Dependable Systems

Component-Based Dependability Modeling

Dr. Peter Tröger

Sources:

Eusgeld, Irene et al.: Dependability Metrics. 4909. Springer Publishing, 2008
Menasce, Daniel A.; Almeida, Virgilio A.: Capacity Planning for Web Services: Metrics, Models, and Methods. Prentice Hall, 2002. , 0-13-065903-7
Krishna B. Misra: Handbook of Performability Engineering. Springer. 2008
www.fault-tree.net
http://www.hq.nasa.gov/office/codeq/doctree/fthb.pdf

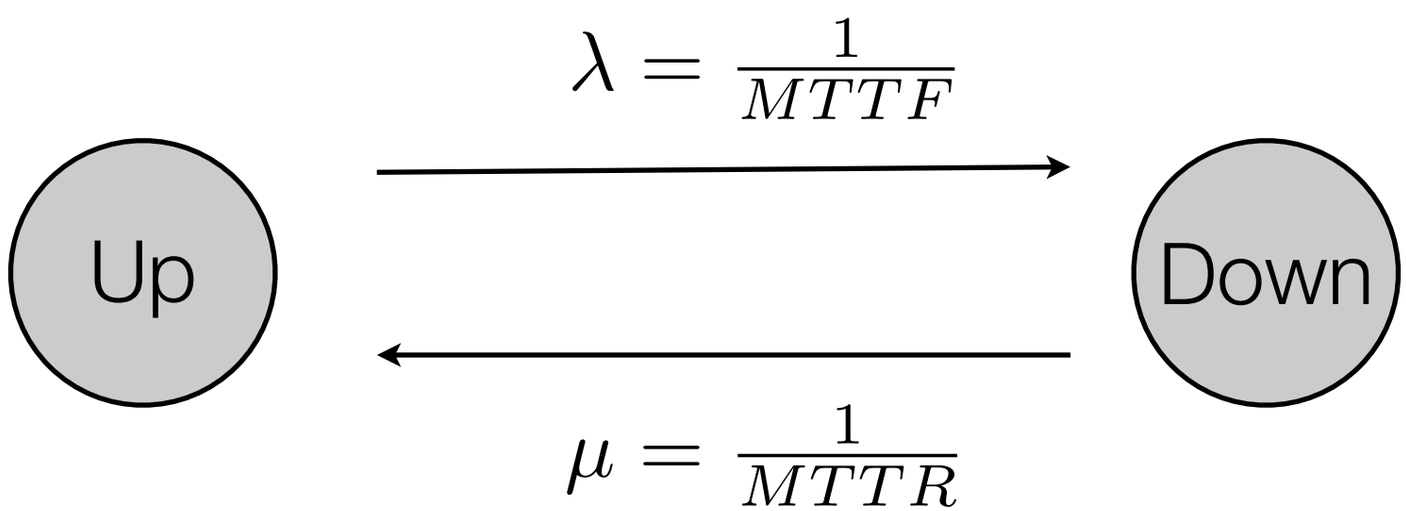
Dependability Modeling

- Default approach: Utilize a formalism to model system dependability
  - Quantify the availability of components, calculate system availability based on this data and a set of assumptions (the availability model)
    - Most models expose the same expressiveness
    - Each formalism allows to focus on certain aspects
      - Component-based models: Reliability block diagram, fault tree
      - State-based models: Markov chain, petri net
- System understanding evolved from hardware to software to IT infrastructures
  - Example: Organization management influence on business service reliability
    - Information Technology Infrastructure Library (ITIL)
    - CoBiT(Control Objectives for Information and related Technology)

Dependability Modeling

- System analysis approaches
  - **Inductive methods** - Reasoning from specific cases to a general conclusion
    - Postulate a particular fault or initiating event, find out system effect
    - Determine what system (failure) states are possible
    - Examples: Failure Mode and Effect Analysis (FMEA), Preliminary Hazards Analysis (PHA), event tree analysis, ...
  - **Deductive methods** - Postulate a system failure, find out what system modes or component behaviors contribute to this failure
    - Determine how a particular system state can occur
    - Examples: Fault Tree Analysis (FTA), Reliability Block Diagrams (RBD)

Dependability Modeling



- Some assumptions
  - All failure and repair events are exponentially distributed
  - Components are either fully working or completely failed
  - All failure and repair events are pair-wisely stochastically independent
  - Correct functioning at t can be treated as event
  - r expresses availability with given MTTF / MTTR, or reliability at one point in time

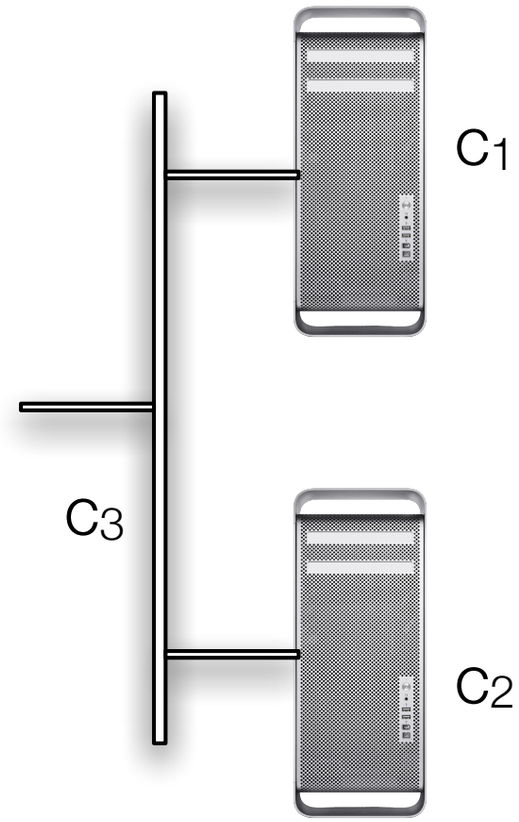
Boolean Algebra Approach

- For stochastically independent events:

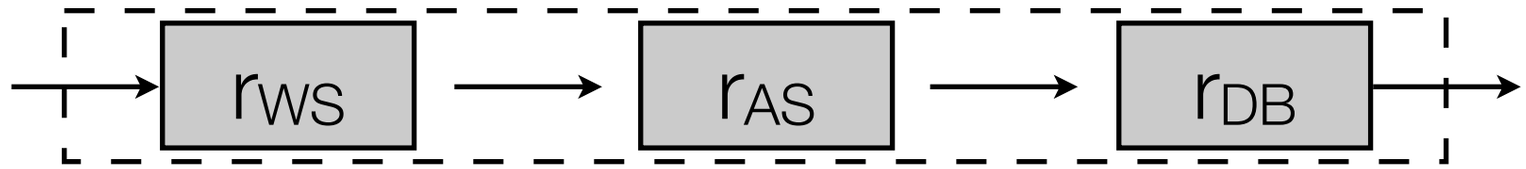
$$Pr(\phi_1 \wedge \phi_2) = Pr(\phi_1) \cdot Pr(\phi_2)$$
$$Pr(\phi_1 \vee \phi_2) = Pr(\phi_1) + Pr(\phi_2) - Pr(\phi_1 \wedge \phi 2)$$
$$Pr(\neg\phi) = 1 - Pr(\phi)$$

- c_i : The event that component c_i is operational
- $r_i = Pr(c_i)$: Probability that c_i occurs

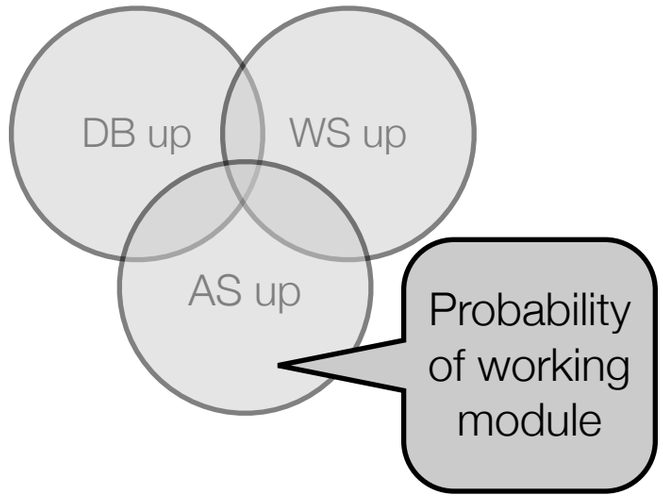
$$\phi = (c_1 \vee c_2) \wedge c_3$$
$$Pr(\phi) = Pr((c_1 \vee c_2) \wedge c3)$$
$$= (r_1 + r_2 - r_1 \cdot r_2) \cdot r_3$$
$$= r_1 r_3 + r_2 r_3 - r_1 r_2 r_3$$



Serial Case



- Help from probability theory: *The probability of an event expressed as the intersection of independent events is the product of the probabilities of the independent events.*



- Example: Chain of web server (r=0.9), application server (r=0.95) and database server (r=0.99)
  - Benefit of replacing the database with an expensive model (r=0.999) ?
  - Benefit of replacing the web server with a new model (r=0.95) ?

