University of Nottingham
UK | CHINA | MALAYSIA

Lloyd's Register Foundation

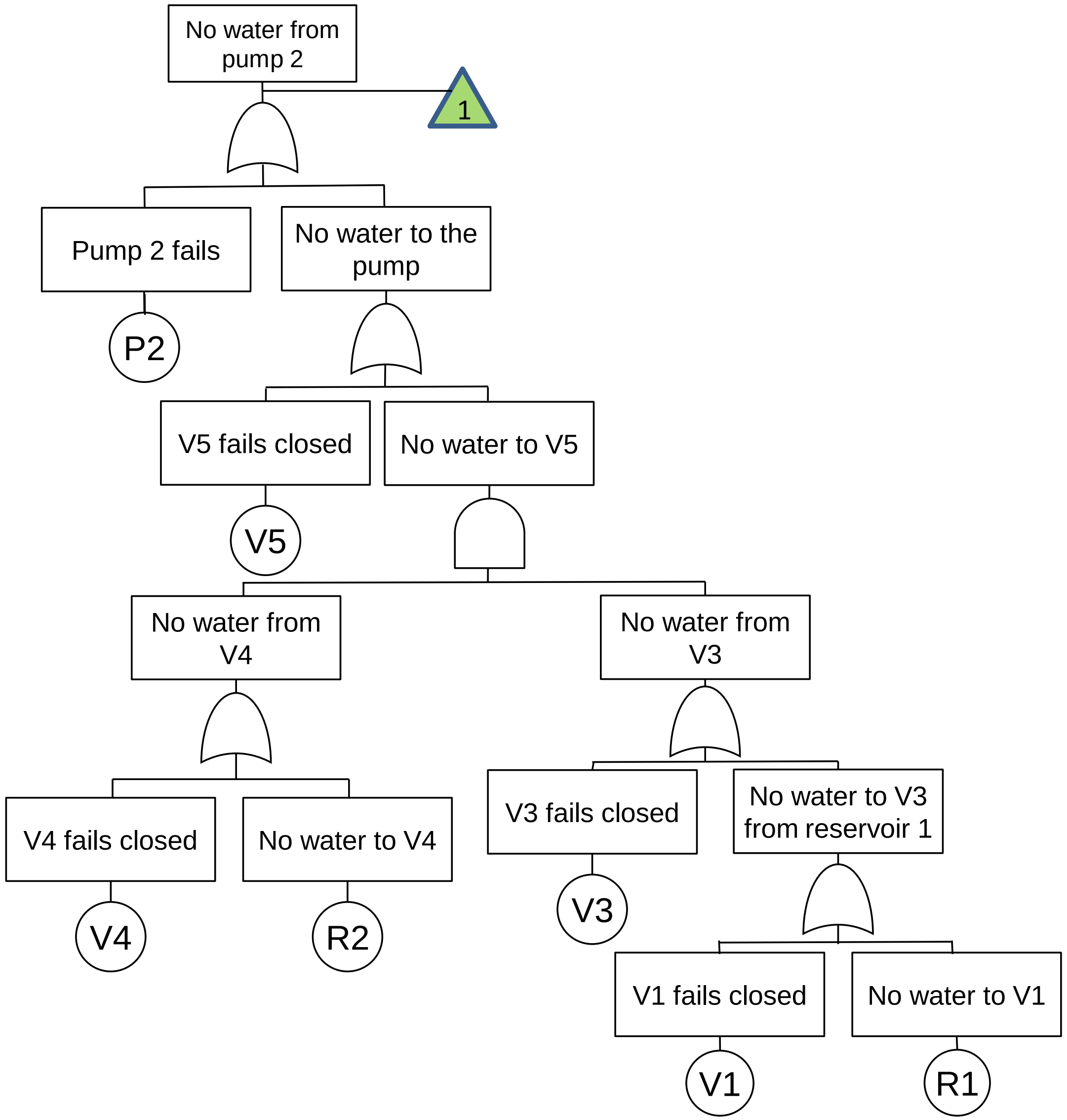
Dependent and Dynamic Fault Tree Analysis

John Andrews

Reliability Lecture Day
Durham University

9th June 2022

Fault Tree Analysis



Component failure models

- Limited maintenance process detail
  - No Repair: $Q(t) = F(t) = 1 - e^{-\lambda t}$
  - Revealed: $Q(t) = \frac{\lambda}{\lambda + \nu}\left(1 - e^{-(\lambda+\nu)t}\right)$
  - Unrevealed: $Q_{AV} = \lambda\left(\frac{\theta}{2} + \tau\right)$
- Snap-shot in time

PROJECT AIMS

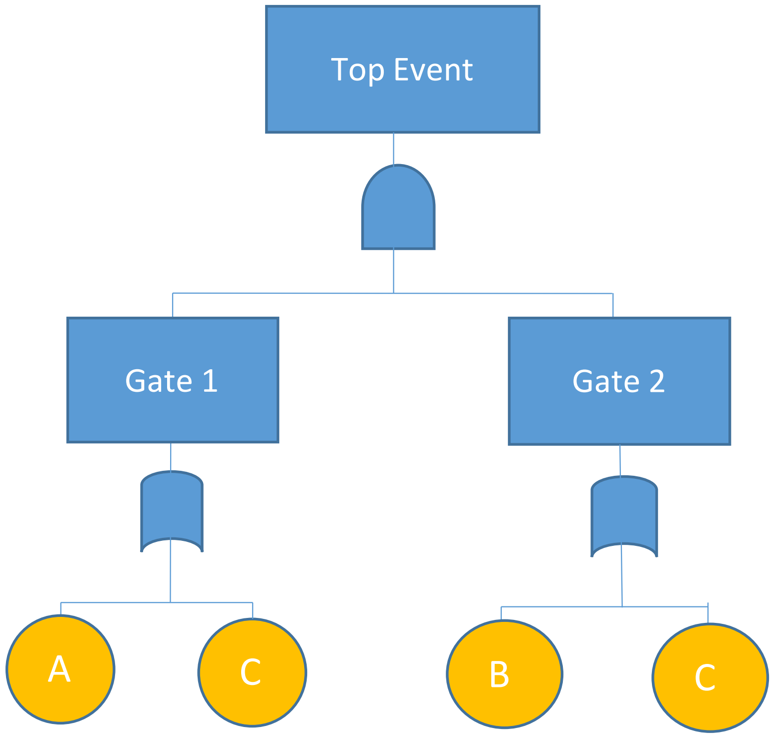
- Incorporate:
  - non-constant failure rates
  - dependent events
  - dynamic features
  - highly complex maintenance strategies

University of Nottingham
UK | CHINA | MALAYSIA

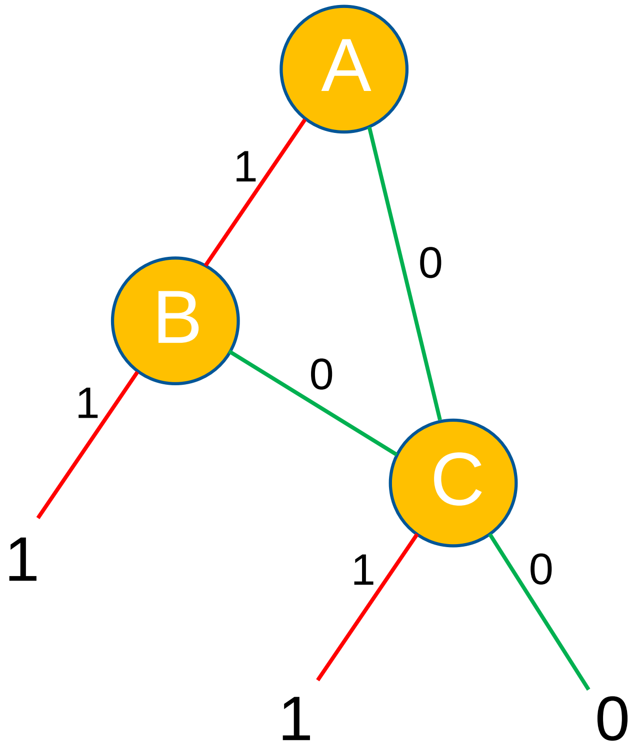
Fault Tree Quantification

Binary Decision Diagrams (BDDs)

