```

A.3 Circuit Breaker

A.3.1 One Input

```
(constraint one-cb-component ((a cell) (b cell) (ok assumption))
 (formulae (b (a ok) (if (alarm-p a)
                         (setq b (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms a) 40)))
                         :dismiss))
           (a (b ok) (if (alarm-p b)
                         (setq a (make-alarm :device "TRIP OPERATION" :time-ms (- (alarm-time-ms b) 40)))
                         :dismiss))))
```

A.3.2 Two Input

```
(constraint two-cb-component ((a cell) (b cell) (c cell) (ok assumption))
 (formulae (c (a b ok) (if (and (alarm-p a) (alarm-p b))
                           (if (< (alarm-time-ms a) (alarm-time-ms b))
                               (setq c (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms a) 40)))
                               (setq c (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms b) 40))))
                       :dismiss))
           (a (b c ok) (if (and (alarm-p b) (alarm-p c))
                           (if (< (- (alarm-time-ms c) (alarm-time-ms b)) 40)
                               (setq a (make-alarm :device "TRIP OPERATION" :time-ms (- (alarm-time-ms c) 40)))
                               :dismiss)
                           :dismiss))

          (b (a c ok) (if (and (alarm-p a) (alarm-p c))
                           (if (< (- (alarm-time-ms c) (alarm-time-ms a)) 40)
                               (setq b (make-alarm :device "TRIP OPERATION" :time-ms (- (alarm-time-ms c) 40)))
                               :dismiss)
                           :dismiss))))
```

A.3.3 Three Input

```
(constraint three-cb-component ((a cell) (b cell) (c cell) (d cell) (ok assumption))
 (formulae (d (a b c ok) (if (and (alarm-p a) (alarm-p b) (alarm-p c))
                           (if (< (alarm-time-ms a) (alarm-time-ms b))
                               (if (< (alarm-time-ms a) (alarm-time-ms c))
                                   (setq d (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms a) 40)))
                                   (setq d (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms c) 40))))
                               (if (< (alarm-time-ms b) (alarm-time-ms c))
                                   (setq d (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms b) 40)))
                                   (setq d (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms c) 40)))))
                         :dismiss))
           (c (a b d ok) (if (and (alarm-p a) (alarm-p b) (alarm-p d))
                           (if (< (alarm-time-ms a) (alarm-time-ms b))
                               (if (< (- (alarm-time-ms d) (alarm-time-ms a)) 40)
                                   (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 40)))
                                   :dismiss)
                               (if (< (- (alarm-time-ms d) (alarm-time-ms b)) 40)
                                   (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 40)))
                                   :dismiss))
                           :dismiss))
          (b (a c d ok) (if (and (alarm-p a) (alarm-p c) (alarm-p d))
                           (if (< (alarm-time-ms a) (alarm-time-ms c))
                               (if (< (- (alarm-time-ms d) (alarm-time-ms a)) 40)
                                   (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 40)))
                                   :dismiss)
                               (if (< (- (alarm-time-ms d) (alarm-time-ms c)) 40)
                                   (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 40)))
                                   :dismiss))
```

```
                 :dismiss))
(a (b c d ok) (if (and (alarm-p b) (alarm-p c) (alarm-p d))
                 (if (< (alarm-time-ms c) (alarm-time-ms b))
                     (if (< (- (alarm-time-ms d) (alarm-time-ms c)) 40)
                         (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 40)))
                         :dismiss)
                     (if (< (- (alarm-time-ms d) (alarm-time-ms b)) 40)
                         (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 40)))
                         :dismiss))
                 :dismiss))
 ))
```

A.3.4 Four Input

```
(constraint four-cb-component ((a cell) (b cell) (c cell) (d cell) (e cell) (ok assumption))
 (formulae (e (a b c d ok)
              (if (and (alarm-p a) (alarm-p b) (alarm-p c))
                 (if (< (alarm-time-ms a) (alarm-time-ms b))
       (if (< (alarm-time-ms a) (alarm-time-ms c))
           (if (< (alarm-time-ms a) (alarm-time-ms d))
               (setq e (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms a) 40)))
               (setq e (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms d) 40))))
           (if (< (alarm-time-ms c) (alarm-time-ms d))
               (setq e (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms c) 40)))
               (setq e (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms d) 40)))))
                       (if (< (alarm-time-ms b) (alarm-time-ms c))
                           (if (< (alarm-time-ms b) (alarm-time-ms d))
                              (setq e (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms b) 40)))
                              (setq e (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms d) 40))))
                           (if (< (alarm-time-ms c) (alarm-time-ms d))
                              (setq e (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms c) 40)))
                              (setq e (make-alarm :device "CB OPEN" :time-ms (+ (alarm-time-ms d) 40))))))
              :dismiss))
           (d (a b c e ok) (if (and (alarm-p a) (alarm-p b) (alarm-p c) (alarm-p e))
                               (if (< (alarm-time-ms a) (alarm-time-ms b))
                                   (if (< (alarm-time-ms a) (alarm-time-ms c))
                                       (if (< (- (alarm-time-ms e) (alarm-time-ms a)) 40)
                                           (setq d (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                           :dismiss)
                                       (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 40)
                                           (setq d (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
```

```
                                :dismiss))
                        (if (< (alarm-time-ms b) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms b)) 40)
                                (setq d (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 40)
                                (setq d (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)))
                :dismiss))
(c (a b d e ok) (if (and (alarm-p a) (alarm-p b) (alarm-p d) (alarm-p e))
                    (if (< (alarm-time-ms a) (alarm-time-ms b))
                        (if (< (alarm-time-ms a) (alarm-time-ms d))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms a)) 40)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms d)) 40)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss))
                        (if (< (alarm-time-ms b) (alarm-time-ms d))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms b)) 40)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms d)) 40)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)))
                :dismiss))
(b (a c d e ok) (if (and (alarm-p a) (alarm-p c) (alarm-p d) (alarm-p e))
                    (if (< (alarm-time-ms a) (alarm-time-ms d))
                        (if (< (alarm-time-ms a) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms a)) 40)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 40)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss))
                        (if (< (alarm-time-ms d) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms d)) 40)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 40)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)))
                :dismiss))
(a (b c d e ok) (if (and (alarm-p b) (alarm-p c) (alarm-p d) (alarm-p e))
                    (if (< (alarm-time-ms d) (alarm-time-ms b))
                        (if (< (alarm-time-ms d) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms d)) 40)
                                (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 40)
                                (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss))
                        (if (< (alarm-time-ms b) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms b)) 40)
                                (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 40)
                                (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 40)))
                                :dismiss)))
                :dismiss))
))
```

A.4 Delayed Automatic Recloser

A.4.1 One Input

```
(constraint one-dar-component ((a cell) (b cell) (ok assumption))
 (formulae (b (a ok) (if (alarm-p a)
                         (setq b (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms a) 20000)))
                         :dismiss))
           (a (b ok) :dismiss)))
```

A.4.2 Two Input

```
(constraint two-dar-component ((a cell) (b cell) (c cell) (ok assumption))
 (formulae (c (a b ok) (if (and (alarm-p a) (alarm-p b))
                               (if (< (alarm-time-ms a) (alarm-time-ms b))
                                   (setq c (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms a) 20000)))
                                   (setq c (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms b) 20000))))
                           :dismiss))
           (a (b c ok) (if (and (alarm-p b) (alarm-p c))
                               (if (< (- (alarm-time-ms c) (alarm-time-ms b)) 19000)
                                   (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms c) 20000))) :dismiss)))
           (b (a c ok) (if (and (alarm-p a) (alarm-p c))
                               (if (< (- (alarm-time-ms c) (alarm-time-ms a)) 19000)
                                   (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms c) 20000)))
                                   :dismiss)))
           ))
```

A.4.3 Three Input