$$\phi_S = c_{WS} \wedge c_{AS} \wedge c_{DB}$$

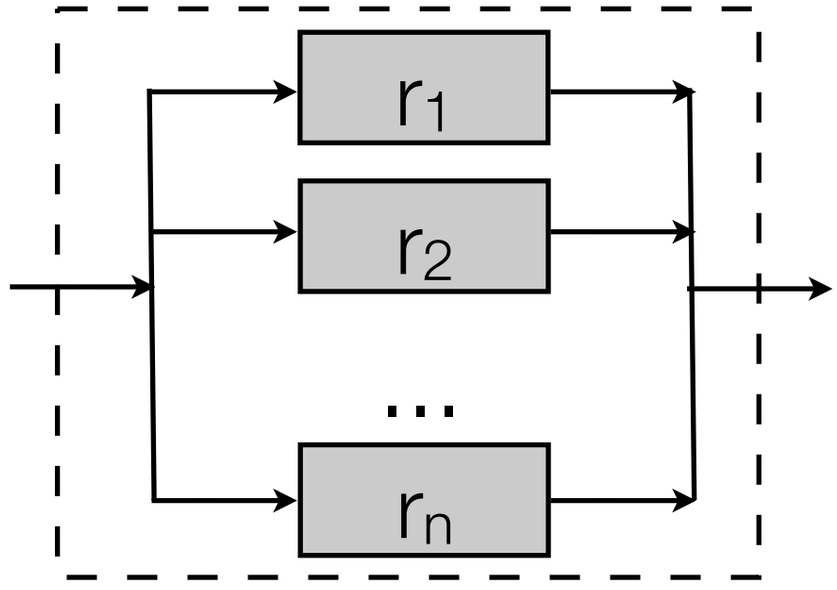
Redundancy structure

Component available

$$R_S = r_1 \times r_2 ... r_n = \prod_{i=1}^{n} r_i$$

Parallel Case

- Parallel case
  - Search engine, cluster node r=0.85 (around 2 months outage / year)
  - How many servers to reach 5 nines of site availability ?



$$\phi_S = c_{WS} \vee c_{AS} \vee c_{DB}$$

Redundancy structure

Component available

$$R_S = 1 - P_{alldown}$$

$$R_S = 1 - ((1 - r_1) \times (1 - r_2) \times ...(1 - r_n))$$

$$R_S = 1 - \prod_{i=1}^{n}(1 - r_i) \qquad n_{min} = \lceil \frac{ln(1-R_S)}{ln(1-r)} \rceil$$

K-of-N Systems

- *At least K out of N components need to work to have a functioning system*
- Algebraic investigation only possible with exponential failure distribution
  - At the beginning, there are N operational units, so failure rate equals $N \cdot \lambda$
  - After first component failure, the failure rate goes down to $(N-1) \cdot \lambda$
  - This goes until the (K+1)th failure has occurred

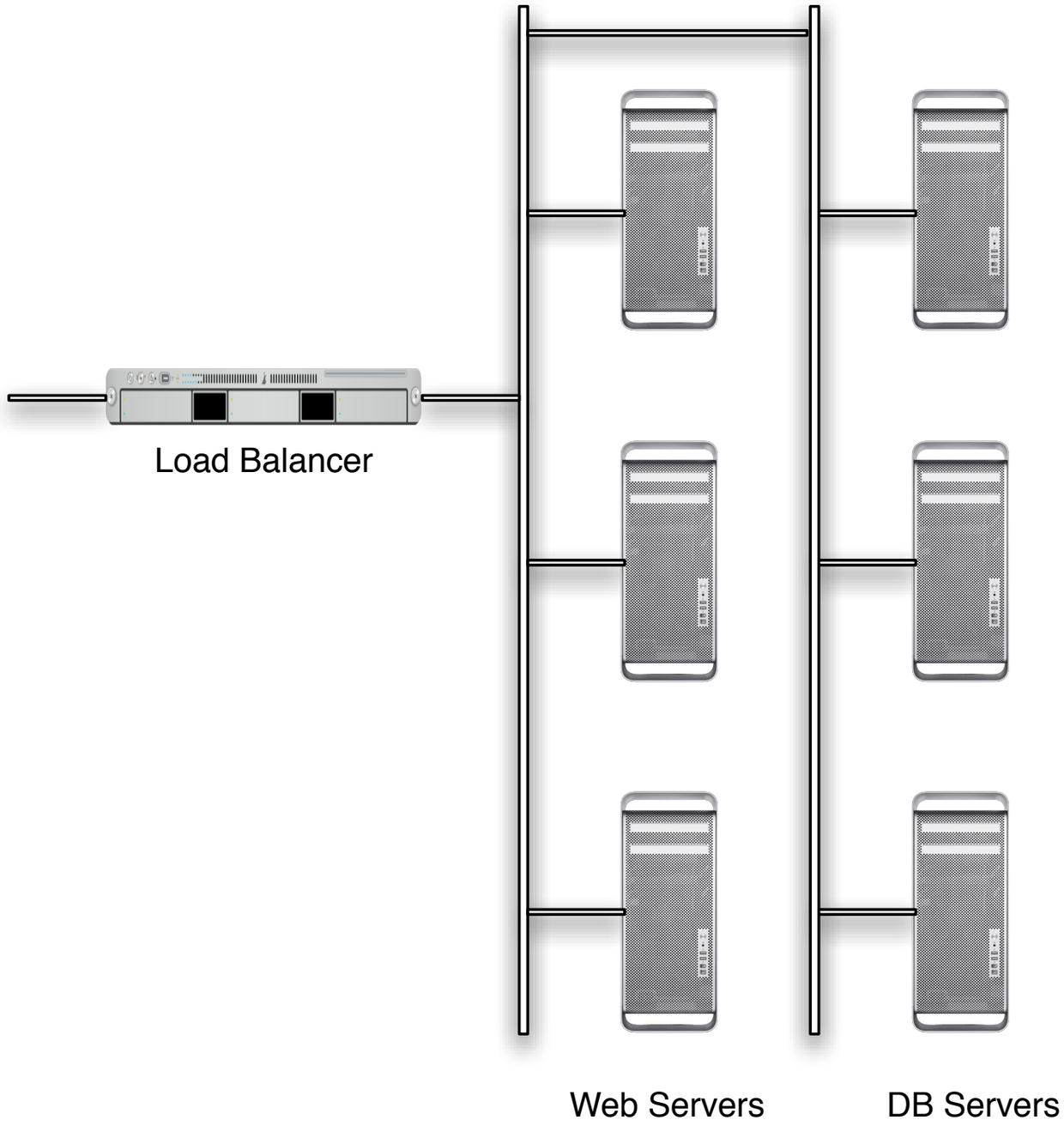
$$MTTF = \frac{1}{\lambda} \sum_{K \leq j \leq N} \frac{1}{j}$$

  - K=1 is the same as the parallel case
  - K=N is the same as the serial case
- Example: Disk RAID system with K=3, N=4, MTTF=1800h, MTTR=4.5h

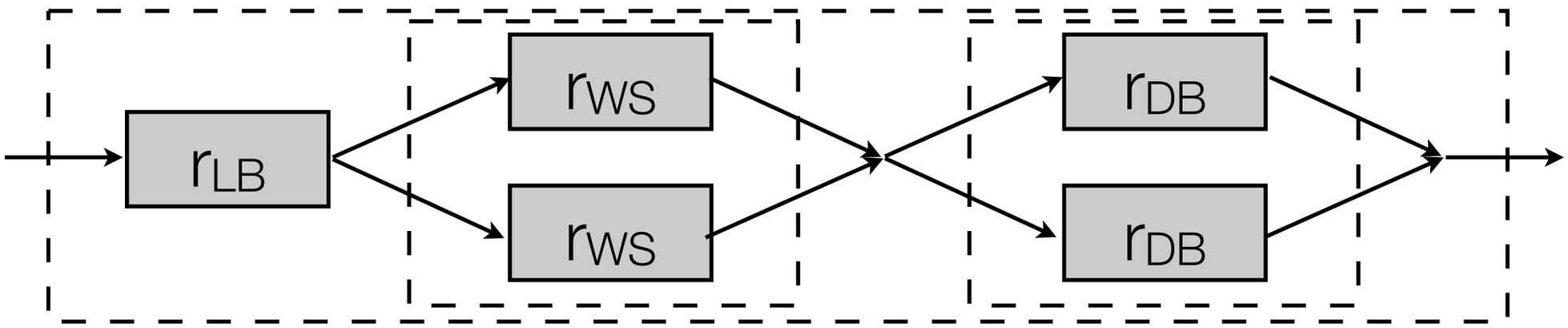
$$A_{Disk} = \frac{1800}{1800+4.5} = 0.9999628 \qquad MTTF = \frac{1}{4} MTTF_{Disk} + \frac{1}{3} MTTF_{Disk} = 1050h$$

Examples

- Online brokerage site to be designed - choice of components needed
- Site availability aimed at 99.99%
- Setup: Load balancer, similar web server hardware, replicated database
- Question: What is the least expensive configuration that reaches 99.99% ?
  - Choice between low-end (r=0.85) and high-end (R=0.999) servers
  - Must also consider purchase and maintenance costs per setup



Examples



$$\phi_S = c_{LB} \wedge (c_{WS1} \vee c_{WS2}) \wedge (c_{DB1} \vee c_{DB2})$$

$$R_{site} = r_{LB} \times R_{WS} \times R_{DB}$$
$$= r_{LB} \times [1 - (1 - r_{WS})^{n_{WS}}] \times [1 - (1 - r_{DB})^{n_{DB}}]$$

$$R_{site} = 0.9999$$
$$R_{LB} = 0.99999$$

$$n_{WS} = \lceil \frac{ln(1-0.99991/[1-(1-r_{DB})^{n_{DB}}}{ln(1-r_{WS})} \rceil$$