Binary Decision Diagrams – Top Event Probability



ORDERING A < B < C



\+ OR

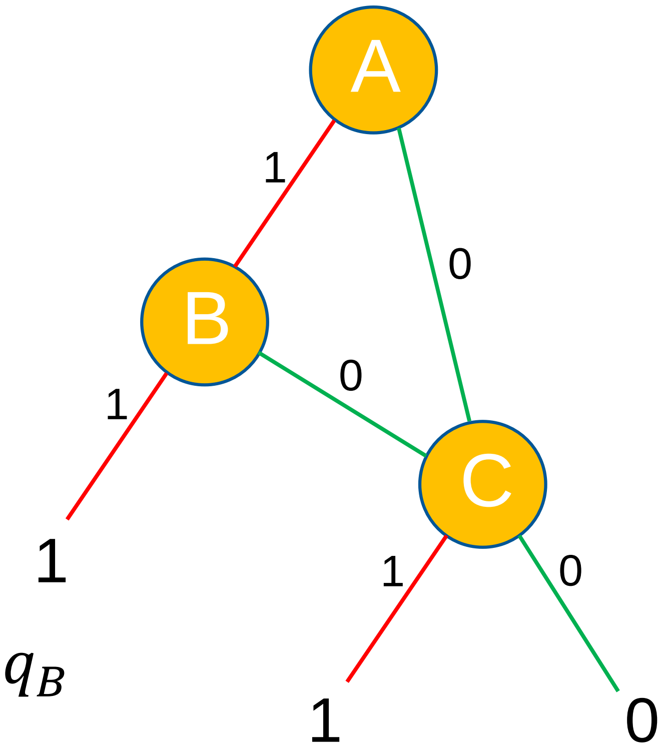
. AND

$$TOP = A.B + C$$

Min Cut Sets: {C}, { A, B}

$$TOP = A.B + A.\bar{B}.C + \bar{A}.C$$

Binary Decision Diagrams – Top Event Probability



$q_A \, q_B$

$$q_A(1-q_B)q_C$$
$$+(1-q_A)\, q_C$$

$$Q_{SYS} = q_A \, q_B + q_A(1-q_B)q_C + (1-q_A)\, q_C$$
$$= q_A \, q_B + q_C - q_A \, q_B \, q_C$$

- Exact
- Fast
- Efficient

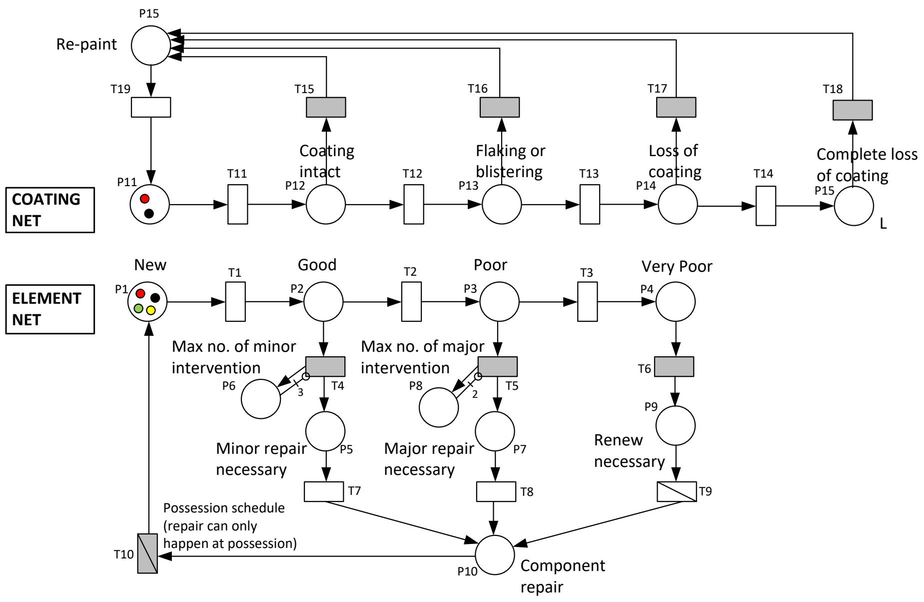
No need to derive the Min Cut Sets as an intermediate step

Modelling Complexities / Dependencies

Petri Nets / Markov Methods

Modelling Methodology

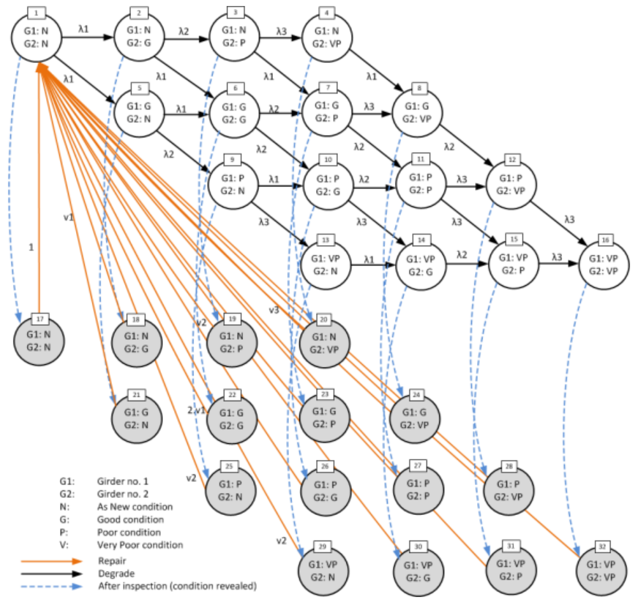
Petri-Net modelling (1962)



Features

- Any distribution of times to transition
- Capable of modelling very complex maintenance strategies / dynamics / dependencies
- Concise structure
- Solution by Monte Carlo simulation
- Produces distributions of durations and no of incidences of different states
- Modular – can form 'system' model by linking asset models

Markov modelling (1906)



Assumes:

- The future condition depends only on the current condition and not the history
- Constant rates of transition

Features

- System states commonly defined by all component states
- Difficult to model decisions based on condition
- Cannot combine asset models to form a 'system' model

Characteristics

Whole system modelling can be challenging

Model Size

- Models can become large for full system analysis
  - State-space explosion for Markov models

Model Solution Times

- Models solution can be CPU intensive
  - Monte Carlo Simulation analysis for Petri Nets can have long convergence times when systems are large or system failures are rare