```
(constraint three-dar-component ((a cell) (b cell) (c cell) (d cell) (ok assumption))
 (formulae (d (a b c ok) (if (and (alarm-p a) (alarm-p b) (alarm-p c))
                            (if (< (alarm-time-ms a) (alarm-time-ms b))
                                (if (< (alarm-time-ms a) (alarm-time-ms c))
                                    (setq d (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms a) 20000)))
                                    (setq d (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms c) 20000))))
                                (if (< (alarm-time-ms b) (alarm-time-ms c))
                                    (setq d (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms b) 20000)))
                                    (setq d (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms c) 20000)))))
                          :dismiss))
          (c (a b d ok) (if (and (alarm-p a) (alarm-p b) (alarm-p d))
                            (if (< (alarm-time-ms a) (alarm-time-ms b))
                                (if (< (- (alarm-time-ms d) (alarm-time-ms a)) 19000)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 20000)))
                                :dismiss)
                                (if (< (- (alarm-time-ms d) (alarm-time-ms b)) 19000)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 40)))
                                :dismiss))
                            :dismiss))
          (b (a c d ok) (if (and (alarm-p a) (alarm-p c) (alarm-p d))
                            (if (< (alarm-time-ms a) (alarm-time-ms c))
                                (if (< (- (alarm-time-ms d) (alarm-time-ms a)) 19000)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 20000)))
                                :dismiss)
                                (if (< (- (alarm-time-ms d) (alarm-time-ms c)) 19000)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 20000)))
                                :dismiss))
```

```
                 :dismiss))
(a (b c d ok) (if (and (alarm-p b) (alarm-p c) (alarm-p d))
                 (if (< (alarm-time-ms c) (alarm-time-ms b))
                     (if (< (- (alarm-time-ms d) (alarm-time-ms c)) 19000)
                         (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 20000)))
                         :dismiss)
                     (if (< (- (alarm-time-ms d) (alarm-time-ms b)) 19000)
                     (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms d) 20000)))
                     :dismiss))
                 :dismiss))
 ))
```

A.4.4 Four Input

```
(constraint four-dar-component ((a cell) (b cell) (c cell) (d cell) (e cell) (ok assumption))
 (formulae (e (a b c d ok)
              (if (and (alarm-p a) (alarm-p b) (alarm-p c))
                 (if (< (alarm-time-ms a) (alarm-time-ms b))
       (if (< (alarm-time-ms a) (alarm-time-ms c))
           (if (< (alarm-time-ms a) (alarm-time-ms d))
               (setq e (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms a) 20000)))
               (setq e (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms d) 20000))))
           (if (< (alarm-time-ms c) (alarm-time-ms d))
               (setq e (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms c) 20000)))
               (setq e (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms d) 20000)))))
                       (if (< (alarm-time-ms b) (alarm-time-ms c))
                           (if (< (alarm-time-ms b) (alarm-time-ms d))
                              (setq e (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms b) 20000)))
                              (setq e (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms d) 20000))))
                           (if (< (alarm-time-ms c) (alarm-time-ms d))
                              (setq e (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms c) 20000)))
                              (setq e (make-alarm :device "CB CLOSED" :time-ms (+ (alarm-time-ms d) 20000))))))
              :dismiss))
           (d (a b c e ok) (if (and (alarm-p a) (alarm-p b) (alarm-p c) (alarm-p e))
                               (if (< (alarm-time-ms a) (alarm-time-ms b))
                                   (if (< (alarm-time-ms a) (alarm-time-ms c))
                                       (if (< (- (alarm-time-ms e) (alarm-time-ms a)) 19000)
                                           (setq d (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                           :dismiss)
                                       (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 19000)
                                           (setq d (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
```

```
                                :dismiss))
                        (if (< (alarm-time-ms b) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms b)) 19000)
                                (setq d (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 19000)
                                (setq d (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)))
                :dismiss))
(c (a b d e ok) (if (and (alarm-p a) (alarm-p b) (alarm-p d) (alarm-p e))
                    (if (< (alarm-time-ms a) (alarm-time-ms b))
                        (if (< (alarm-time-ms a) (alarm-time-ms d))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms a)) 19000)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms d)) 19000)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss))
                        (if (< (alarm-time-ms b) (alarm-time-ms d))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms b)) 19000)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms d)) 19000)
                                (setq c (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)))
                :dismiss))
(b (a c d e ok) (if (and (alarm-p a) (alarm-p c) (alarm-p d) (alarm-p e))
                    (if (< (alarm-time-ms a) (alarm-time-ms d))
                        (if (< (alarm-time-ms a) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms a)) 19000)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 19000)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss))
                        (if (< (alarm-time-ms d) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms d)) 19000)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 19000)
                                (setq b (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)))
                :dismiss))
(a (b c d e ok) (if (and (alarm-p b) (alarm-p c) (alarm-p d) (alarm-p e))
                    (if (< (alarm-time-ms d) (alarm-time-ms b))
                        (if (< (alarm-time-ms d) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms d)) 19000)
                                (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)
                            (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 19000)
                                (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss))
                        (if (< (alarm-time-ms b) (alarm-time-ms c))
                            (if (< (- (alarm-time-ms e) (alarm-time-ms b)) 19000)
                                (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                :dismiss)
```

```
                                       (if (< (- (alarm-time-ms e) (alarm-time-ms c)) 19000)
                                           (setq a (make-alarm :device "TR ON" :time-ms (- (alarm-time-ms e) 20000)))
                                           :dismiss)))
                           :dismiss))
           ))
```

A.5 Manual Reclose

```
(constraint man-reclose-component ((a cell) (b cell) (ok assumption))
 (formulae (b (a ok)
                         :dismiss)
           (a (b ok)
                         :dismiss)))
```

A.6 Missing SCADA Alarm Model