Examples

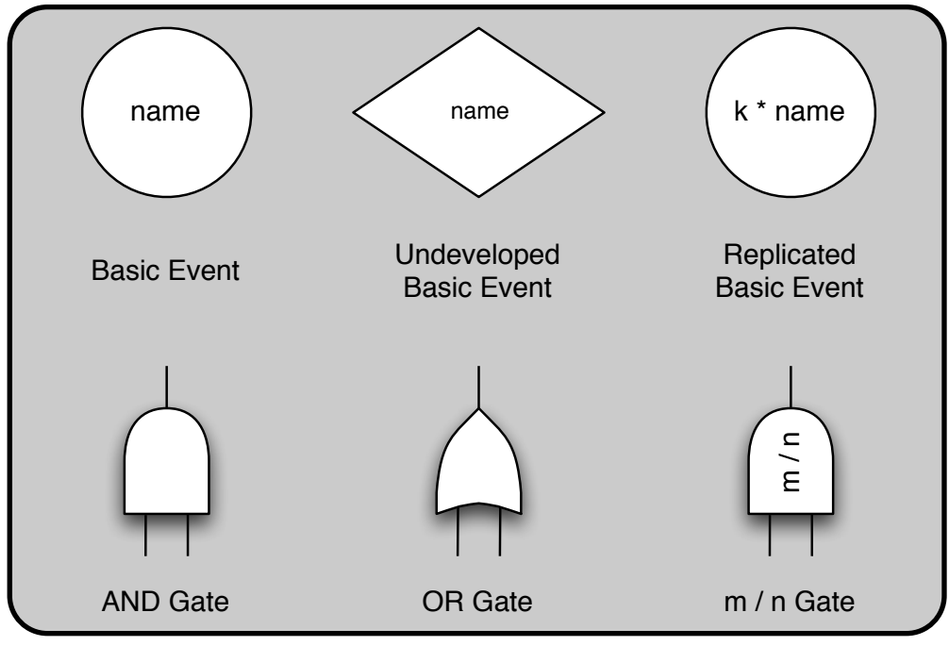
r_{WS}	r_{DB}	Minimum n_{WS}	Minimum n_{DB}	R_{site}
0,85	0,85	6	5	99,990%
0,85	0,999	5	2	99,991%
0,999	0,999	2	2	99,999%

Fault Tree Analysis

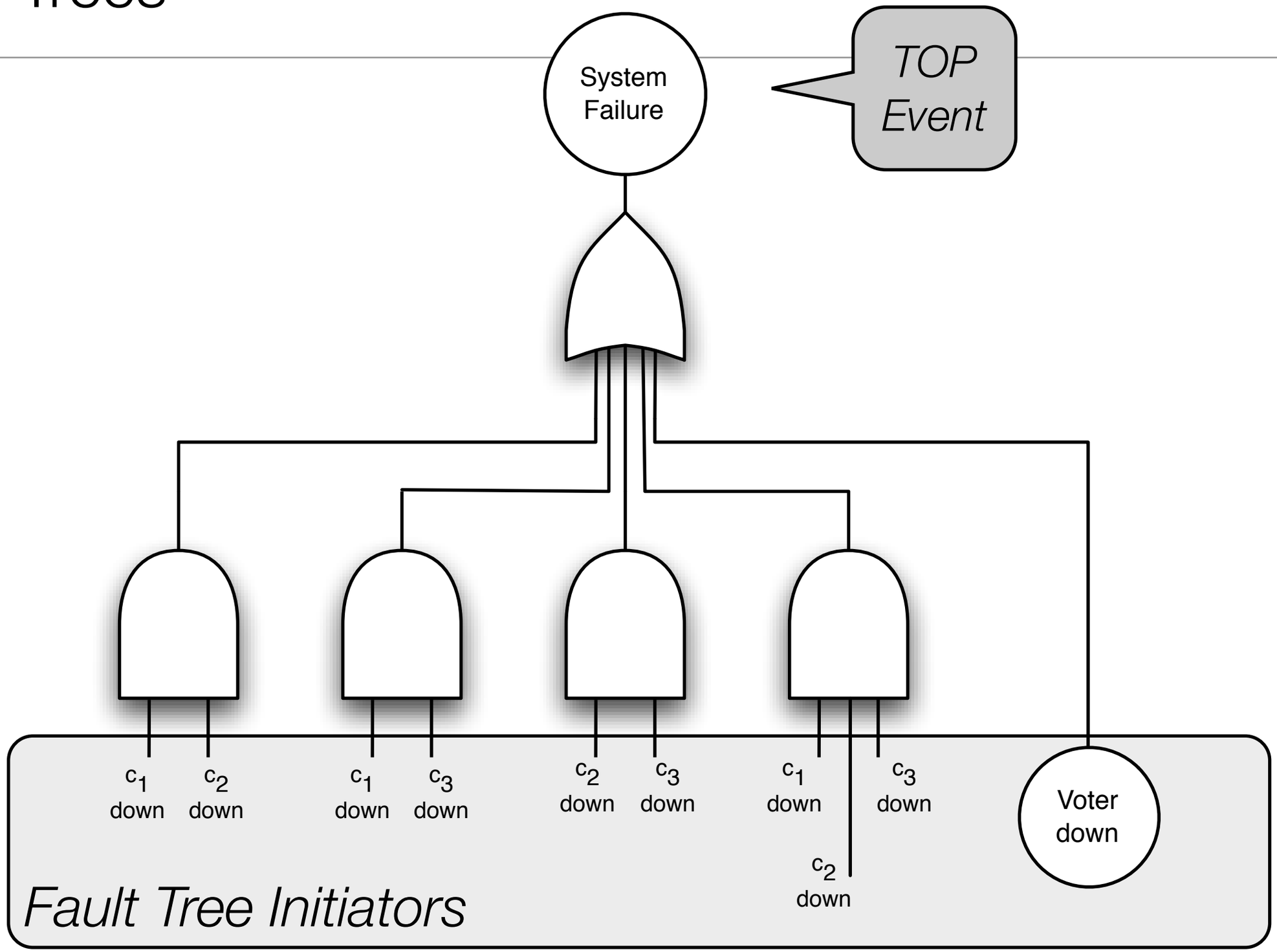
- Structure analysis effort grows exponentially with the number of components
- Fault Trees: Application of deductive logic to produce a failure-oriented analysis
  - Invented 1961 by H. Watson (Bell Telephone Laboratories) to facilitate analysis of the launch control system of the intercontinental Minuteman missile
  - Used by Boeing since 1966, meanwhile adopted by aerospace and nuclear power industry
  - Graphical representation of structure formula, helps to identify fault classes
  - Complex system failures are broken down into simpler subsystem, component, block and single element failures
  - Probability of a higher-level event can be calculated by lower level probabilities
  - Root cause analysis, risk assessment, safety assessment
- Tools: SAVE, SHARPE, Fault Tree+, AvSim+, ReliabilityWorkbench, BlockSim6, Figaro/KB3, BQR Care

Static Fault Trees

- Event types, all characterized by failure rate resp. probability
  - Basic event - Initiating fault or failure event
  - Undeveloped basic event - No information available, or insignificant consequences
  - Replicated basic event - k statistically identical copies of a component
- Gates act as intermediate events
  - AND gate - Output event occurs if all input events occur
  - OR gate - Output event occurs if one or more input events occur
  - m/n gate - Output event occurs if m or more of the n inputs occur
- Events and gates are not system components, but symbols representing the analysis



Fault Trees

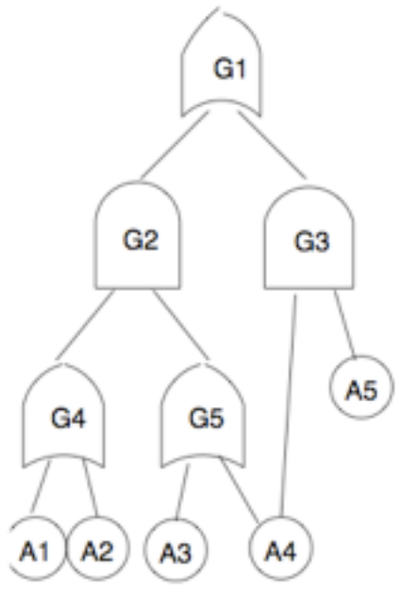

System Failure
TOP Event
c1 down
c2 down
c1 down
c3 down
c2 down
c3 down
c1 down
c2 down
c3 down
Voter down
Fault Tree Initiators

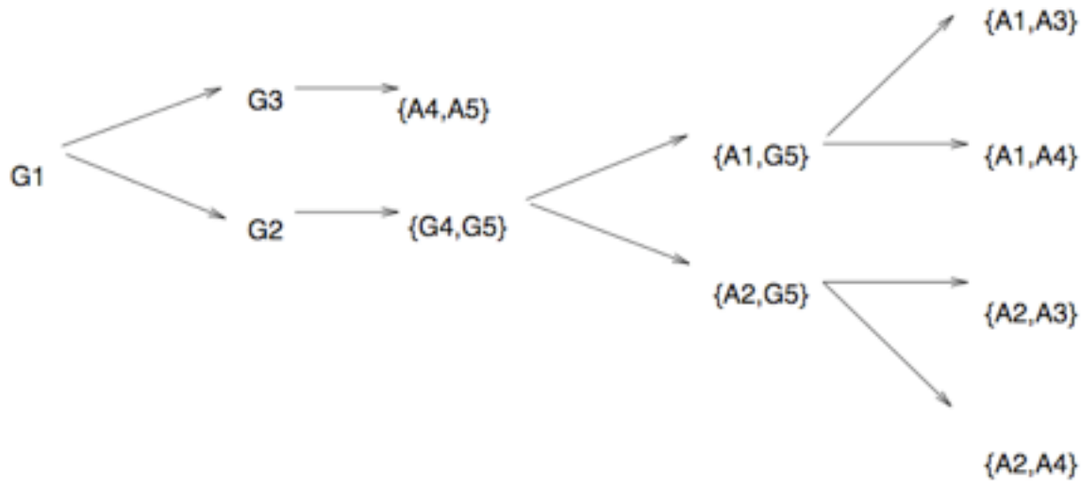
Fault Tree Construction [Misra]