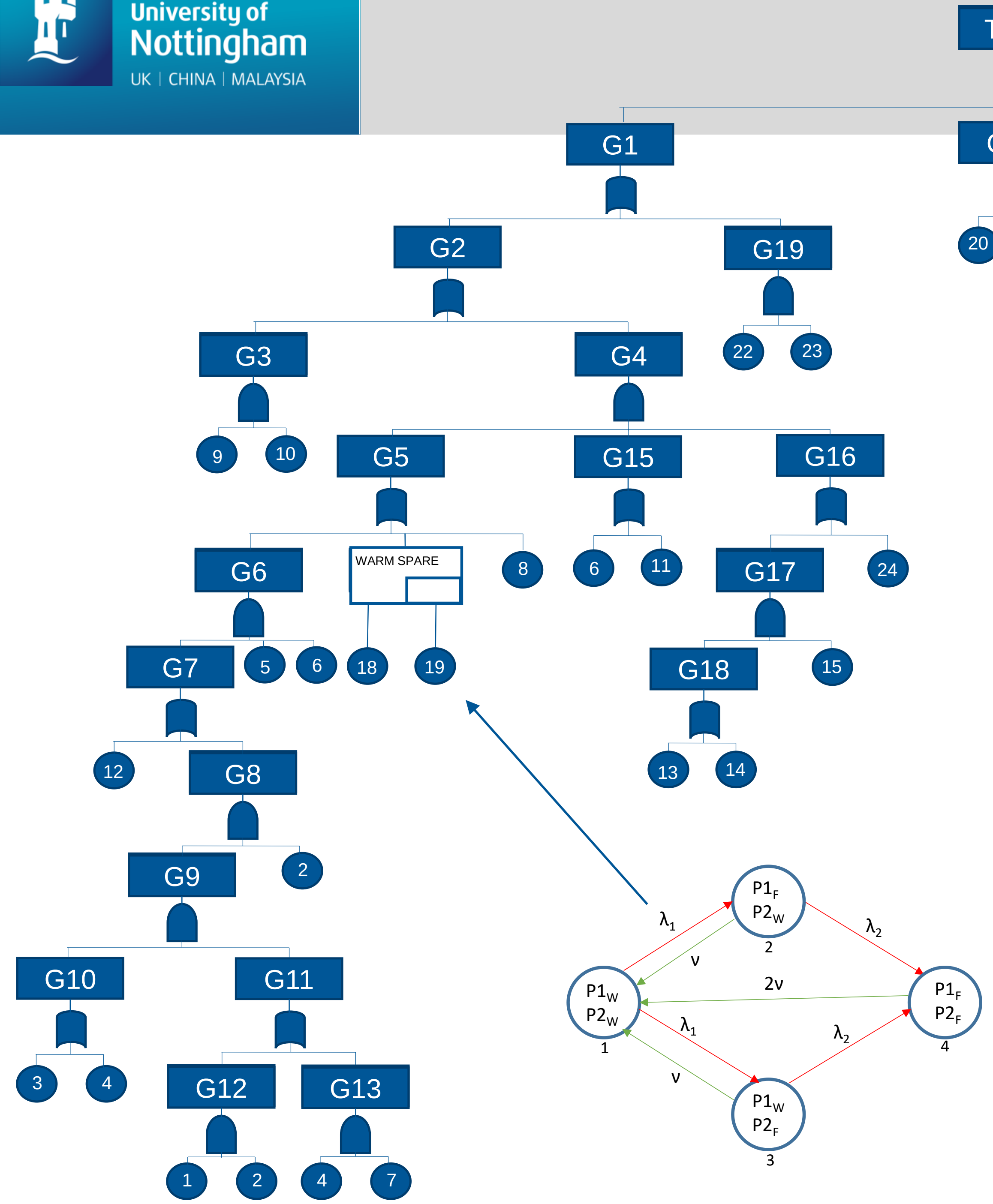
Auditability

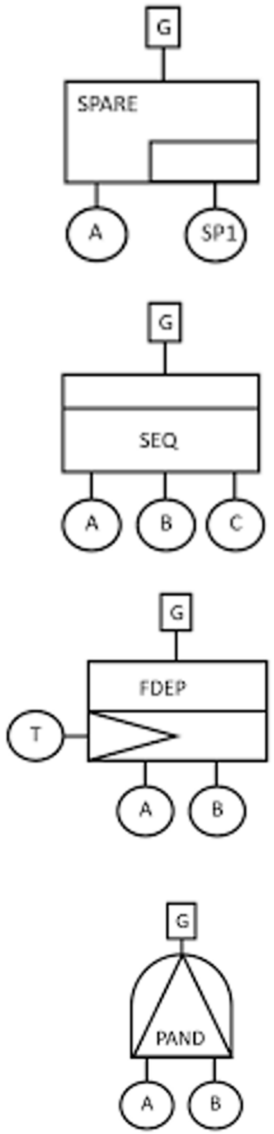
- Lack the causality structure of Fault Trees
  - Peer review and auditing difficult for regulators

University of Nottingham
UK | CHINA | MALAYSIA

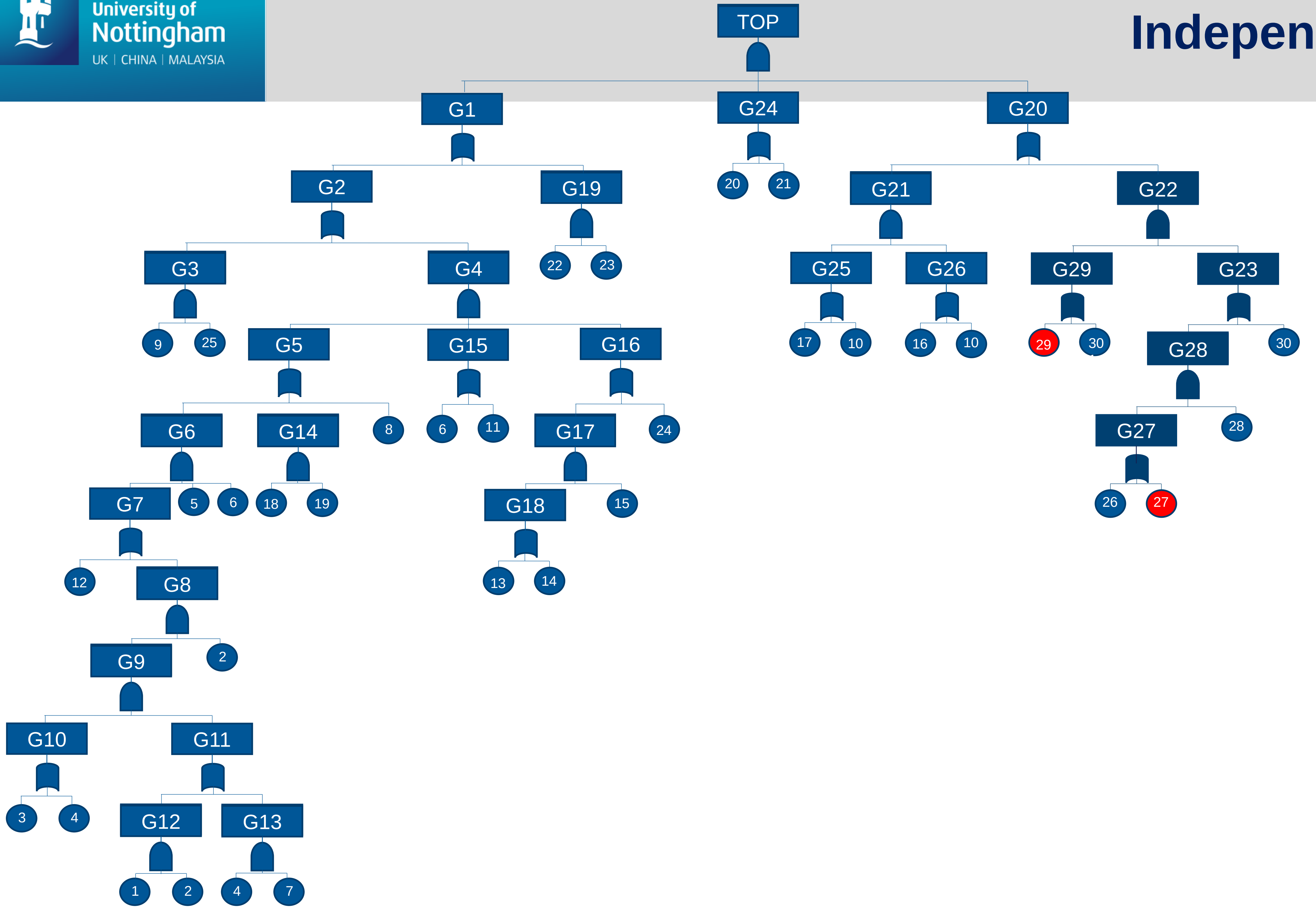
FTA Approaches to Modelling Complexities and Dependencies