```
(constraint scada-component ((a cell) (b cell) (ok assumption))
            (formulae (b (a ok) (if (alarm-p a)
                                    (setq b (make-alarm :device (alarm-device a) :time-ms (alarm-time-ms a)))
                                    :dismiss))
                      (a (b ok) (if (alarm-p b)
                                    :dismiss
                                    :dismiss))))
```

Appendix B

Protection Database Schema

The protection database has been constructed as a relational database made up of a number of components which are connected through specific indices. The simplified schema of the database can be seen in Figure B.1 demonstrating the relationship from circuits down to individual protection devices.

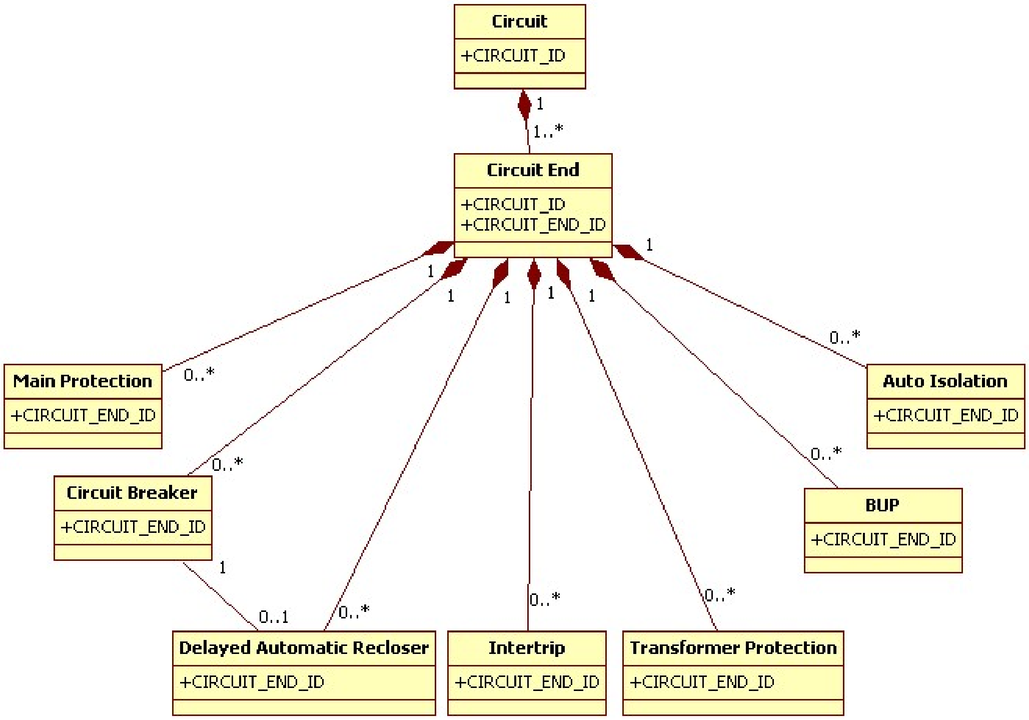


Figure B.1: Database Schema - Circuits

This hierarchy is then linked to SCADA alarms and models to allow the independent update of

SCADA alarm legends, actual models and location identifiers. The simplified schema demonstrating the relationship between these tables can be seen in Figure B.2.

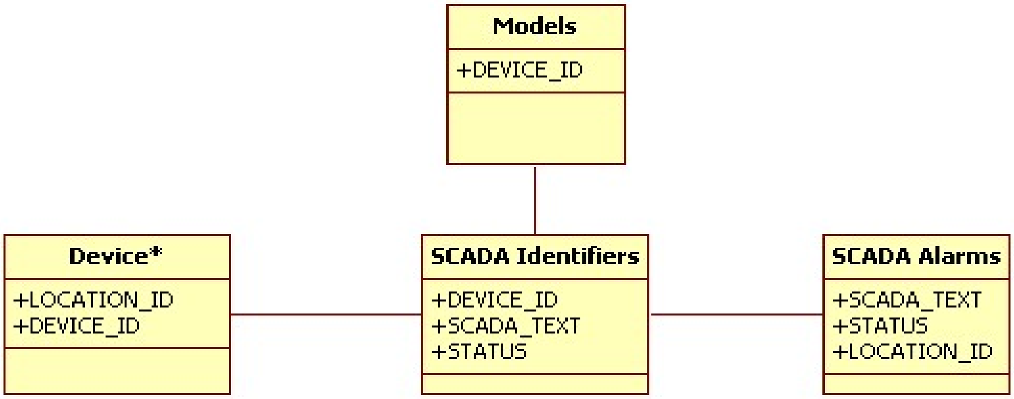


Figure B.2: Database Schema - Models and SCADA Alarms
* - Main Protection, Intertrip, Circuit Breaker, Delayed Automatic Recloser or Trip Relay

The protection database is made up of over 65 database tables which collectively allow the protection network, SCADA alarms and models to be updated and accessed through a web-based front end.

Appendix C

Expert System Rules

C.1 Model Configuration and Building

A set of simple rules were produced to provide a means of extracting data from the protection database and setting up the models and alarms for use during diagnosis.

C.1.1 Identify Models

The first requirements of the rules is to identify the devices present within the protection scheme and therefore select the correct model and assign the correct inputs.

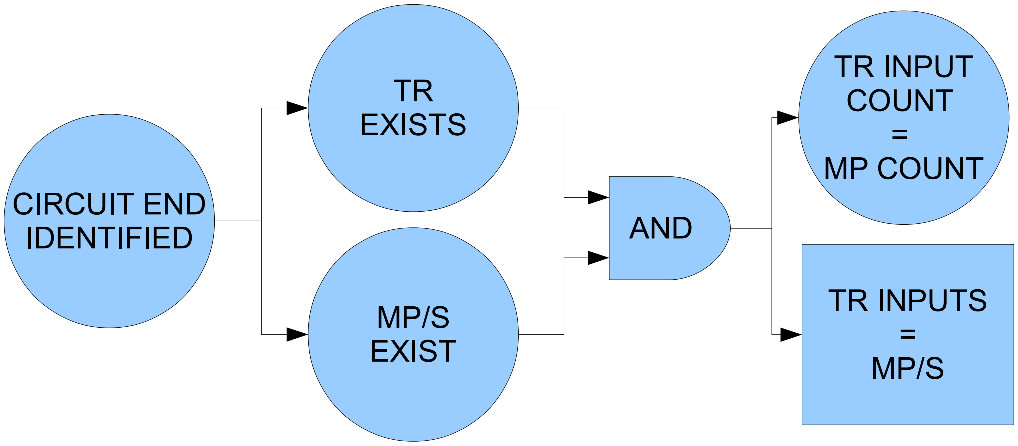


Figure C.1: Identify the required trip relay model and assign inputs

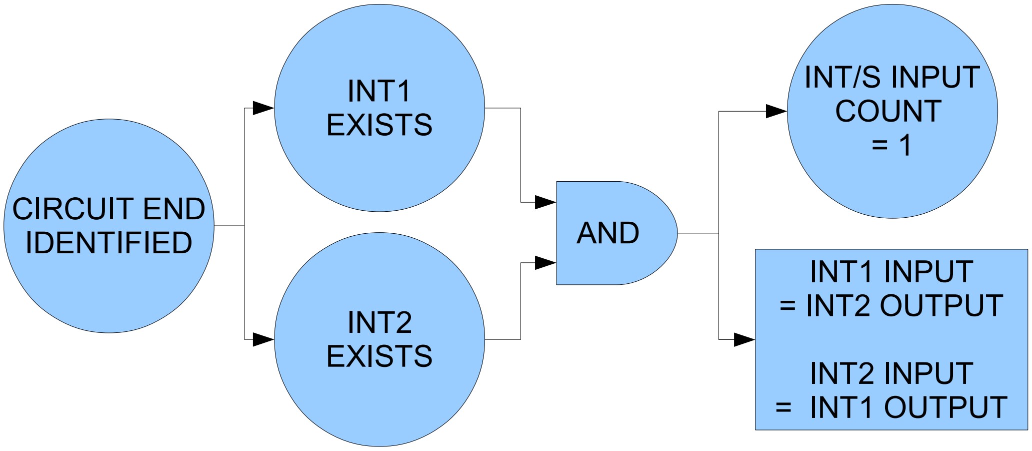


Figure C.2: Identify the required Intertrip model and assign inputs

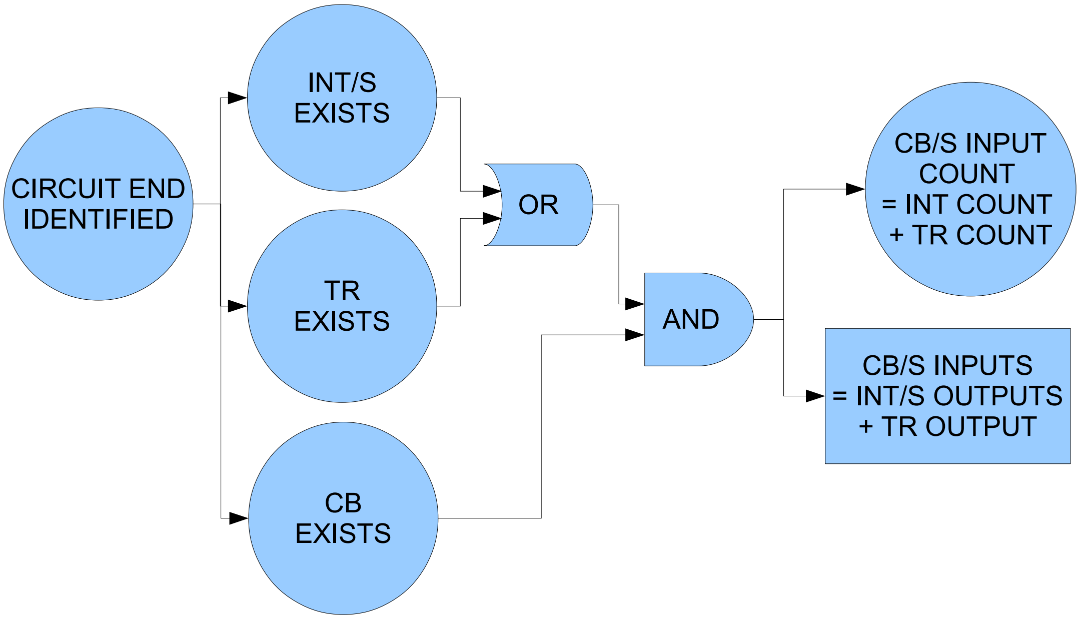

CIRCUIT END IDENTIFIED
INT/S EXISTS
TR EXISTS
CB EXISTS
OR
AND
CB/S INPUT COUNT = INT COUNT + TR COUNT
CB/S INPUTS = INT/S OUTPUTS + TR OUTPUT

Figure C.3: Identify the required Circuit Breaker model and assign inputs

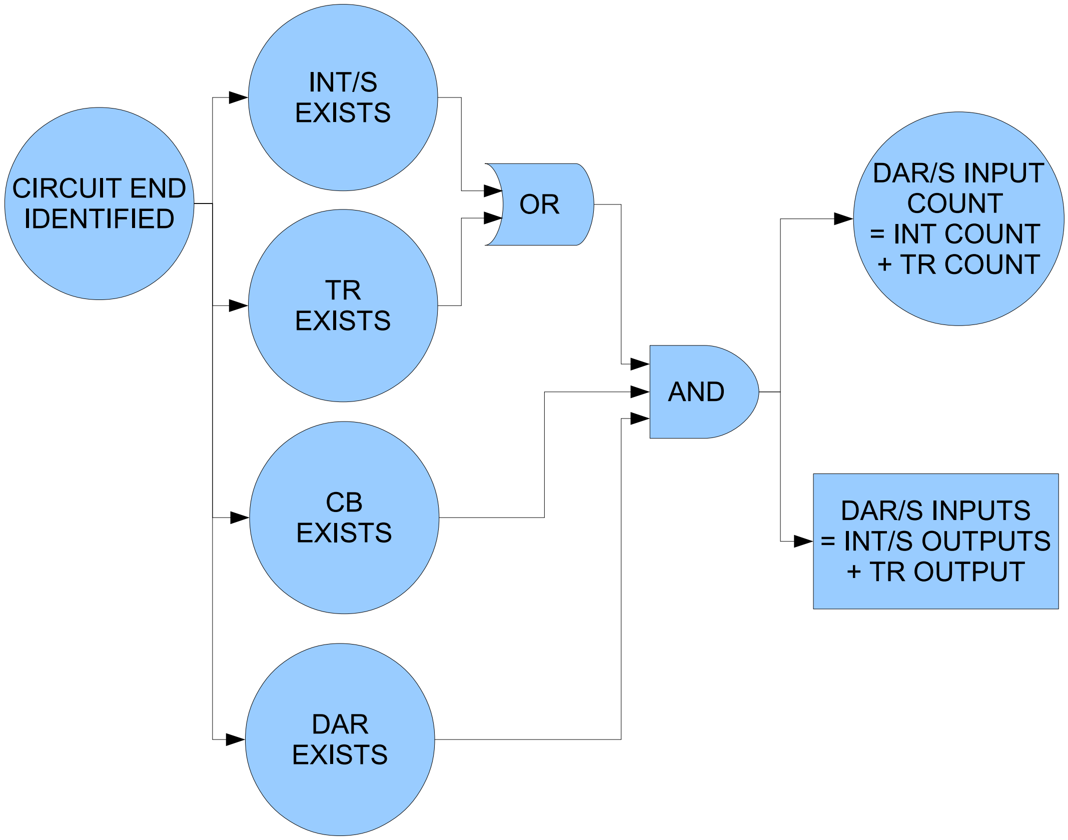

CIRCUIT END IDENTIFIED
INT/S EXISTS
TR EXISTS
CB EXISTS
DAR EXISTS
OR
AND
DAR/S INPUT COUNT = INT COUNT + TR COUNT
DAR/S INPUTS = INT/S OUTPUTS + TR OUTPUT

Figure C.4: Identify the required DAR model and assign inputs

C.1.2 Configure Models

Due to the range of device types used on a typical protection scheme can vary greatly in functionality with some modern circuit breakers expected to operate significantly faster than their older counterparts for some schemes the use of non-default timings need to be utilised and for others their range of acceptable operation timings also must be utilised.

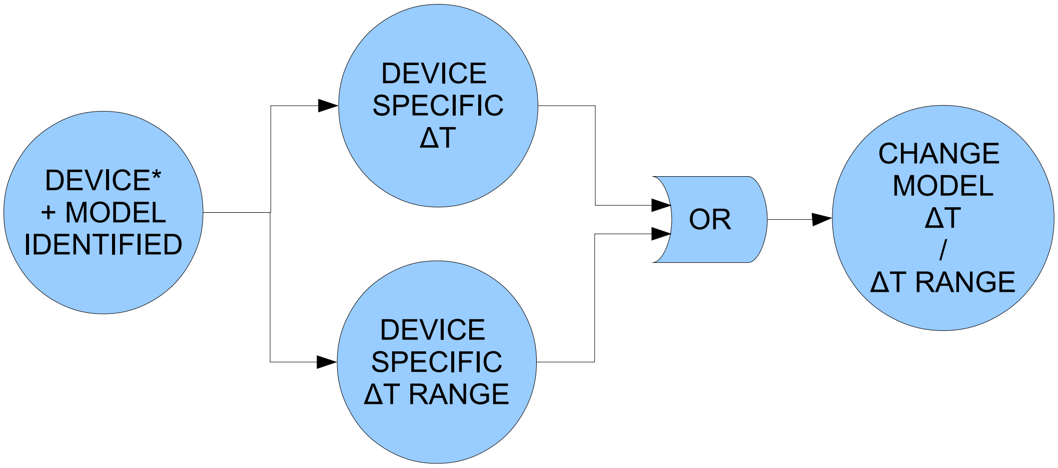


Figure C.5: Configure non-default timings and time ranges

C.1.3 Add SCADA Models

In order to carry out missing alarm detection all devices present on the network must also have a SCADA model associated with it. This includes main protection devices where although they cannot be validated on their performance, other than actual arrival, they must be covered by missing alarm detection.

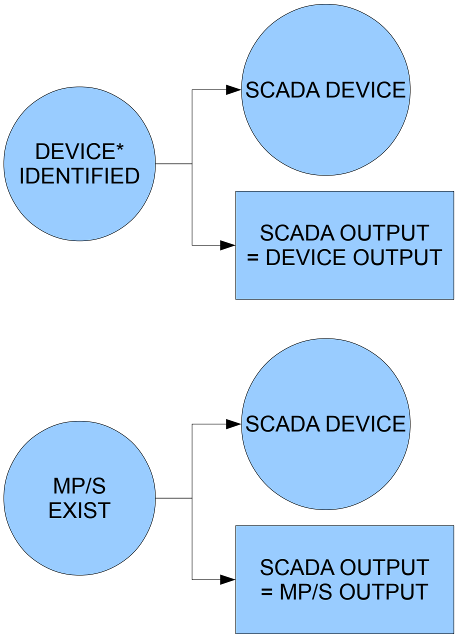