- Step 1: Define the undesired event to be analyzed - what, where, when
- Step 2: Define boundary conditions for the analysis
  - Physical boundaries - What constitutes the system ?
  - Environmental stress boundaries - What is included (earthquake, bombs, ...) ?
  - Level of resolution - How detailed should be the analysis for potential reasons ?
- Step 3: Identify and evaluate fault events
  - Primary failures as basic event, secondary failures as intermediate event
- Step 4: Complete the gates
  - All inputs should be completely defined before further analysis of them
- Complete fault tree level by level

Cut Sets

- **Cut set:** Any group of initiators which, if all occur, cause the TOP event
- **Minimal cut set (mincut)**: Least group of cut set initiators - minimal combination of basic events that induce the top event
  - A long mincut signals low vulnerability, a small mincut signals high vulnerability
  - Presence of numerous cut sets signals high vulnerability
  - A singleton cut set signals a single point of failure
- Complement are **path sets**





(C) Joanne Bechta Dugan

FTA Cutsets

- Determine probabilities for cut sets to find **critical path**
  - Critical and weak links in a system design
- Analyze cutset for
  - Unexpected root cause combinations
  - Weak points in the design
  - Bypass of intended safety features
  - Common cause problems
- Methods for cutset finding
  - Boolean reduction, bottom-up reduction, top-down reduction, mapping to binary decision diagram, shannon decomposition, genetic algorithms, **...**

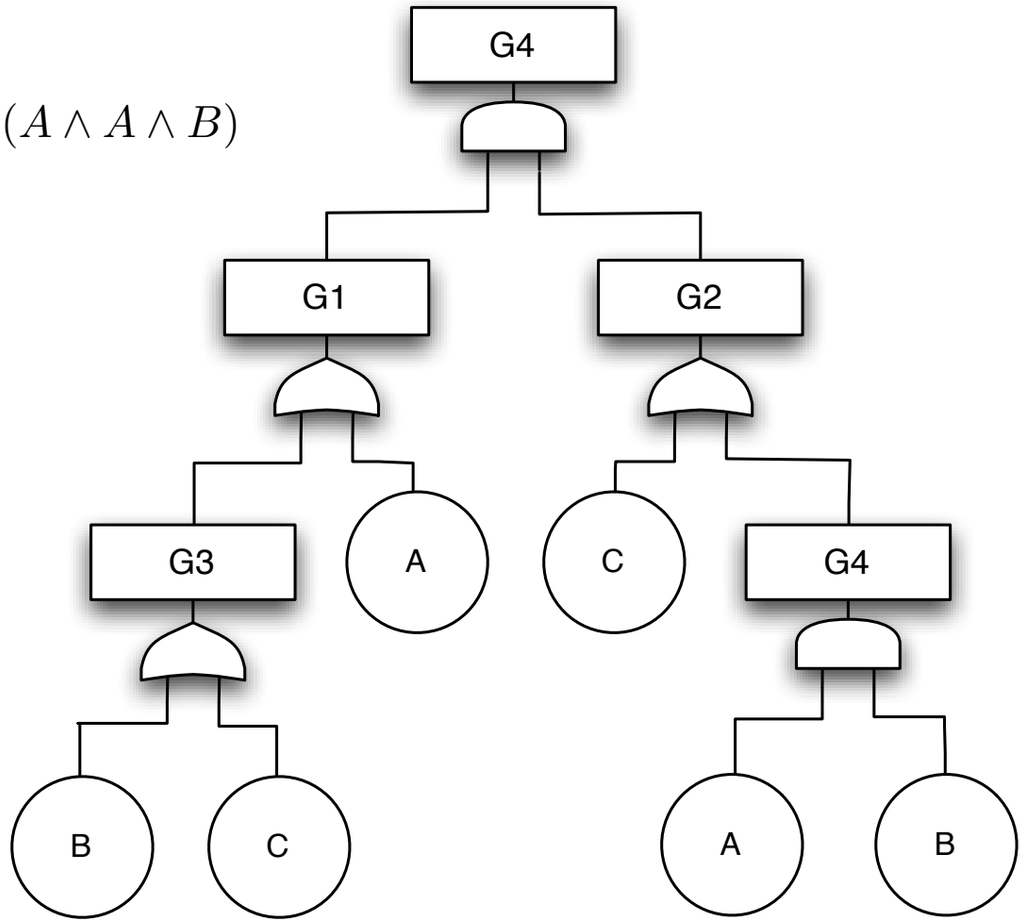
Boolean Reduction Example

$$(A \vee B) \wedge (C \vee D) = (A \wedge C) \vee (A \wedge D) \vee (B \wedge C) \vee (B \wedge D)$$

$$A \vee A = A \qquad A \wedge A = A \qquad A \vee (A \wedge B) = A$$

$$\begin{aligned} TOP &= (B \vee C \vee A) \wedge (C \vee A \wedge B) \\ &= (B \wedge C) \vee (B \wedge A \wedge B) \vee (C \wedge C) \vee (C \wedge A \wedge B) \vee (A \wedge C) \vee (A \wedge A \wedge B) \\ &= (B \wedge C) \vee (A \wedge B) \vee C \vee (C \wedge A \wedge B) \vee (A \wedge C) \vee (A \wedge B) \\ &= (B \wedge C) \vee (A \wedge B) \vee C \vee (C \wedge A \wedge B) \vee (A \wedge C) \\ &= A \wedge B \vee C \end{aligned}$$

-> 2 resulting minimal cut sets



Example by Dr. John Andrews / Loughborough University

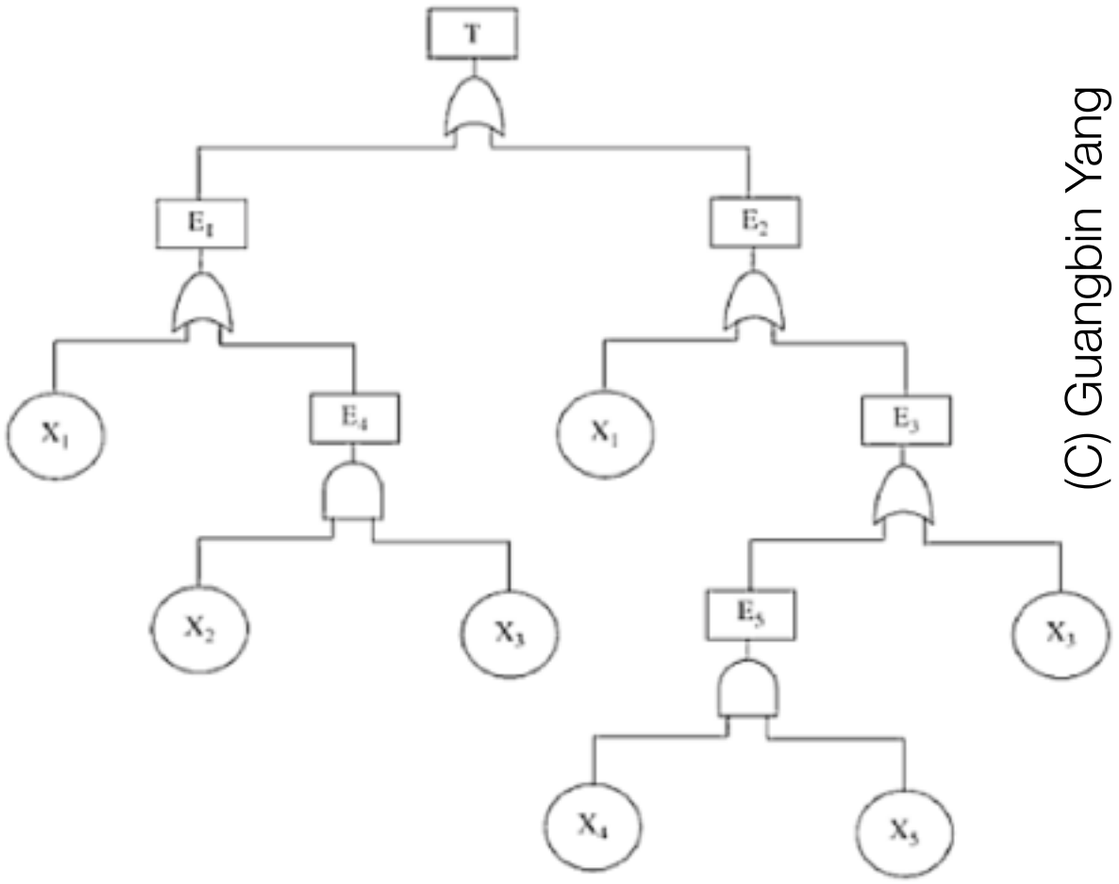
Quantitative Analysis of Fault Trees

$$TOP = X_1 \vee X_3 \vee X_4 \wedge X_5$$

$$P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$$

$$Pr(TOP) = Pr(X_1) + Pr(X_3) + Pr(X_4 * X_5) - Pr(X_1 * X_3) - Pr(X_1 * X_4 * X_5) - Pr(X_3 * X_4 * X_5) + Pr(X_1 * X_3 * X_4 * X_5)$$