Dynamic Fault Trees



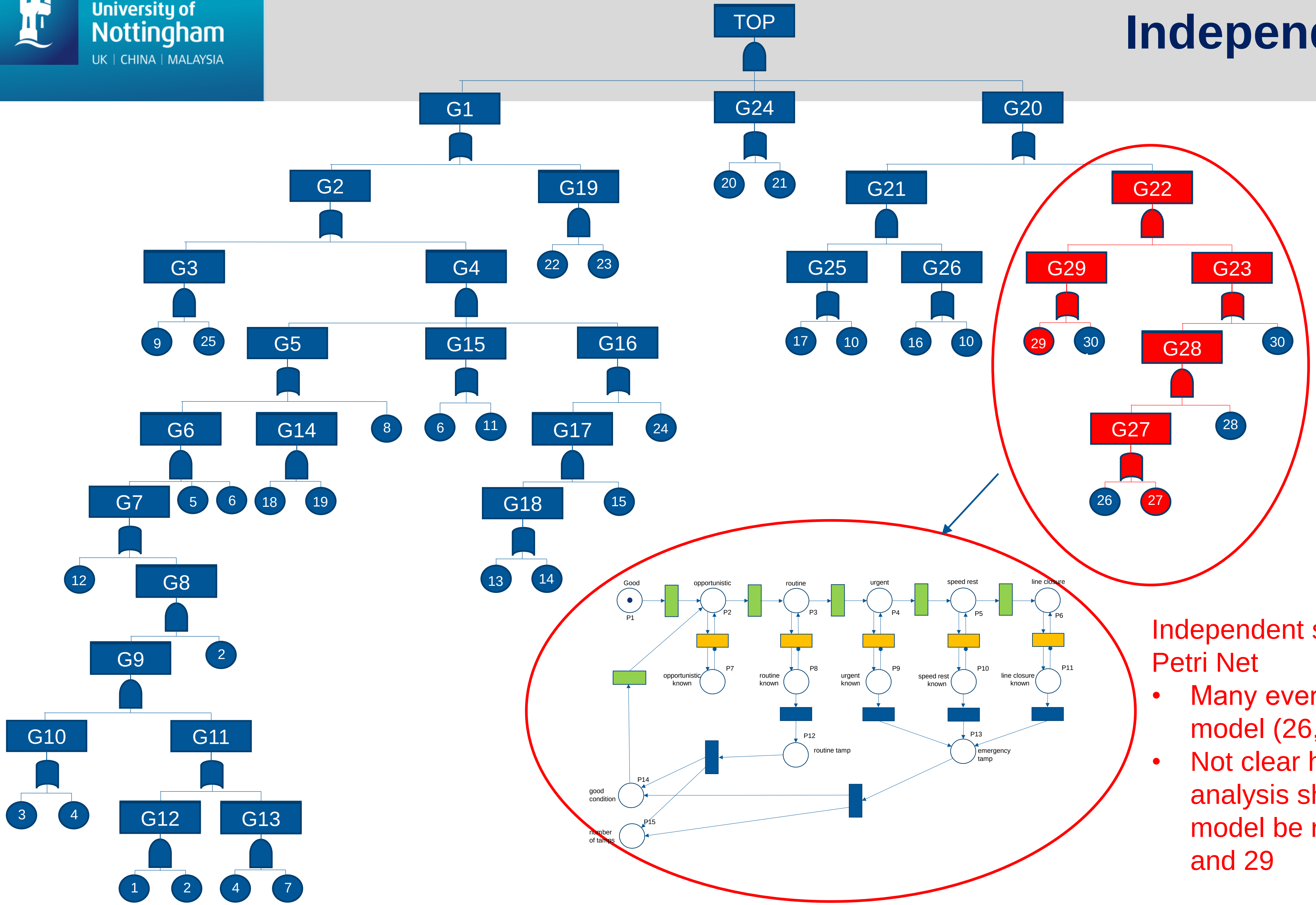


Independent Modules



Dependencies between 27 and 29

Independent Modules

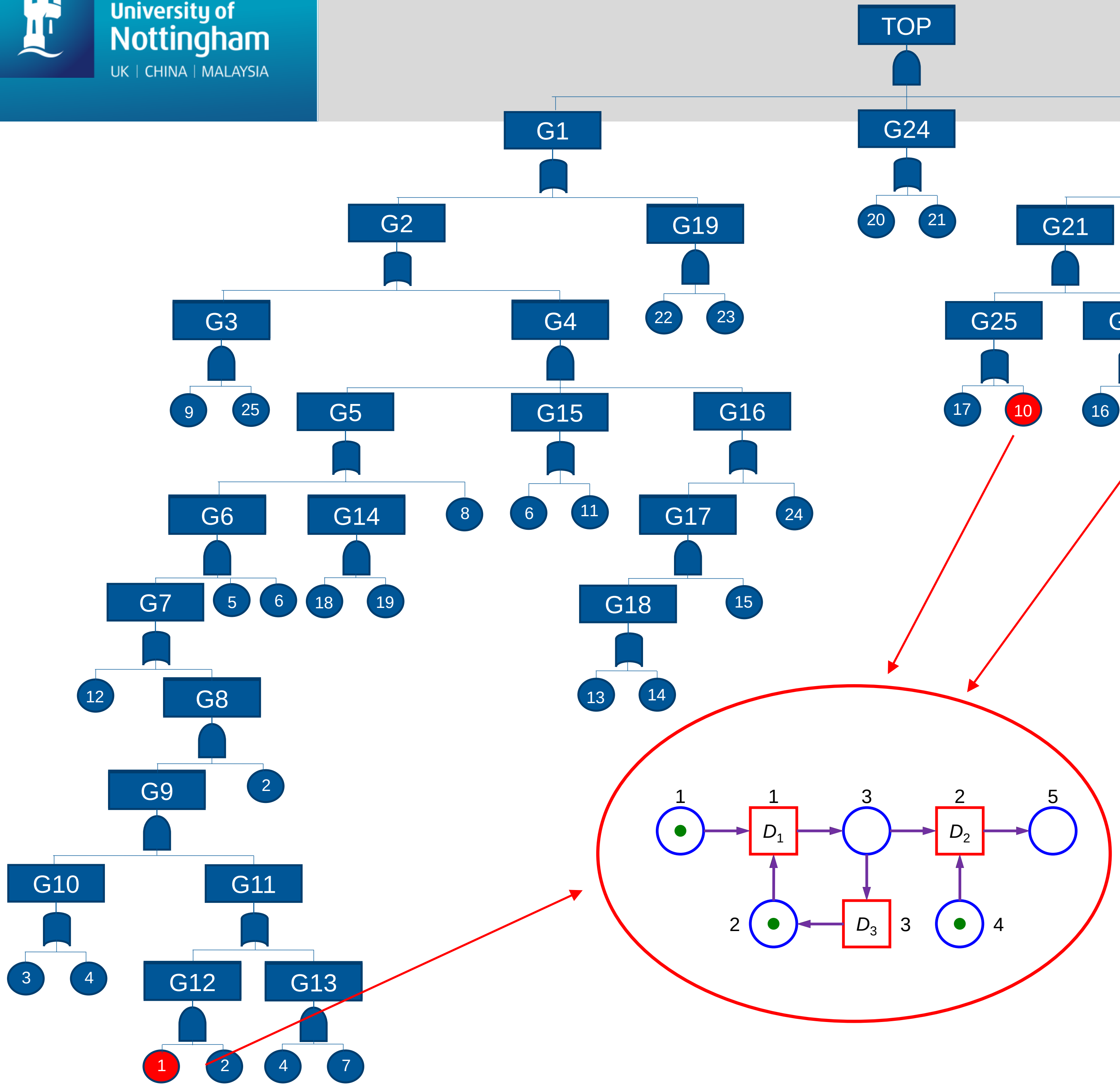


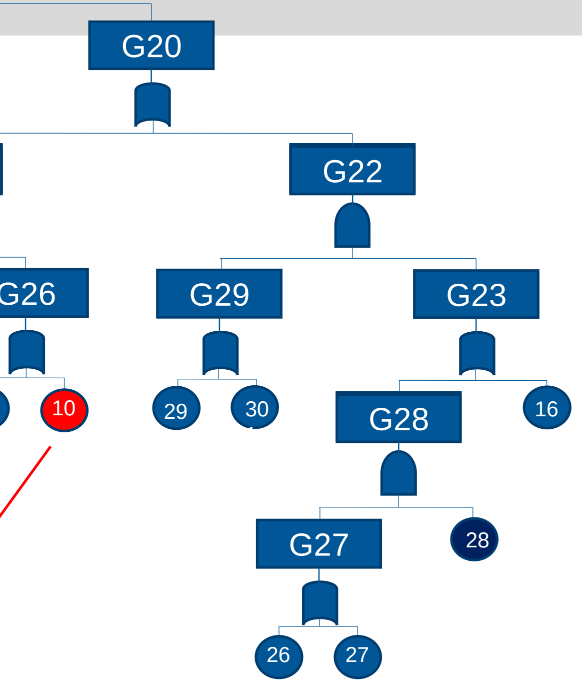
Dependencies between 27 and 29

Independent section solved using a Petri Net

- Many events don’t need to be in this model (26, 28, 30)
- Not clear how to include them in the analysis should the dependency model be reduced to just events 27 and 29

Independent Modules





Maintenance dependency's can affect events which are not geographically close in the FT structure