Figure C.6: Add missing alarm models

C.2 Missing Alarm Insertion

After a 2 minute time out, any alarms deemed to have not arrived/signal the non-operation of a device, must trigger the insertion of a missing alarm fact.

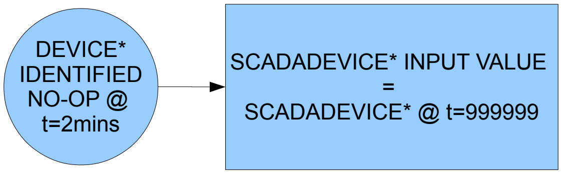

DEVICE* IDENTIFIED NO-OP @ t=2mins
SCADADEVICE* INPUT VALUE = SCADADEVICE* @ t=999999

Figure C.7: Add missing alarms

C.3 Priority and Summary Generation

A small set of rules are utilised to provide engineers with a high level summary and priority for incidents. These allow engineers to prioritise incidents and quickly identify problems of a particular nature, for example where DARs have locked out or a circuit breaker has failed to operate, where a quick response is of high value. The rules associated with assigning these priorities, red, amber and green, and fault summaries are shown in this appendix. The rules are evaluated using the Drools Rule Engine

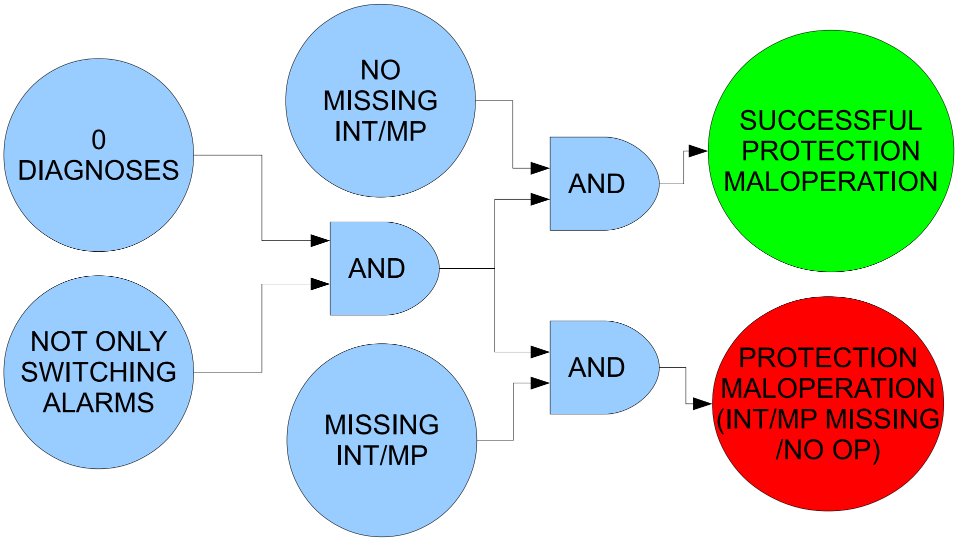

0 DIAGNOSES
NOT ONLY SWITCHING ALARMS
NO MISSING INT/MP
MISSING INT/MP
AND
AND
AND
SUCCESSFUL PROTECTION MALOPERATION
PROTECTION MALOPERATION (INT/MP MISSING /NO OP)

Figure C.8: Green - Successful Protection Operation

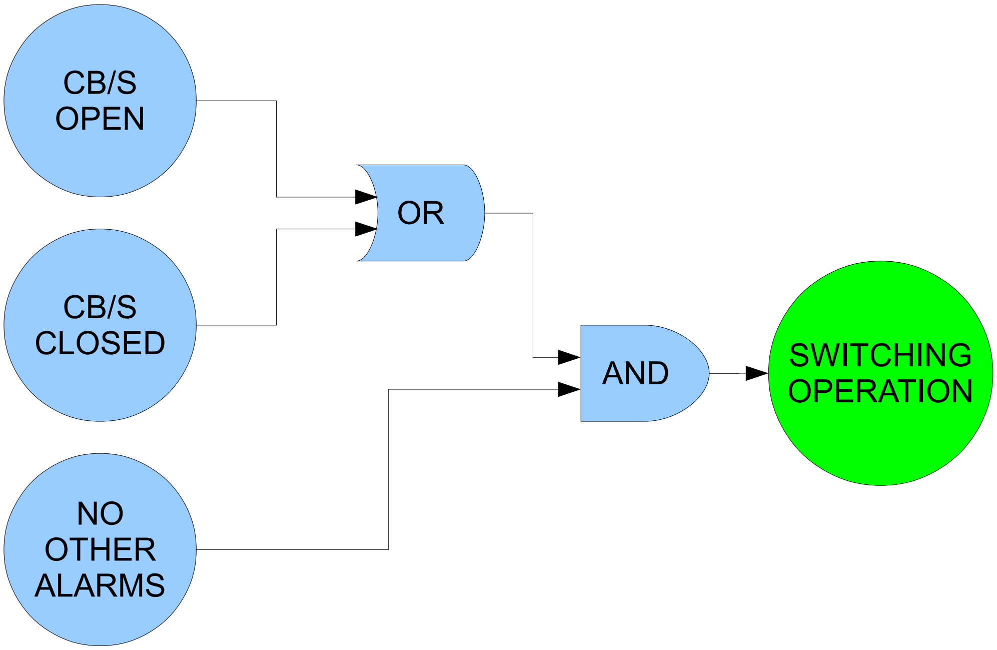


Figure C.9: Green - Switching Operation

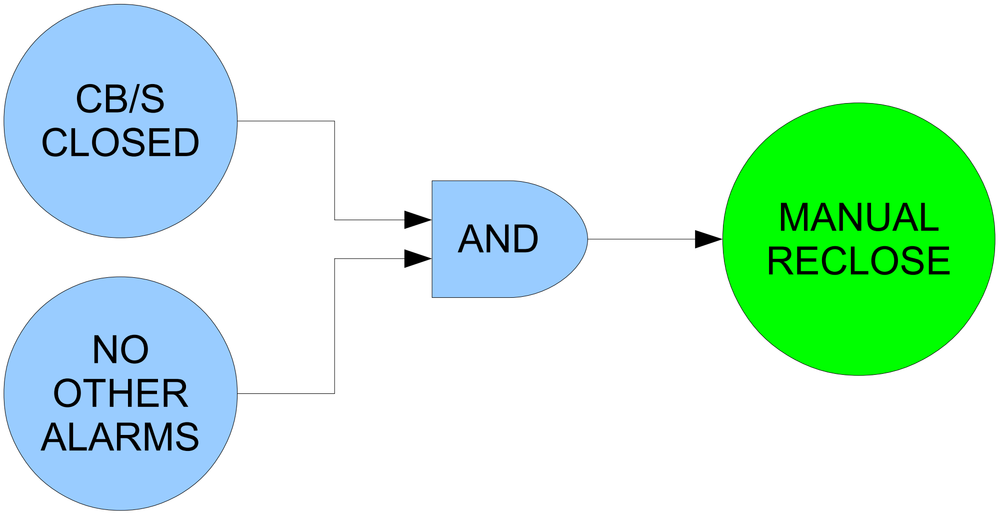


Figure C.10: Green - Manual Reclose

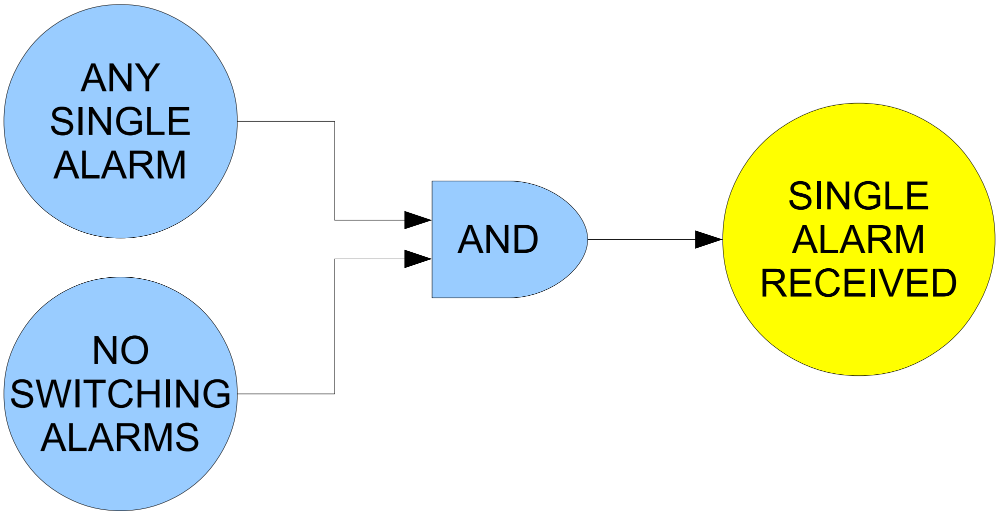


Figure C.11: Amber - Single (Non-switching) Alarm Received

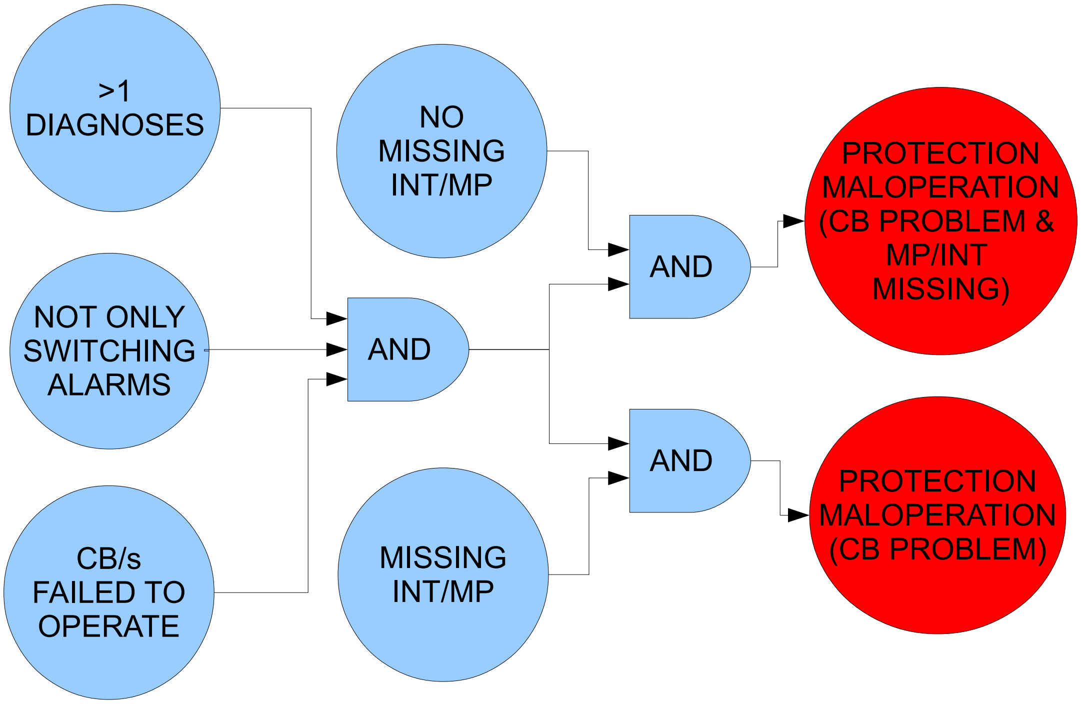

>1 DIAGNOSES
NOT ONLY SWITCHING ALARMS
CB/s FAILED TO OPERATE
AND
NO MISSING INT/MP
MISSING INT/MP
AND
AND
PROTECTION MALOPERATION (CB PROBLEM & MP/INT MISSING)
PROTECTION MALOPERATION (CB PROBLEM)

Figure C.12: Red - Circuit Breaker/s Failed to Operate

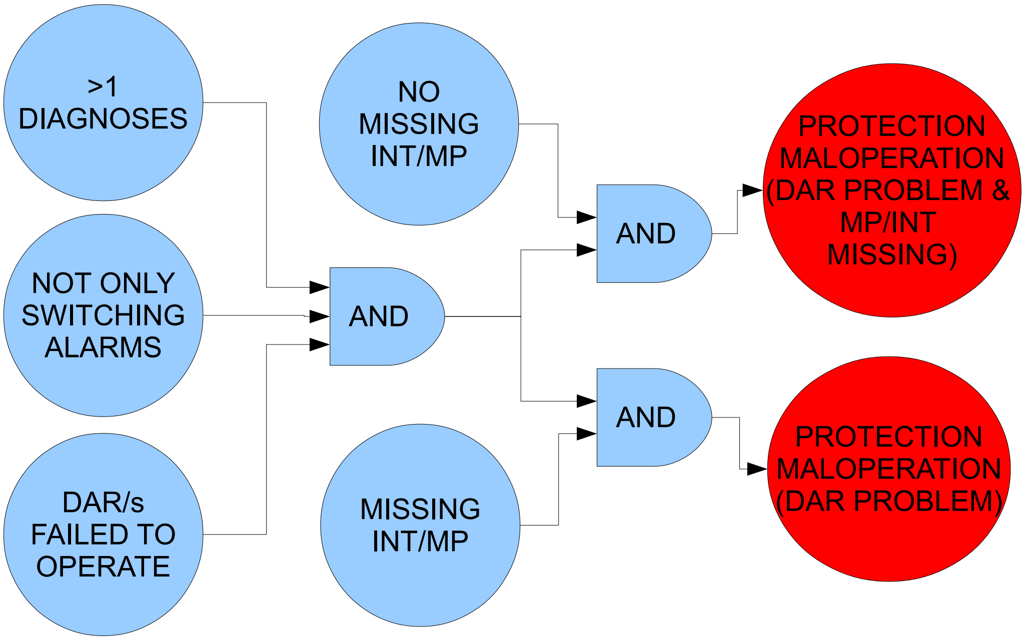

>1 DIAGNOSES
NOT ONLY SWITCHING ALARMS
DAR/s FAILED TO OPERATE
AND
NO MISSING INT/MP
MISSING INT/MP
AND
AND
PROTECTION MALOPERATION (DAR PROBLEM & MP/INT MISSING)
PROTECTION MALOPERATION (DAR PROBLEM)

Figure C.13: Red - Delayed Automatic Recloser/s Failed to Operate

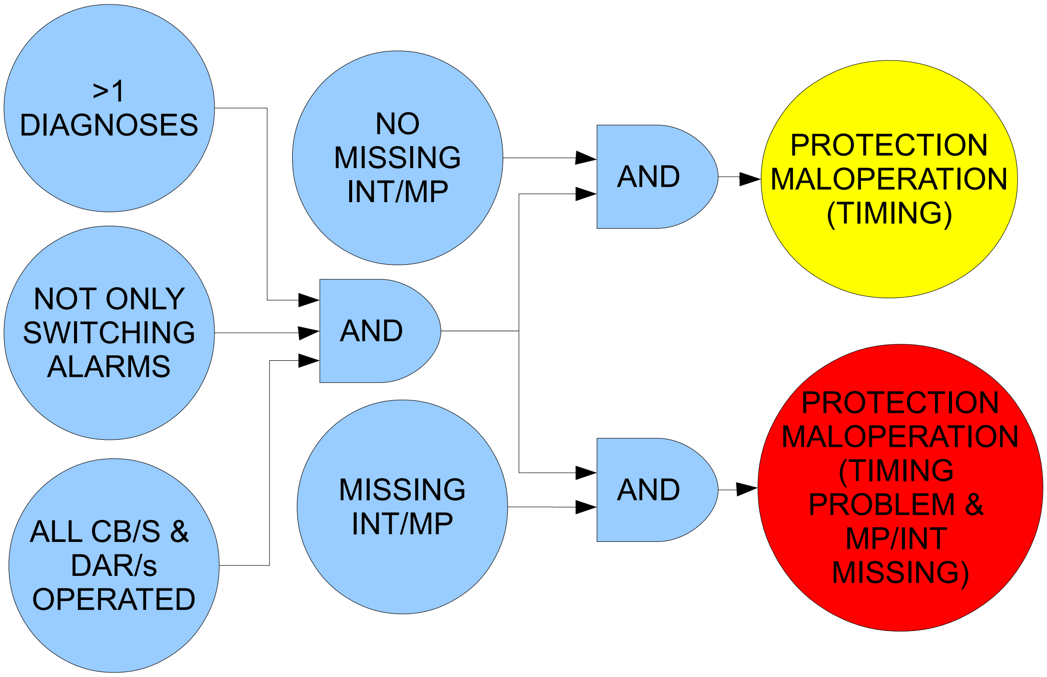

>1 DIAGNOSES
NOT ONLY SWITCHING ALARMS
ALL CB/S & DAR/s OPERATED
AND
NO MISSING INT/MP
MISSING INT/MP
AND
AND
PROTECTION MALOPERATION (TIMING)
PROTECTION MALOPERATION (TIMING PROBLEM & MP/INT MISSING)

Figure C.14: Amber - Protection Operated with a timing problem

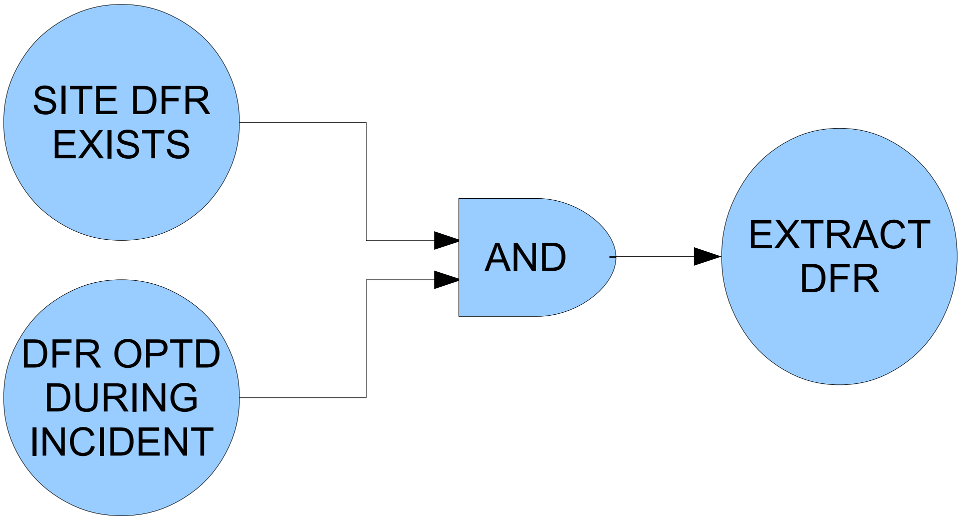


Figure C.15: Query Existance of DFR and Retrieve

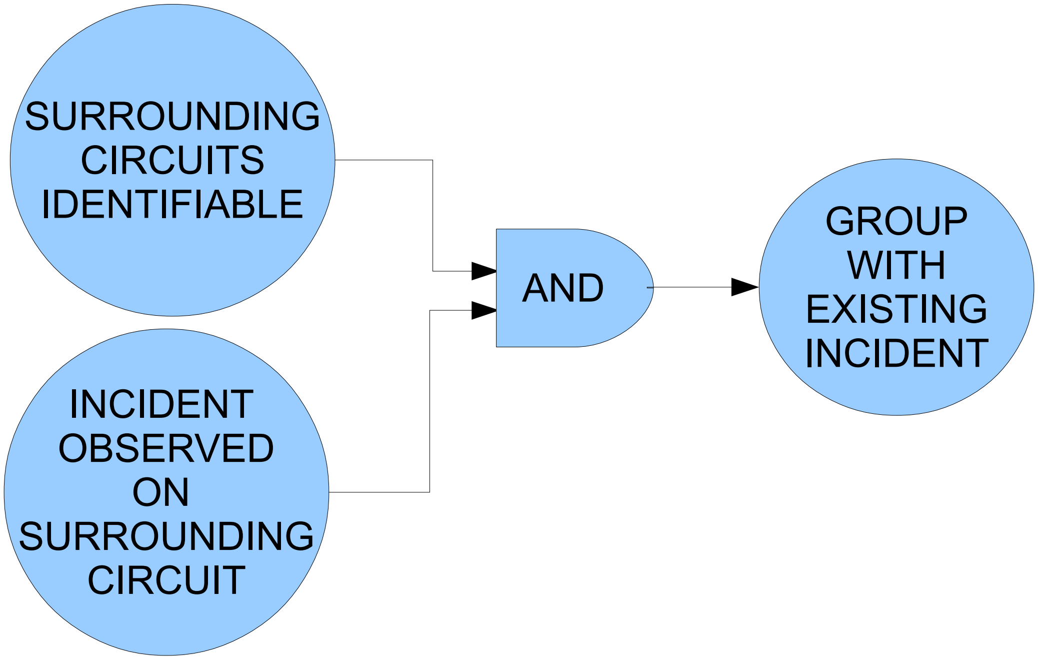

SURROUNDING CIRCUITS IDENTIFIABLE
INCIDENT OBSERVED ON SURROUNDING CIRCUIT
AND
GROUP WITH EXISTING INCIDENT

Figure C.16: Query Existance of Surrounding Incidents and Retrieve

References

[Albert, 1989] Albert, L. (1989). Average case complexity analysis of RETE pattern-match algorithm and average size of join in databases. *Foundations of Software Technology and Theoretical Computer Science, 1989. Proceedings of the 9th Conference on,*, 1010:223–241. (cited on page 30)

[Bell et al., 1998] Bell, S. C., McArthur, S. D. J., McDonald, J. R., Burt, G. M., Mather, R., and Cumming, T. (1998). Model-based analysis of protection system performance. *Generation, Transmission and Distribution, IEE Proceedings-*, 145(5):547 –552. (cited on page 43)

[Beschta et al., 1993] Beschta, A., Dressler, O., Freitag, H., Montag, M., and Struss, P. (1993). A model-based approach to fault localisation in power transmission networks. *Intelligent Systems Engineering*, 2(1):3 –14. (cited on pages 31, 44, 45, 49)