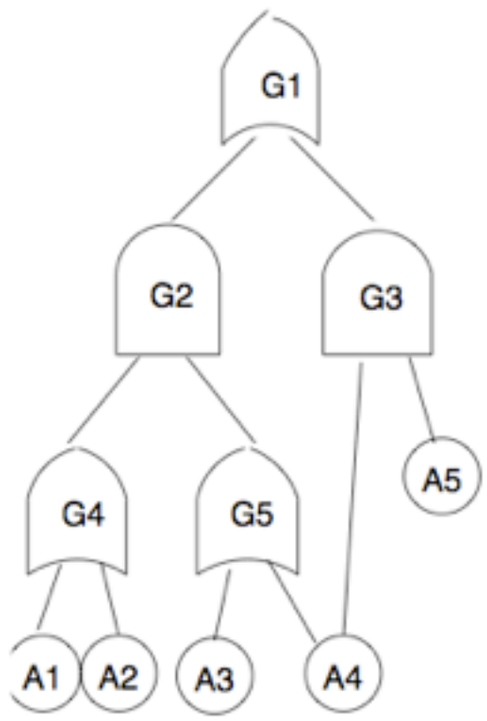
- Determine probability of TOP event by
  - Assuming independence of basic events
  - Utilize probability of independent basic events to compute probability of TOP event



(C) Guangbin Yang

Fixing Cut Sets

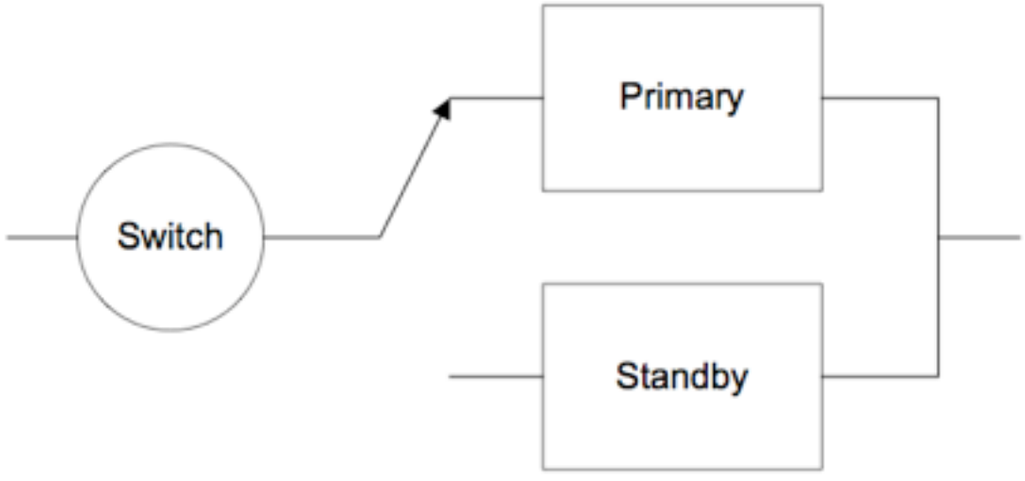
- AND gates can be protected by disallowing one of the inputs
  - Exhaustive testing or formal proof to show that the component cannot fail
  - Test for failure condition and recovery routine
- OR gates can be protected by disallowing all inputs or by providing error recovery
- Example
  - Protect G3 by preventing failure of A4
  - Protect G2 by
    - preventing failure of A3
    - preventing failure of both A1 and A2
    - providing fault tolerance for G4



Dynamic Fault Trees (DFT)

- Failure criteria of a system might depend not only on logical combination of basic events in the same time frame
  -> sequence-dependent failure

- Dynamic fault tree gates support sequences and dynamic probability changes

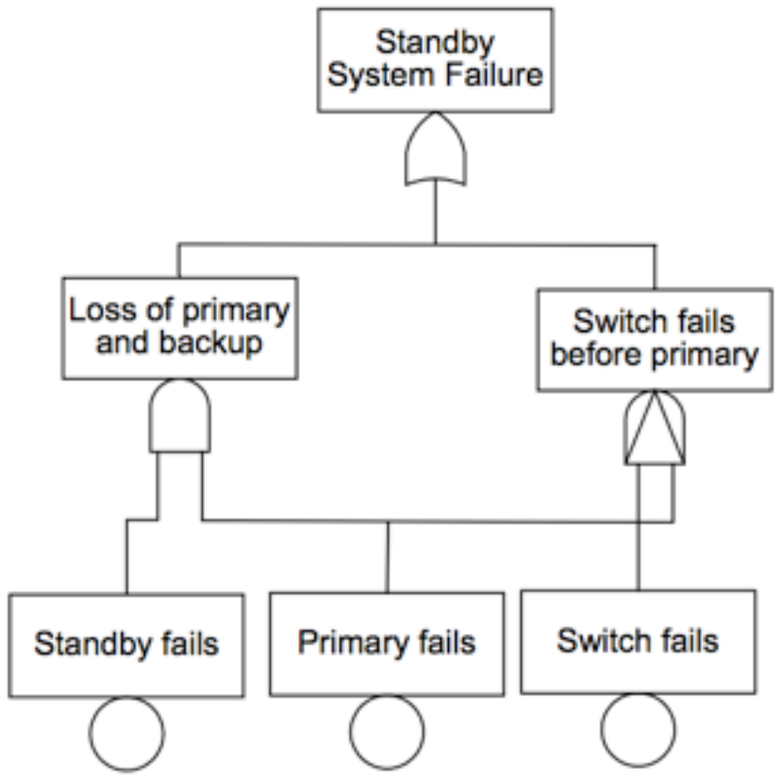


- Dynamic sub parts of the fault tree are typically analyzed by Markov model

- Example

  - Failure of switch only relevant if it happens before outage of primary

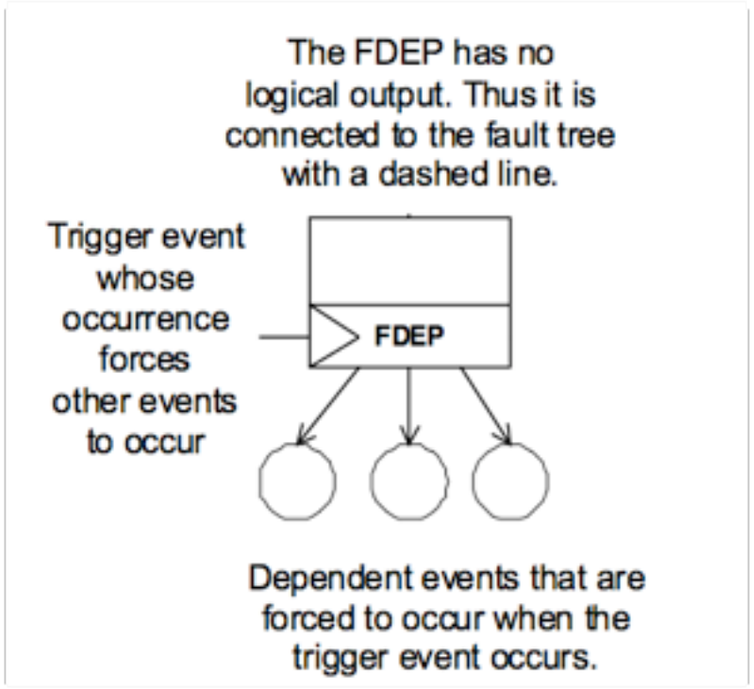
  - What is the probability of „switch fails before primary“ ?