Small model containing only the dependent events

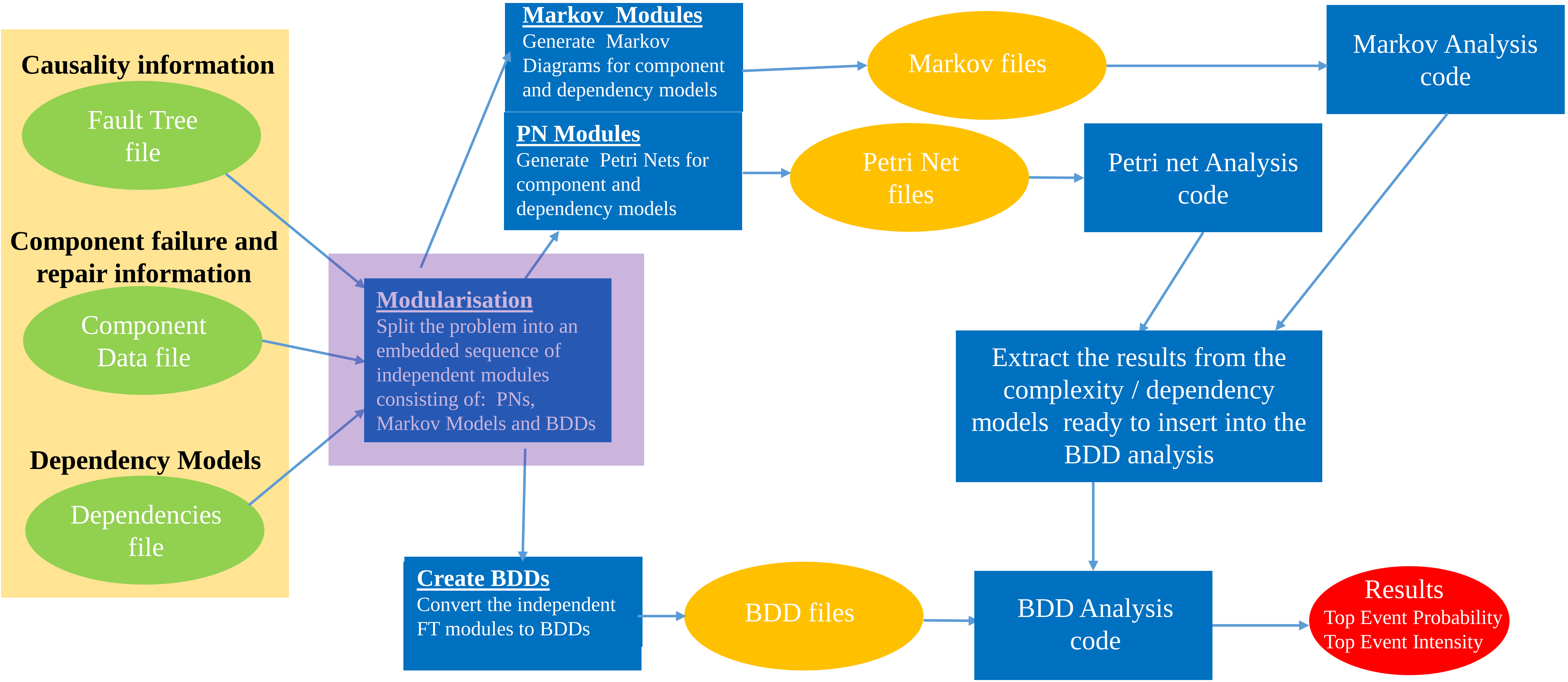
Results integrated back into the assessment of the remainder of the FT

Approach

- Retain the FT to represent the causality of system failure.
  - Exploit the characteristics of the BDDs for FT Analysis
    - Dependencies are just required to be considered on each path
      - Path numbers can be very high so every effort needs to be made to *minimise the size of the BDD*
        - effective variables ordering
        - make the smallest size of fault tree using an effective modularisation

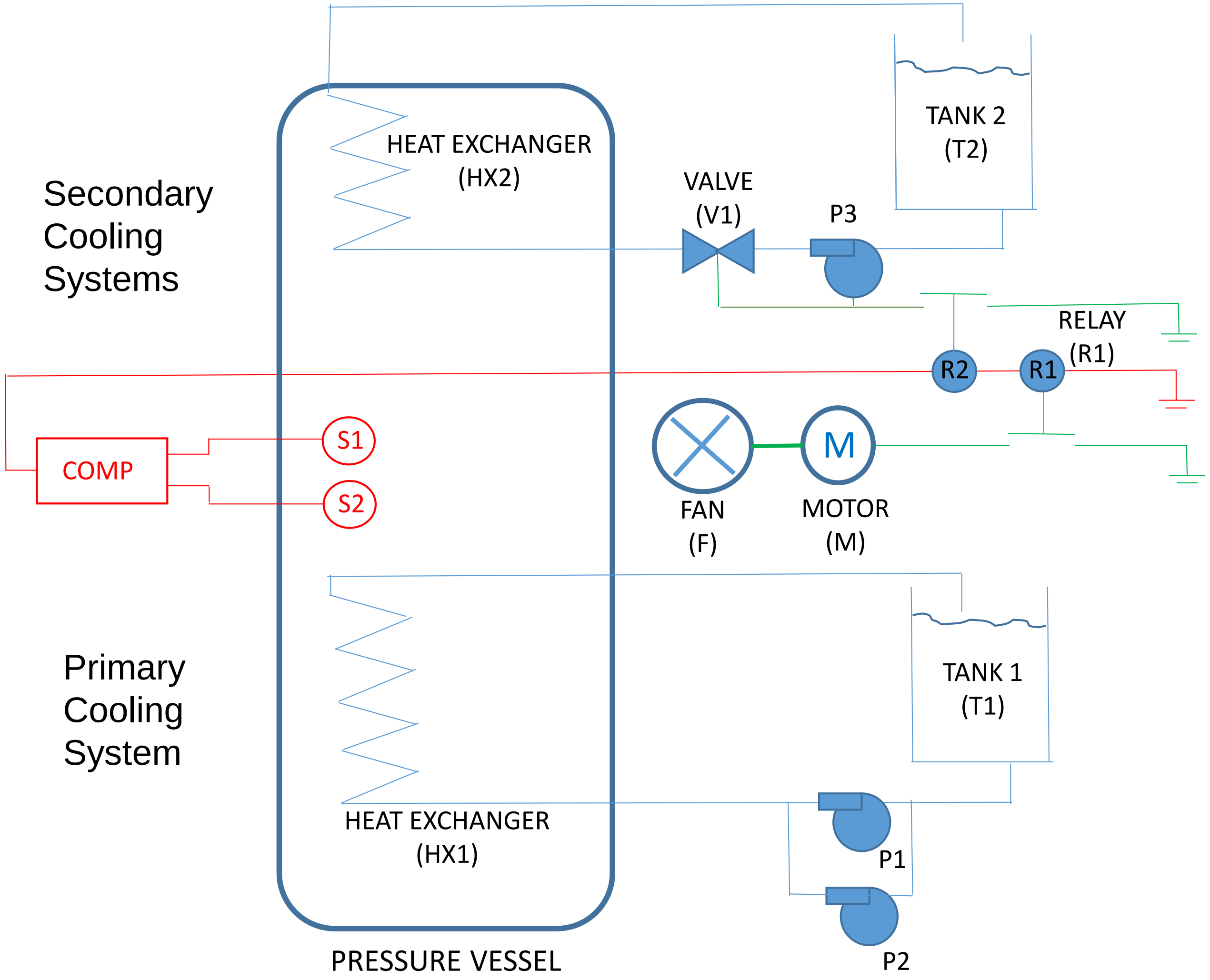
- Model the dependencies and complexities using Petri Nets or Markov as appropriate.
  - No matter where or how many of the dependent basic events occur in the FT - the *simplest dependency model* is used for those events alone