[Bi and Li, 2007] Bi, T. and Li, Q. (2007). Multi-layer disturbance processing system based on multiple information sources. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–4. (cited on page 18)

[Biswas et al., 2004] Biswas, T., Davari, A., and Feliachi, A. (2004). Modeling and analysis of discrete event behaviors in power system using petri nets. In *System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on*, pages 165 – 169. (cited on pages 39, 40)

[Bratko, 2001] Bratko, I. (2001). *Prolog Programming for Artificial Intelligence*. International Computer Science Series. Addison Wesley. (cited on page 30)

[Brunner et al., 1993] Brunner, T., Nejdl, W., Schwarzjirg, H., and Sturm, M. (1993). On-line expert system for power system diagnosis and restoration. *Intelligent Systems Engineering*, 2(1):15–24. (cited on page 30)

[Chantler et al., 2000] Chantler, M., Pogliano, P., Aldea, A., Tornielli, G., Wyatt, T., and Jolley, A. (2000). The use of fault-recorder data for diagnosing timing and other related faults in electricity transmission networks. *Power Systems, IEEE Transactions on*, 15(4):1388–1393. (cited on pages 2, 19, 36, 37)

[David and Alla, 2005] David, R. and Alla, H. (2005). *Discrete, continuous, and hybrid Petri Nets*. Springer. (cited on page 39)

[Davidson et al., 2008] Davidson, E. M., McArthur, S. D. J., Cumming, T., and Watt, I. (2008). Automated analysis of scada and dfr data: Post-fault diagnosis of power system disturbances and condition assessment of plant. In *11th Annual Georgia Tech Fault and Disturbance Analysis Conference*, pages 1–7. (cited on pages 21, 118)

[Davidson et al., 2003] Davidson, E. M., McArthur, S. D. J., and McDonald, J. R. (2003). A toolset for applying model-based reasoning techniques to diagnostics for power systems protection. *Power Systems, IEEE Transactions on*, 18(2):680 – 687. (cited on pages 30, 44)

[Davidson et al., 2005] Davidson, E. M., McArthur, S. D. J., McDonald, J. R., Cumming, T., and Watt, I. (2005). Automating the analysis and management of power system data using multi-agent systems technology. (cited on page 44)

[Davidson et al., 2006] Davidson, E. M., McArthur, S. D. J., McDonald, J. R., Cumming, T., and Watt, I. (2006). Applying Multi-Agent System Technology in Practice: Automated Management and analysis of SCADA and Digital Fault Recorder Data. *IEEE Transactions on Power Systems*, 21(2):559–567. (cited on pages 2, 19)

[de Kleer, 1990] de Kleer, J. (1990). Using crude probability estimates to guide diagnosis. *Artificial Intelligence*, 45(3):381 – 391. (cited on page 48)

[De Kleer and Williams, 1987] De Kleer, J. and Williams, B. C. (1987). Diagnosing multiple faults. *ARTIFICIAL INTELLIG.*, 32(1):97–130. (cited on pages 42, 48, 49, 50)

[Dreyfus, 2005] Dreyfus, G. (2005). *Neural networks: methodology and applications*. Springer Verlag. (cited on page 33)

[Edwards et al., 2013] Edwards, C., Davidson, E., McArthur, S., Watt, I., and Cumming, T. (2013). Flexible model-based alarm processing for protection performance assessment and incident identification. *Power Systems, IEEE Transactions on*, PP(99):1–8. (cited on page 4)

[Forbus and de Kleer, 1993] Forbus, K. D. and de Kleer, J. (1993). *Building problem solvers*. MIT Press, Cambridge, MA, USA. (cited on pages 1, 59)

[Forgy, 1982] Forgy, C. L. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1):17 – 37. (cited on pages 27, 30)