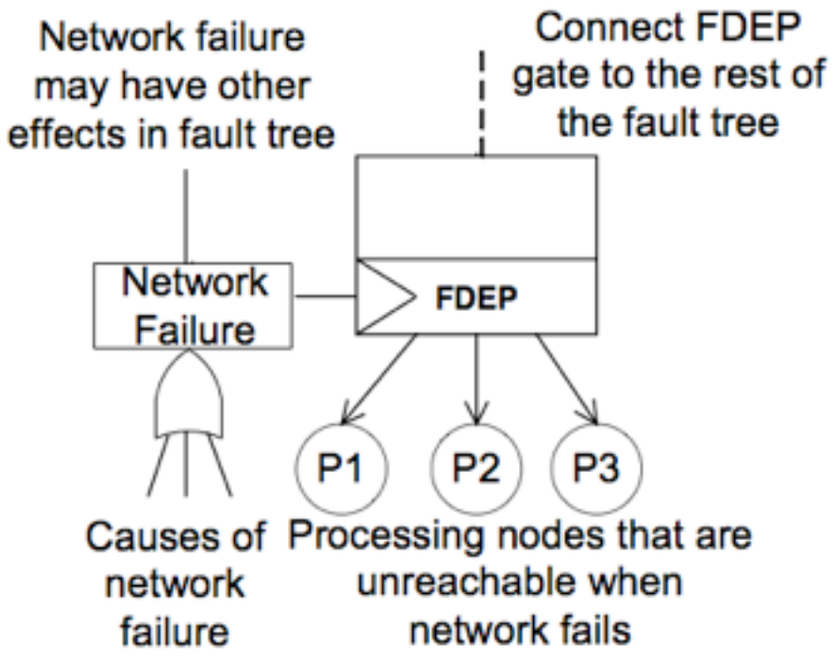


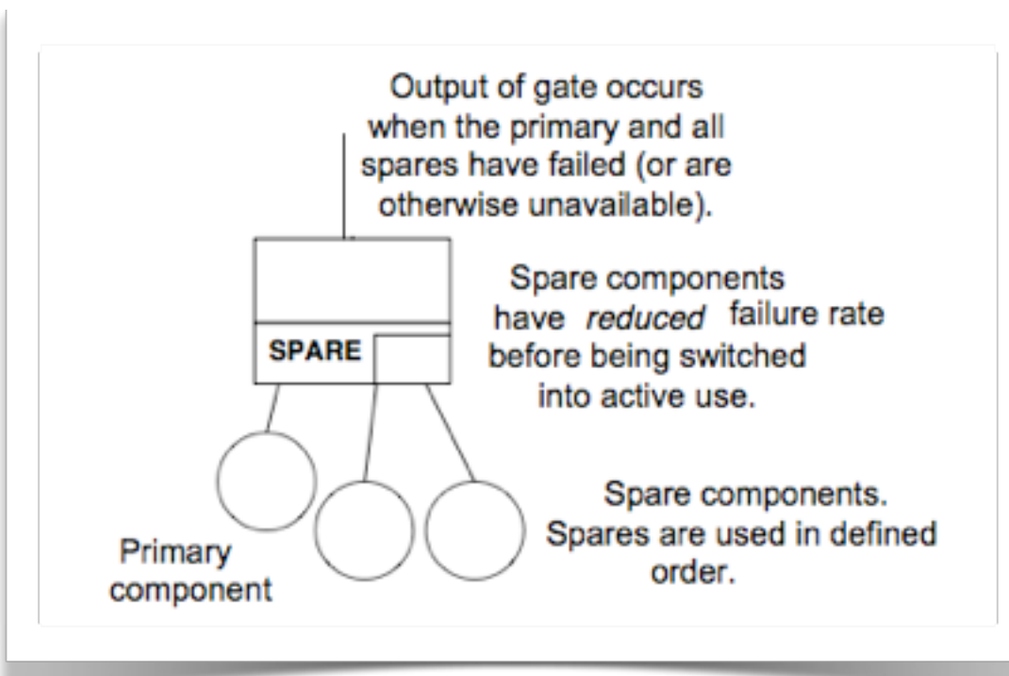
Dynamic Fault Trees

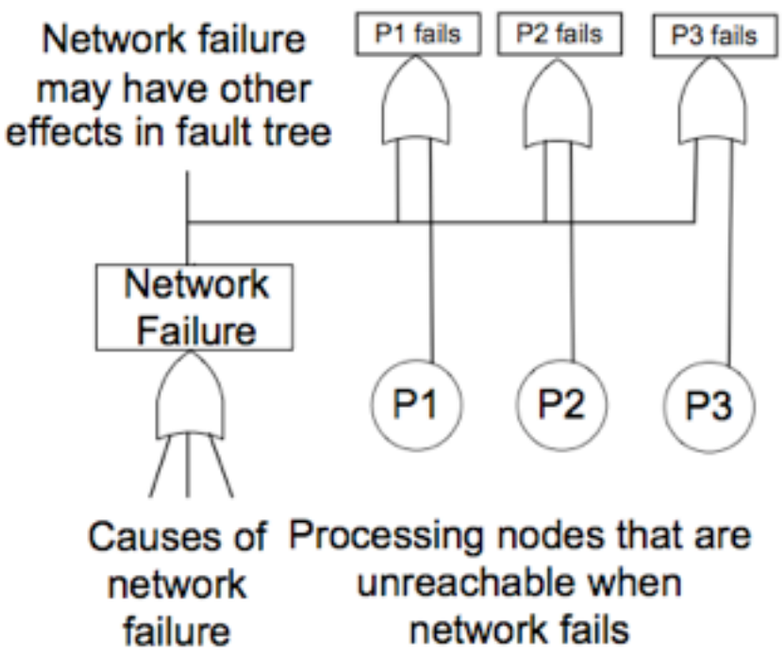
- **Functional dependency (FDEP) gate**
  - Single trigger input event, *forces* dependent events to occur on activation
  - No logical gate output - connected through a dashed line
  - Separate occurrence of the dependent events has no effect on trigger event
- **Cold Spare (CSP) Gate**
  - One primary basic input event, one or more cold spare input events
  - Alternate inputs are initially unpowered, serve as replacement for primary
  - Output occurs if all the input events occur -> primary and all spares fail
  - Support modeling of *cold spares* (zero failure rate when unpowered), *warm spares* (reduced failure rate when unpowered) or *hot spares*
    - *Dormancy factor* defines decrease of failure probability without primary event

Dynamic Fault Trees - Examples [Veseley]