Basic Structure of the Code



Case Study

Plant Cooling System and Features

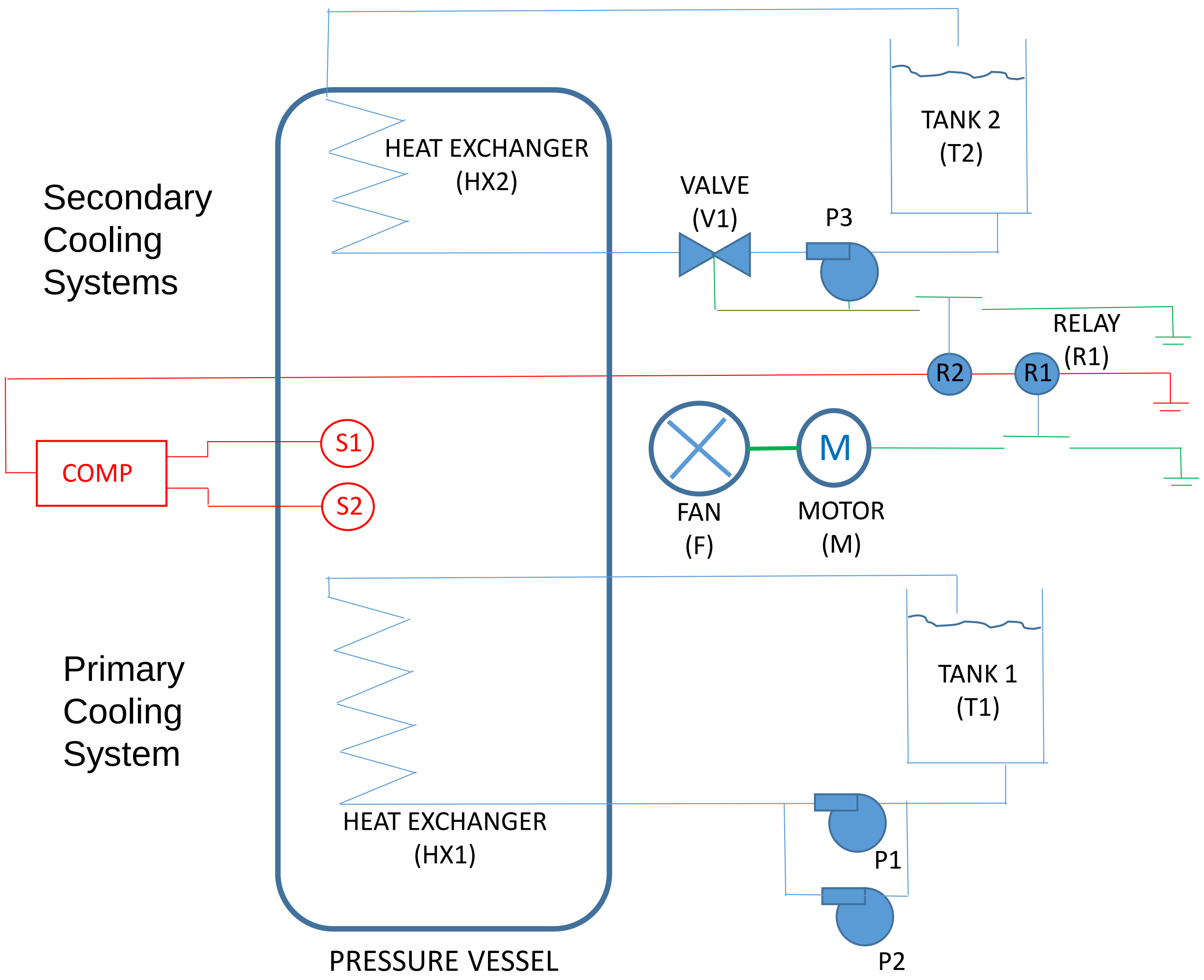


Power supply to all pumps and the valve – PoW

Sub-Systems

- Primary Cooling Water System
  - Tank (T1), Pumps (P1,P2), Heat Exchanger (Hx1), Power Supply (PoW)
- Detection System
  - Sensors (S1,S2), Computer (Comp)
- Secondary Cooling Water System
  - Tank(T2), Pump (P3), Heat Exchanger (Hx2), Valve (V1), Relay (R2), Power Supply (PoW)
- Secondary Cooling Fan System
  - Fan (F), Motor (M), Relay (R1)

Plant Cooling System and Features



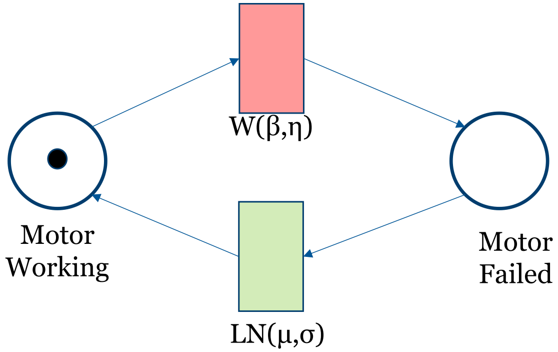
Power supply to all pumps and the valve – PoW

Complex Features

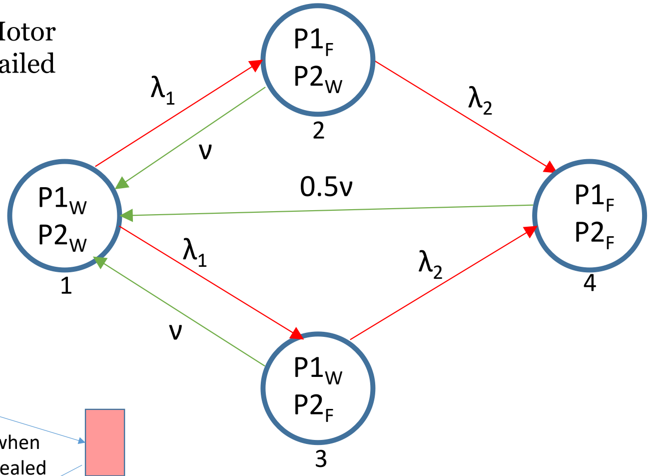
- Non-constant failure / repair rates
  - Motor M - Weibull failure time distribution and a lognormal repair time distribution
- Dependencies
  - Pumps P1 & P2 – if one fails it puts increased load (and increases the failure rate) of the other
  - Heat Exchangers Hx1 & Hx2 - when one needs replacement – needs specialist equipment and both are replaced
  - Pump P3 - two events P3S and P3R are clearly dependent

Complexity and Dependency Models

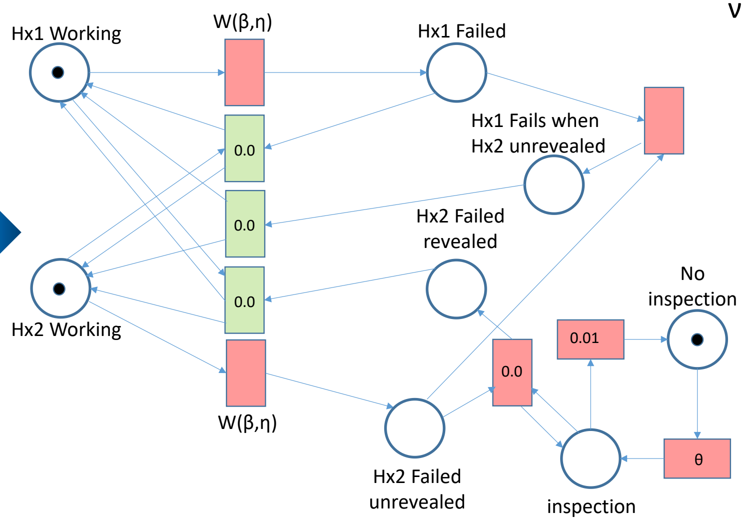
- Non-constant failure / repair rates
  - Motor M - Weibull failure time distribution and a lognormal repair time distribution



- Dependencies
  - Pumps P1 & P2 – if one fails it puts increased load (and increases the failure rate) of the other



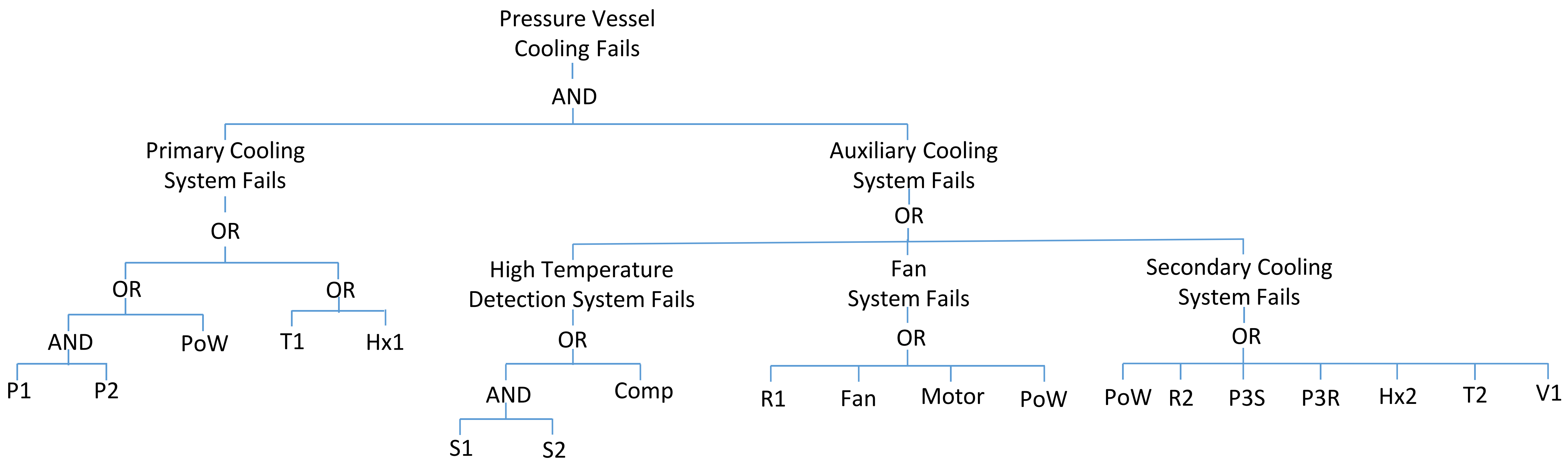
  - Heat Exchangers Hx1 & Hx2 - when one needs replacement – needs specialist equipment and both are replaced