[Fritzen et al., 2010] Fritzen, P., Cardoso, G., Zauk, J., de Morais, A., Bezerra, U., and Beck, J. (2010). Alarm processing and fault diagnosis in power systems using artificial neural networks and genetic algorithms. In *Industrial Technology (ICIT), 2010 IEEE International Conference on*, pages 891 –896. (cited on page 33)

[Fukuyama and Ueki, 1991] Fukuyama, Y. and Ueki, Y. (1991). Development of an expert system for analyzing faults in power system using artificial neural network based waveform recognition. In *Circuits and Systems, 1991., IEEE International Sympoisum on*, pages 1137 –1140 vol.2. (cited on pages 33, 34)

[Grice et al., 2011] Grice, A., Peer, J. M., and Morris, G. T. (2011). Today's aging workforce 2014; who will fill their shoes? In *Protective Relay Engineers, 2011 64th Annual Conference for*, pages 483 –491. (cited on page 1)

[Hamscher et al., 1992] Hamscher, W., Console, L., and de Kleer, J., editors (1992). *Logical Foundations*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (cited on page 48)

[Hopcroft et al., 2007] Hopcroft, J., Motwani, R., and Ullman, J. (2007). *Introduction to automata theory, languages, and computation*. Pearson/Addison Wesley. (cited on page 37)

[Hor et al., 2007] Hor, C.-L., Crossley, P. A., and Millar, D. L. (2007). Application of genetic algorithm and rough set theory for knowledge extraction. In *Power Tech, 2007 IEEE Lausanne*, pages 1117 –1122. (cited on pages 2, 19, 47)

[Hossack et al., 2002] Hossack, J., McArthur, S. D. J., McDonald, J. R., Stokoe, J., and Cumming, T. (2002). A multi-agent approach to power system disturbance diagnosis. In *Power System Management and Control, 2002. Fifth International Conference on (Conf. Publ. No. 488)*, pages 317–322. (cited on pages 30, 45)

[Hossack et al., 2003a] Hossack, J., Menal, J., McArthur, S. D. J., and McDonald, J. R. (2003a). A multi-agent architecture for protection engineering diagnostic assistance. *Power Systems, IEEE Transactions on*, 18(2):639–647. (cited on page 44)

[Hossack et al., 2003b] Hossack, J. A., Menal, J., McArthur, S. D. J., and McDonald, J. R. (2003b). A Multi-Agent Architecture for Protection Engineering Diagnostic assistance. *IEEE Transactions on Power Systems*, 18(2):639–647. (cited on page 2)

[IEC, 2005] IEC (2005). Communications networks and Systems in Substations. Document IEC 61850. (cited on page 1)

[Jackson, 1985] Jackson, P. (1985). *Introduction to Artificial Intelligence*. Dover books explaining science. Dover. (cited on page 24)

[Jiang et al., 2003] Jiang, S., Kumar, R., and Garcia, H. (2003). Diagnosis of repeated/intermittent failures in discrete event systems. *Robotics and Automation, IEEE Transactions on*, 19(2):310 – 323. (cited on pages 37, 45)

[Kezunovic et al., 2010] Kezunovic, M., Zheng, C., and Pang, C. (2010). Merging pmu, operational, and non-operational data for interpreting alarms, locating faults and preventing cascades. *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1 –9. (cited on pages 2, 2)

[Knill et al., 1996] Knill, E., Cox, P. T., and Pietrzykowski, T. (1996). Completing fault models for abductive diagnosis. (cited on page 48)

[Kosko and Isaka, 1993] Kosko, B. and Isaka, S. (1993). Fuzzy logic. *Scientific American*, 269(1):62–7. (cited on page 35)

[Krivine and Jehl, 1996] Krivine, J. P. and Jehl, O. (1996). The austral system for diagnosis and power restoration: an overview. In *Intelligent Systems Applications to Power Systems, 1996. Proceedings, ISAP '96., International Conference on*, pages 180–186. (cited on page 2)