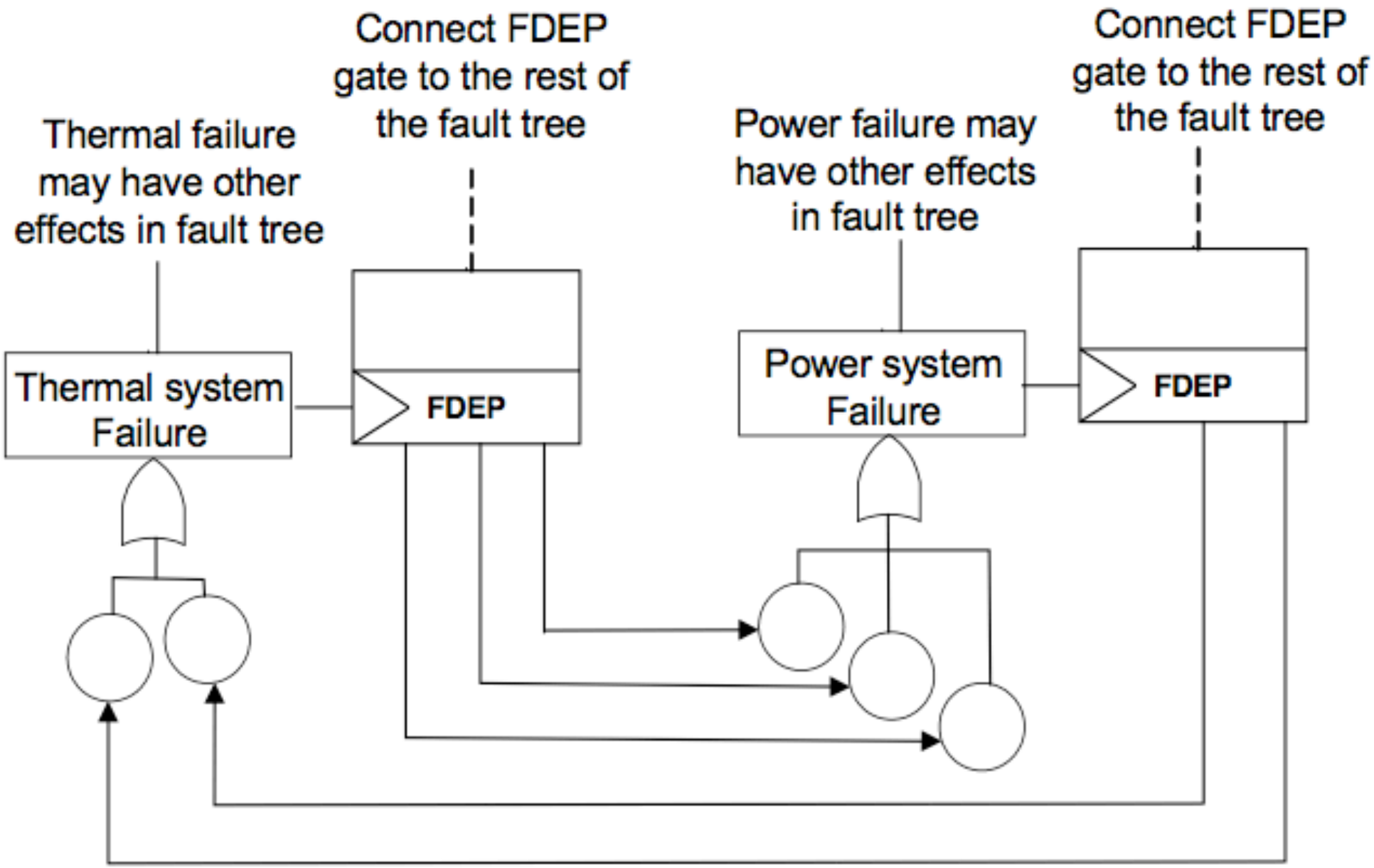








Dynamic Fault Trees - Feedback Modeling

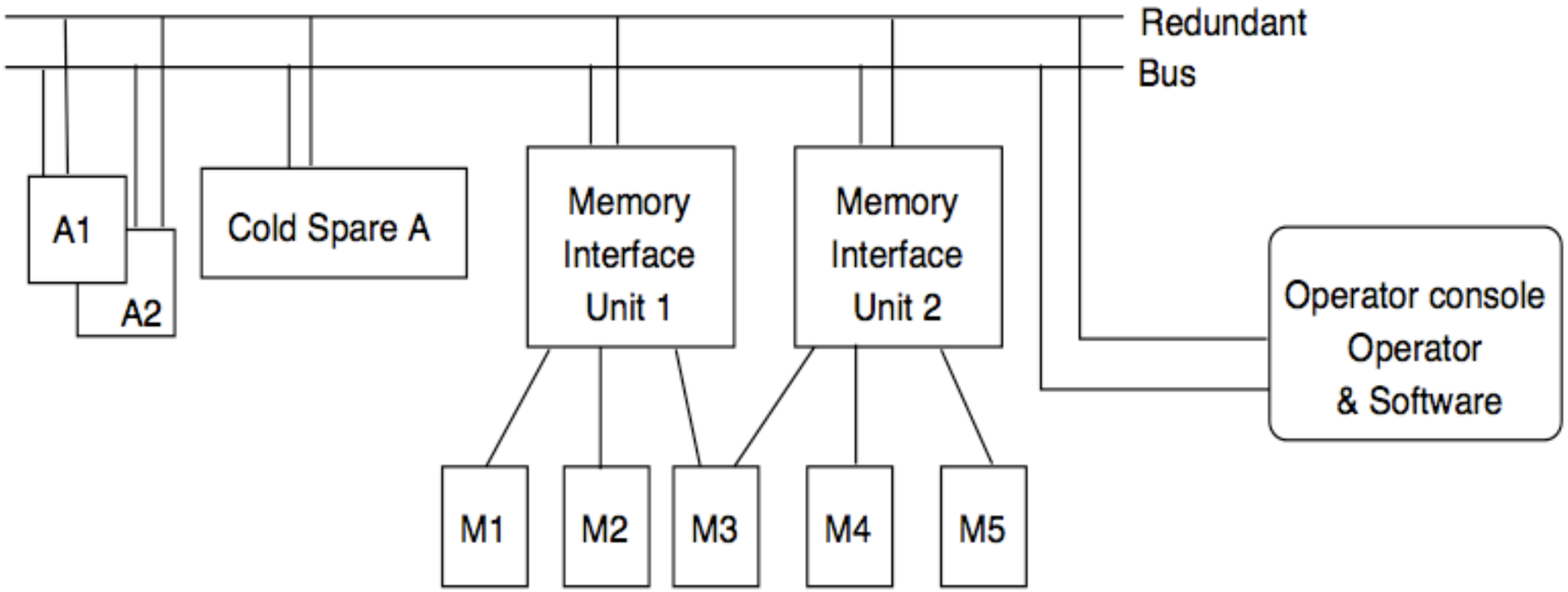


Dynamic Fault Trees

- **Priority-AND (PAND) Gate**
  - Normal AND gate, plus extra condition that input events must occur in specified order to trigger the output
- **Sequence Enforcing (SEQ / SENF) Gate**
  - Model mandates that the input events occur in given order

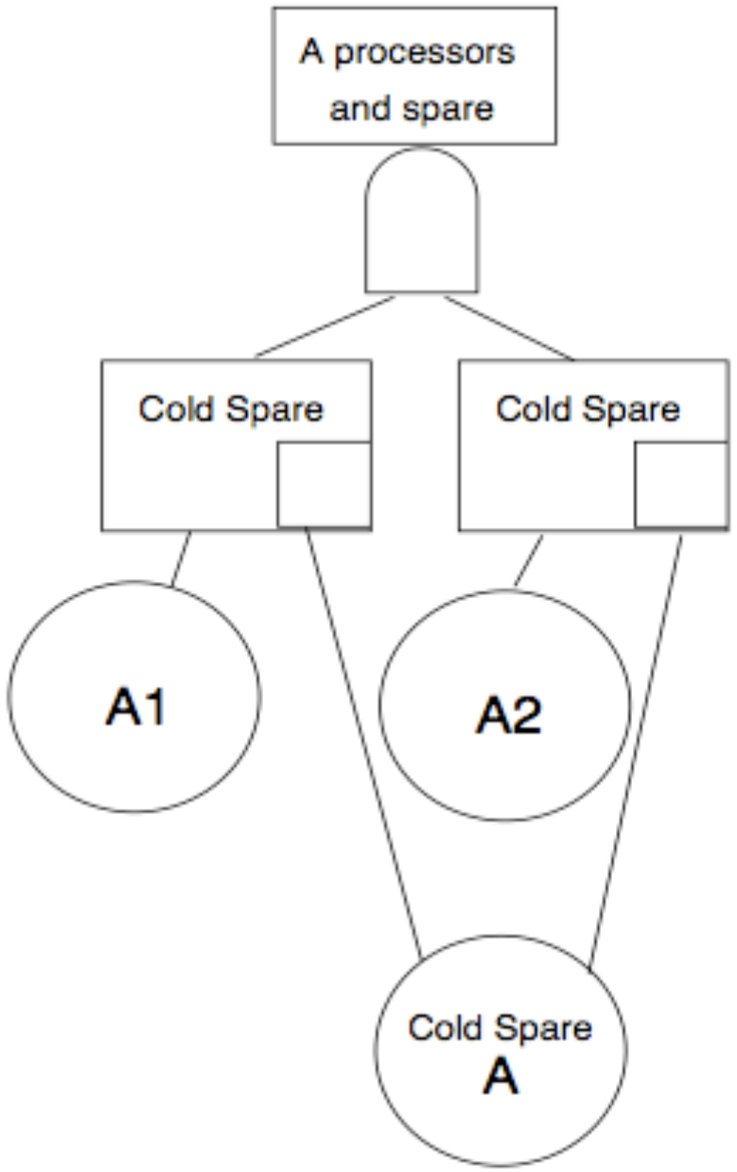
- Modeling of dynamic fault trees with Markov chains considers ordering condition for probabilistic events
  - Example: With cold spare gate, probability of spare failure changes, depending on the probability of earlier primary outage

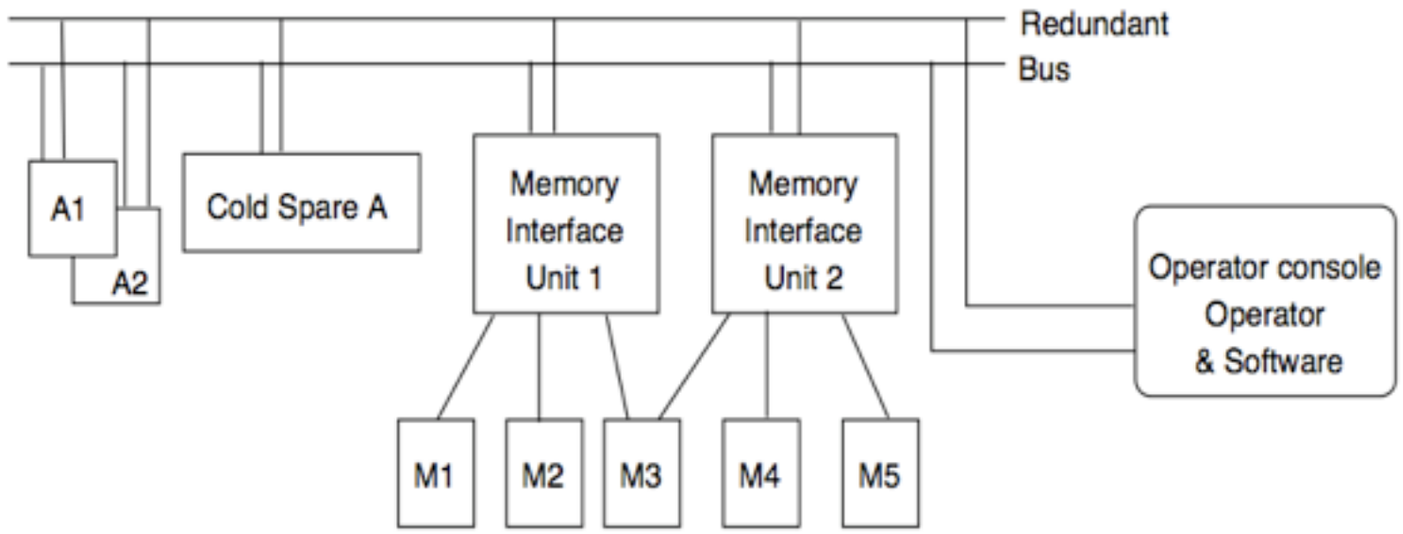
Hypothetic Example Computer System (HECS)



- Minimum demands for operation
  - One functional processor, three memory modules, one bus, operator + console
- Example taken from Joanne Bechta Dugan, University of Virginia (www.fault-tree.net)