  - Pump P3 - two events P3S and P3R are clearly dependent

$$q_{P3} = q_{P3S} + (1.0 - q_{P3S})\lambda_{P3R} t_{period}$$
$$= 0.05 + 0.095 \times 10^{-4} \times 30$$
$$= 0.05285$$

Fault Tree Structure



University of Nottingham
UK | CHINA | MALAYSIA

Modularisation

Modified Faunet and LT Algortihm

Faunet

Three phased repeatedly applied:

- Contraction

  Subsequent gates of the same type are contracted into a single gate

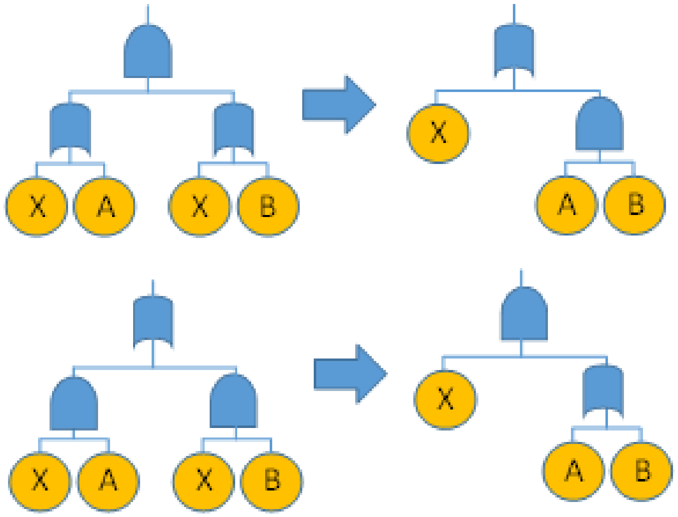
- Factorisation

  Extracts factors expressed as groups of events that always occur together in the same gate type. The factors can be any number of events if they satisfy the following:

  - All events in the group are independent and either initiators or enablers.
  - All events in the group feature a dependency and contain all events in the same dependency group.

- Extraction

  Restructure:



Quantification of the factors

For combinations formed from independent events

OR combinations, $Cf_i = x_1 + x_2 + \cdots x_n$

$$Q_{Cfi} = 1 - \prod_{j=1}^{n} \left(1 - q_{x_j}\right)$$

If the factor contains only initiating events:

$$w_{Cfi} = \sum_{j=1}^{n} w_j \prod_{\substack{k=1 \\ k \neq j}}^{n} \left(1 - q_{x_k}\right)$$

AND combinations, $Cf_i = x_1 . x_2 . \ldots . x_n$

$$Q_{Cfi} = \prod_{j=1}^{n} q_{x_j}$$

$$w_{Cfi} = \sum_{\substack{j=1 \\ initiators}}^{n} \left(w_j \prod_{\substack{k=1 \\ k \neq j}}^{n} q_{x_k} \right)$$

Quantification of the factors

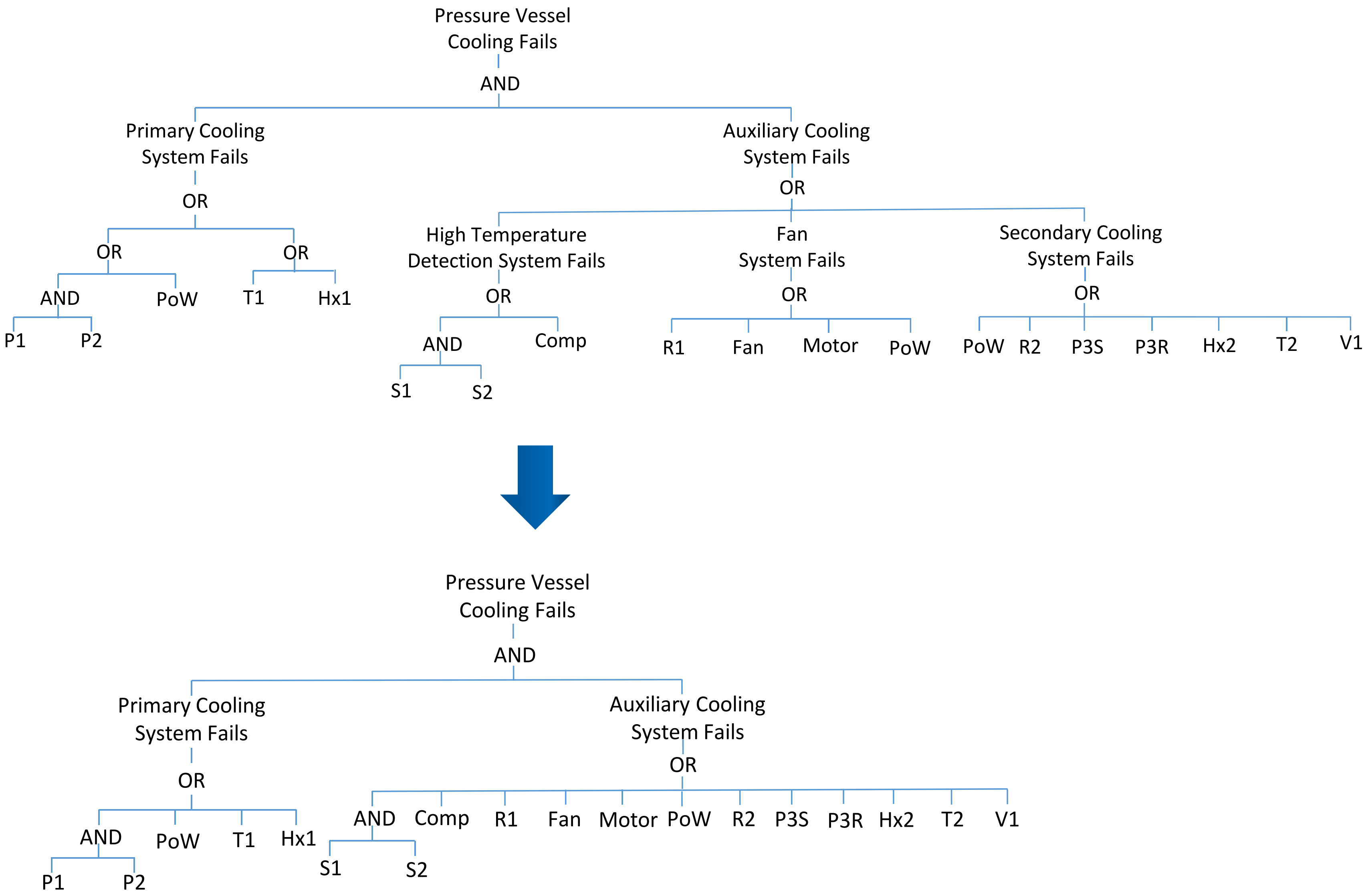
For combinations of events from a dependency group

OR combinations, $Cf_i = x_1 + x_2 + \cdots x_n$

AND combinations, $Cf_i = x_1 . x_2 . \dots . x_n$

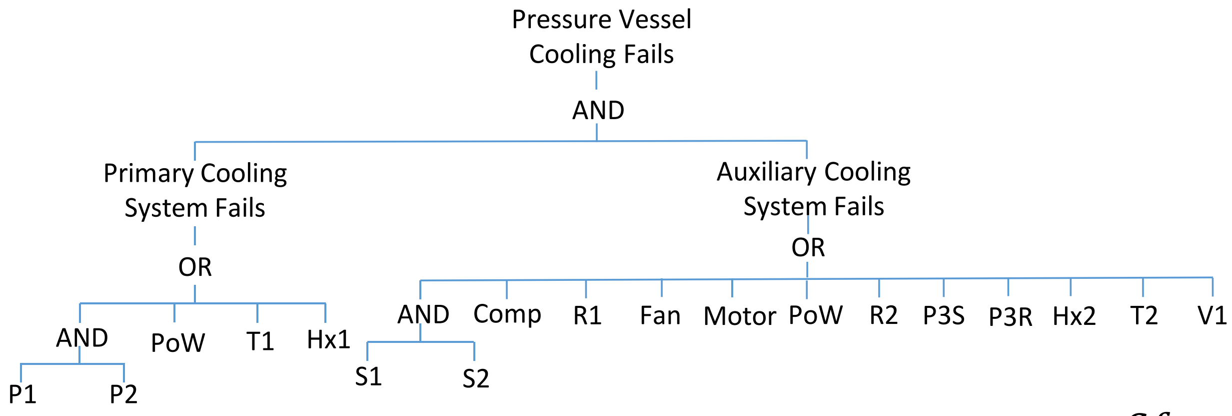
Q_{Cfi}, w_{Cfi} are extracted from the PN / Markov model