[Kwong and Yonge-Mallo, 2011] Kwong, R. H. and Yonge-Mallo, D. L. (2011). Fault diagnosis in discrete-event systems: Incomplete models and learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(1):118 –130. (cited on pages 37, 38)

[Lamperti and Zanella, 2011] Lamperti, G. and Zanella, M. (2011). Monitoring of active systems with stratified uncertain observations. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(2):356 –369. (cited on pages 2, 2)

[Luger, 2005] Luger, G. F. (2005). *Artificial intelligence: Structures and strategies for complex problem solving*, chapter 7, pages 247–302. Addison-Wesley Longman, 4 edition. (cited on page 35)

[Luo and Kezunovic, 2005] Luo, X. and Kezunovic, M. (2005). An expert system for diagnosis of digital relay operation. In *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*, page 6 pp. (cited on page 19)

[Luo and Kezunovic, 2008] Luo, X. and Kezunovic, M. (2008). Implementing fuzzy reasoning petri-nets for fault section estimation. *Power Delivery, IEEE Transactions on*, 23(2):676–685. (cited on pages 35, 39, 40, 45)

[Ma et al., 1992] Ma, T. K., Liu, C. C., Tsai, M. S., Rogers, R., Muchlinski, S. L., and Dodge, J. (1992). Operational experience and maintenance of online expert system for customer restoration and fault testing. *Power Systems, IEEE Transactions on*, 7(2):835 –842. (cited on pages 30, 30, 31, 32, 45)

[Malheiro et al., 2005] Malheiro, N., Vale, Z., Ramos, C., Cordeiro, M., Marques, A., and Couto, V. (2005). Decision support for power system control centers-a model based reasoning component. *Engineering intelligent systems for electrical engineering and communications*, 13(4):205. (cited on pages 43, 44, 45)

[Malheiro et al., 2007] Malheiro, N., Vale, Z., Ramos, C., Cordeiro, M., Marques, A., and Couto, V. (2007). Decision support system with incomplete and domain incoherent information management. *Control Intell. Syst.*, 35(3):202–210. (cited on page 2)

[Marathe et al., 1991] Marathe, H. Y., Liu, C. C., Tsai, M. S., Rogers, R. G., and Maurer, J. M. (1991). An online operational expert system with data validation capabilities [for power systems]. *Power Systems, IEEE Transactions on*, 6(2):882 –889. (cited on pages 30, 30, 31)

[McArthur et al., 1996] McArthur, S. D. J., Dysko, R., McDonald, J. R., Bell, S. C., Mather, R., and Burt, S. M. (1996). The application of model based reasoning within a decision support system for protection engineers. *Power Delivery, IEEE Transactions on*, 11(4):1748 –1754. (cited on pages 43, 45)

[McArthur et al., 2003] McArthur, S. D. J., McDonald, J. R., and Hossack, J. A. (2003). A multi-agent approach to power system disturbance diagnosis. In Rehtanz, C., editor, *Autonomous Systems and Intelligent Agents in Power System Control and Operation (Power Systems)*, pages 75–99. Springer-Verlag. (cited on pages 44, 45)

[McCarthy et al., 1955] McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html. (cited on page 23)

[McDonald et al., 1992] McDonald, J. R., Burt, G. M., and Young, D. J. (1992). Alarm processing and fault diagnosis using knowledge based systems for transmission and distribution network control. *Power Systems, IEEE Transactions on*, 7(3):1292 –1298. (cited on pages 2, 31, 43, 45)

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York. (cited on page 33)

[Monsef et al., 1997] Monsef, H., Ranjbar, A., and Jadid, S. (1997). Fuzzy rule-based expert system for power system fault diagnosis. *Generation, Transmission and Distribution, IEE Proceedings-*, 144(2):186 –192. (cited on page 35)

[Pawlak, 1985] Pawlak, Z. (1985). Rough sets and fuzzy sets. *Fuzzy Sets and Systems*, 17(1):99 – 102. (cited on page 35)

[Pfau-Wagenbauer and Nejdl, 1992a] Pfau-Wagenbauer, M. and Nejdl, W. (1992a). Integrating model-based and heuristic features in a real-time expert system for power distribution networks. In *Artificial*

Intelligence for Applications, 1992., Proceedings of the Eighth Conference on, pages 303–309. (cited on page 2)

[Pfau-Wagenbauer and Nejdl, 1992b] Pfau-Wagenbauer, M. and Nejdl, W. (1992b). Integrating model-based and heuristic features in a real-time expert system for power distribution networks. In *Artificial Intelligence for Applications, 1992., Proceedings of the Eighth Conference on*, pages 303–309. (cited on pages 30, 30, 45)

[Poole, 1989] Poole, D. (1989). Normality and faults in logic-based diagnosis. *Proceedings of the 11th international joint conference on Artificial intelligence - Volume 2*, pages 1304–1310. (cited on page 48)

[Rich and Knight, 1990] Rich, E. and Knight, K. (1990). *Artificial Intelligence.* McGraw-Hill Science/Engineering/Math. (cited on pages 24, 35)

[Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*, chapter 1, pages 3–30. Artificial Intelligence. Prentice Hall, 2nd edition. (cited on page 23)

[Schrieber et al., 1999] Schrieber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., van der Velde, W., and Wielinga, B. (1999). *Knowledge Engineering and Management: The CommonKADS Methodology.* The MIT Press. (cited on page 26)

[Thorburn, 1915] Thorburn, W. M. (1915). Occam's razor. *Mind*, 24(2):287–288. (cited on page 48)

[Tomsovic et al., 1987] Tomsovic, K., Liu, C.-C., Ackerman, P., and Pope, S. (1987). An expert system as a dispatchers' aid for the isolation of line section faults. *Power Delivery, IEEE Transactions on*, 2(3):736 –743. (cited on pages 30, 31)

[Turing, 1950] Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):pp. 433–460. (cited on page 23)

[Vale and Machado e Moura, 1993] Vale, Z. A. and Machado e Moura, A. (1993). An expert system with temporal reasoning for alarm processing in power system control centers. *Power Systems, IEEE Transactions on*, 8(3):1307 –1314. (cited on pages 31, 45)

[Vale et al., 1999] Vale, Z. A., Ramos, C., Faria, L., Malheiro, N., Silva, A., and Marques, A. (1999). Sparse-an intelligent alarm processor for portuguese transmission control centres. *Human Interfaces in Control Rooms, Cockpits and Command Centres, 1999. International Conference on*, pages 446 –451. (cited on pages 2, 19, 47)

[Wang et al., 1998] Wang, Z., Liu, Y., and Griffin, P. J. (1998). A Combined ANN and expert System Tool for Transformer Fault Diagnosis. *IEEE Transactions on Power Delivery*, 13(4):1224–1229. (cited on pages 32, 33)

[Wei et al., 2011] Wei, L., Guo, W., Wen, F., Ledwich, G., Liao, Z., and Xin, J. (2011). An online intelligent alarm-processing system for digital substations. *Power Delivery, IEEE Transactions on*, 26(3):1615 –1624. (cited on page 2)

[Zadeh et al., 1996] Zadeh, L., Klir, G., and Yuan, B. (1996). *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers.* Advances in Fuzzy Systems. World Scientific. (cited on page 35)

[Zhanjun et al., 2012] Zhanjun, G., Nuo, G., Lei, W., and Zhaofei, L. (2012). Power system fault diagnosis based on power grid. In *Developments in Power Systems Protection, 2012. DPSP 2012. 11th International Conference on*, pages 1 –4. (cited on pages 40, 40, 45)