Modularisation (1)

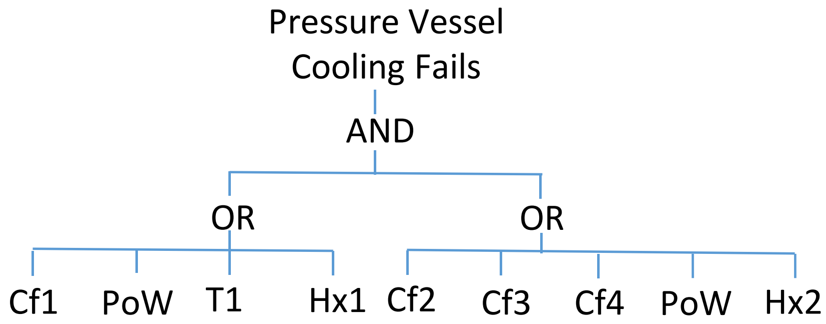


Contraction 1

Modularisation (2)



Factorisation 1



$Cf_1 = P1.P2$

(dependency group D1 – initiators)

$Cf_2 = S1.S2$

(independent enablers)

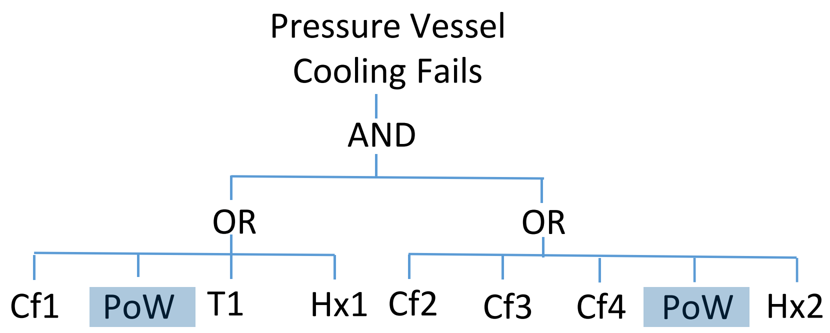
$Cf_3 = Comp + R1 + Fan + Motor + R2 + T2 + V1$

(independent enablers)

$Cf_4 = P3S + P3R$

(dependency group D3 – enablers)

Modularisation (3)



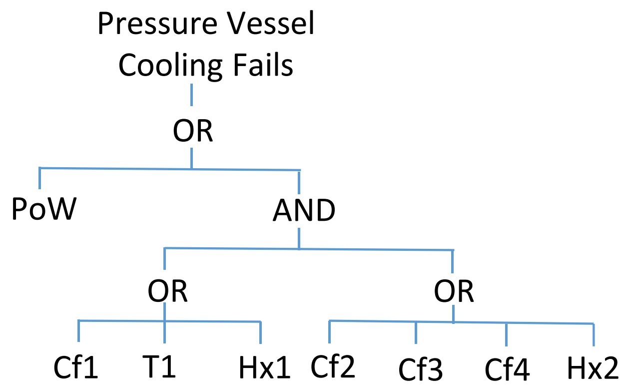
$$Cf_1 = P1.P2$$

$$Cf_2 = S1.S2$$

$$Cf_3 = Comp + R1 + Fan + Motor + R2 + T2 + V1$$

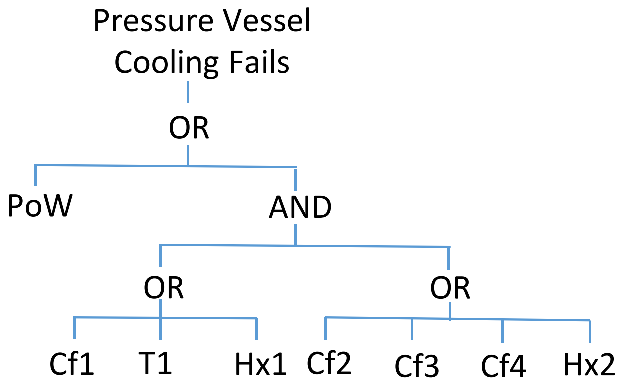
$$Cf_4 = P3S + P3R$$

Extraction 1



Contraction 2 -- No change

Modularisation (4)



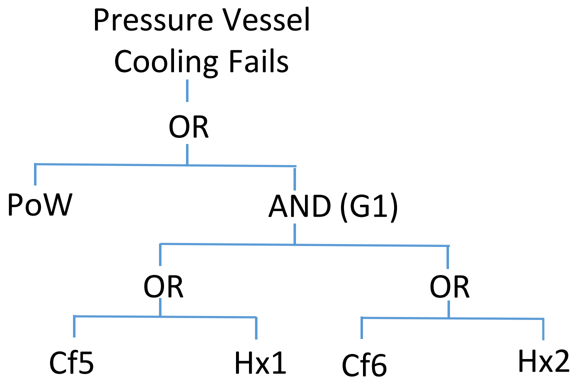
$$Cf_1 = P1.P2$$

$$Cf_2 = S1.S2$$

$$Cf_3 = Comp + R1 + Fan + Motor + R2 + T2 + V1$$

$$Cf_4 = P3S + P3R$$

Factorisation 2

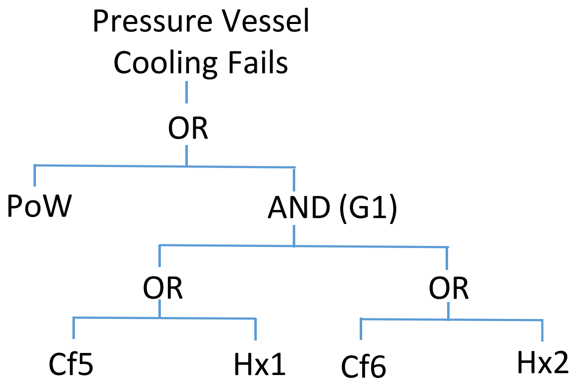


$$Cf_5 = Cf_1 + T1$$

$$Cf_6 = Cf_2 + Cf_3 + Cf_4$$

Simplest possible Faunet representation

Modularisation (5)



$$Cf_1 = P1.P2$$

$$Cf_2 = S1.S2$$

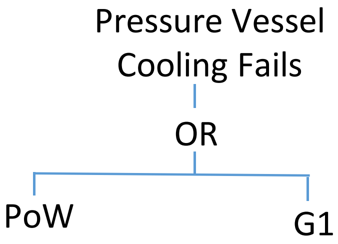
$$Cf_3 = Comp + R1 + Fan + Motor + R2 + T2 + V1$$

$$Cf_4 = P3S + P3R$$

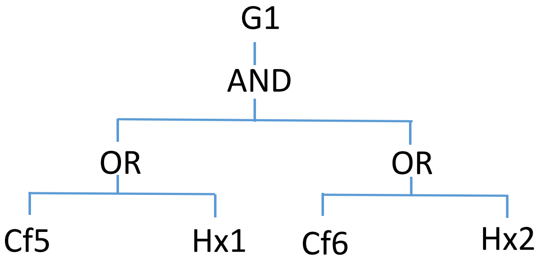
$$Cf_5 = Cf_1 + T1$$

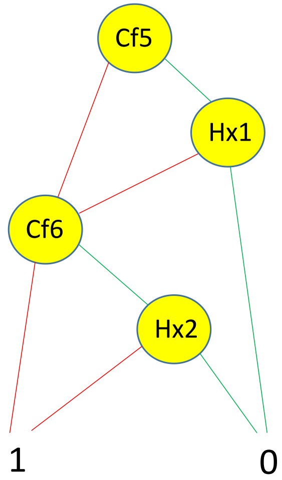
$$Cf_6 = Cf_2 + Cf_3 + Cf_4$$

Applying the Rauzy & Dutuit algorithm gives independent section Top and G1



$$Cf_7 = PoW + G1$$





Conclusions

- Dynamic and Dependent Tree Theory, D^2T^2, enables the evaluation of fault trees which are not limited by the restrictions which apply to conventional fault trees solved by Kinetic Tree Theory.

- Retains the familiar and popular fault tree causality structure.

- Utilises BDDs, Petri Nets and Markov Models.

- The Petri net and Markov models dedicated to solve the complexities and dependencies are minimal in size.

- Modularisation of the fault tree minimises the size of the BDD utilised in the system evaluation (and therefore the number of paths).

Thank you for your attention

Any Questions?