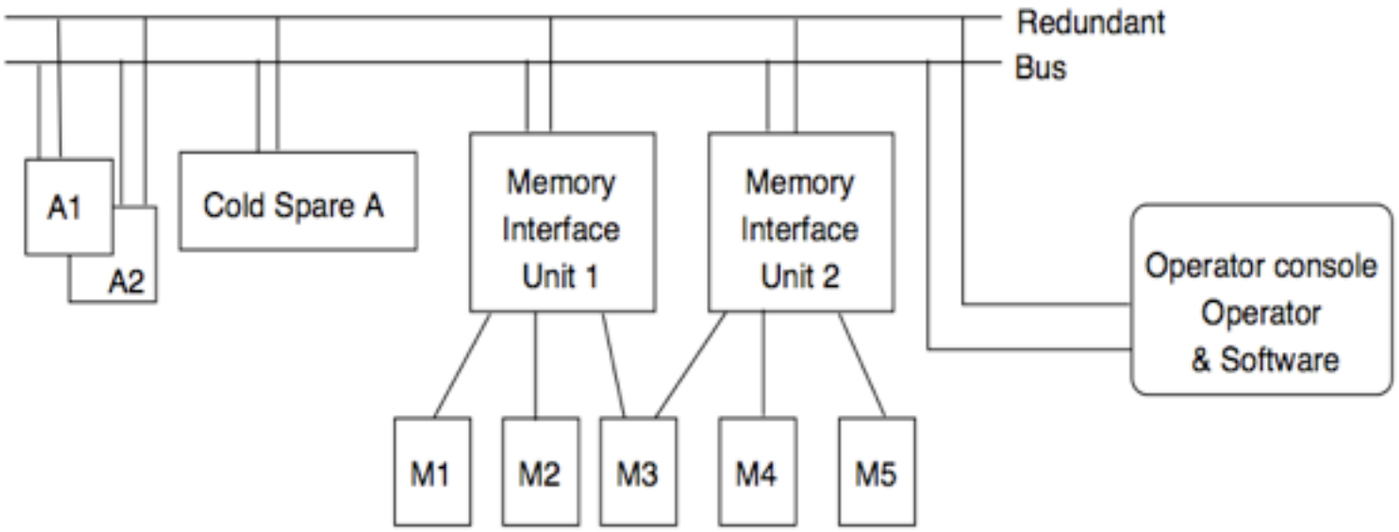
HECS Example

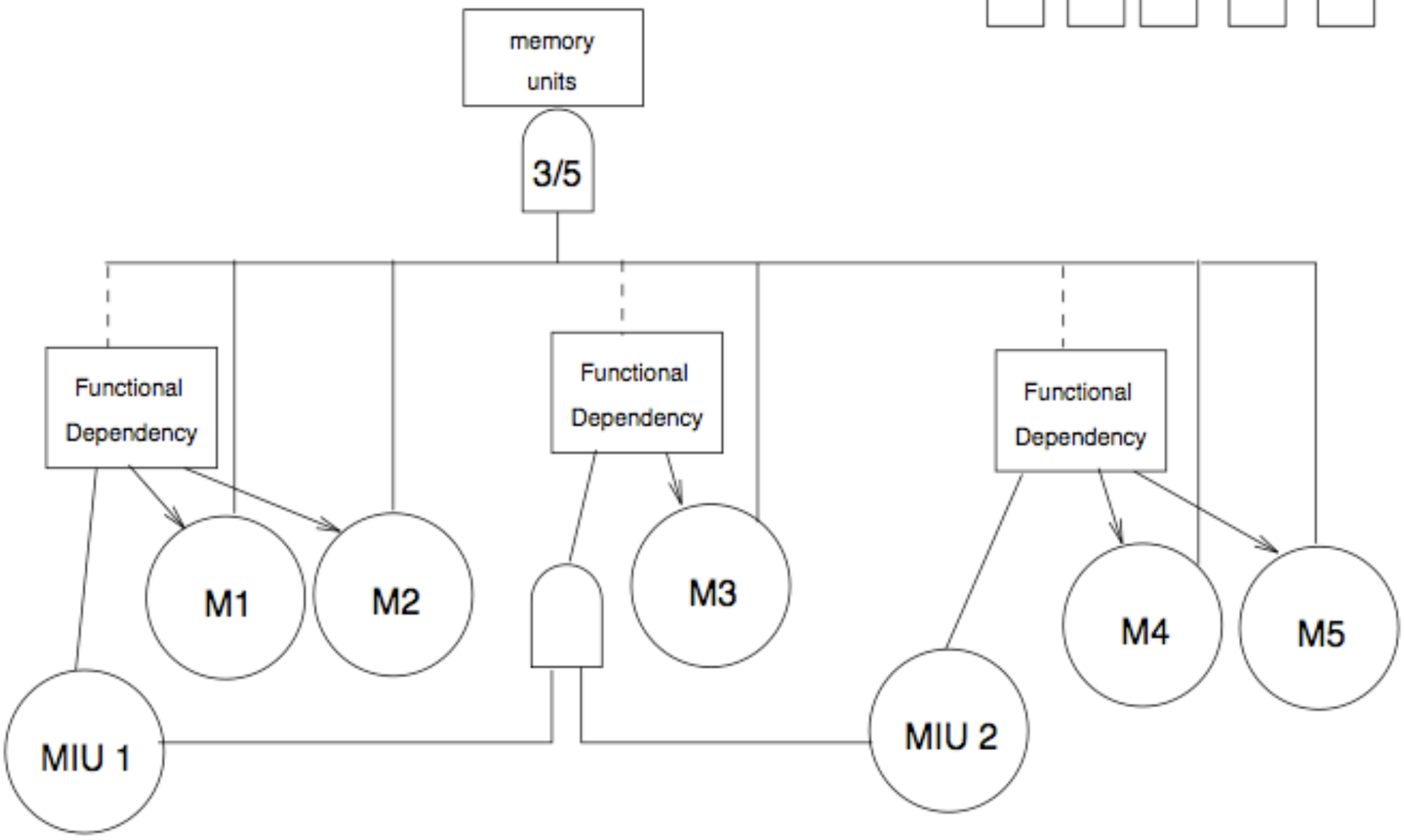




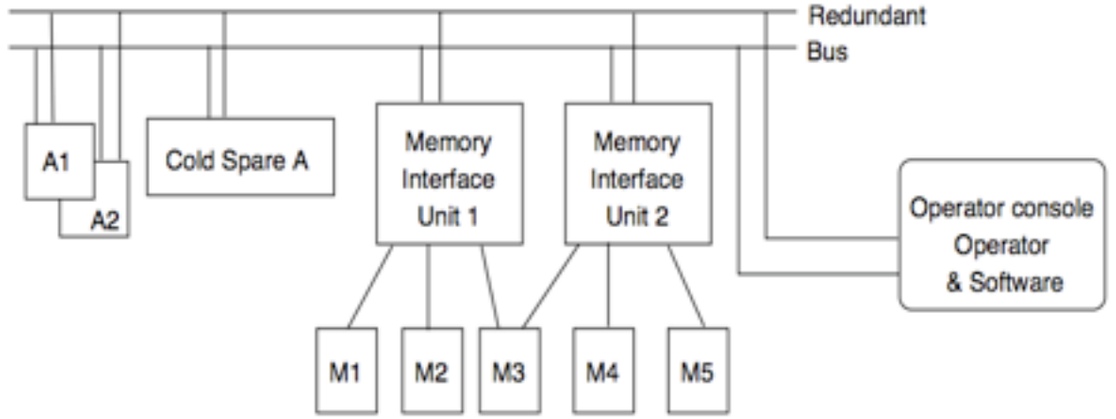
- Failure rate of active processor is different from cold spare failure rate when not activated
  - Cold spare - dormancy factor of 0

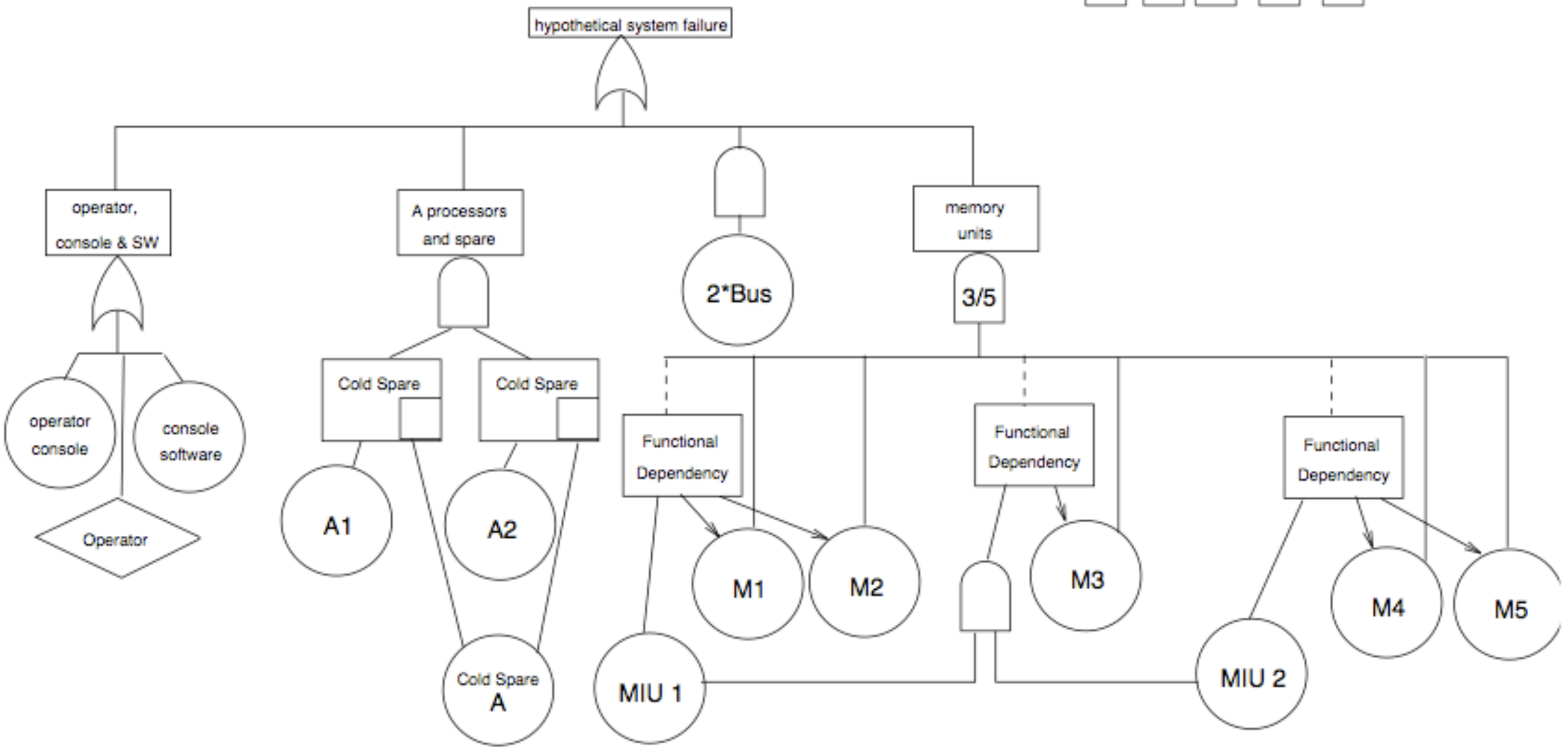
HECS Example



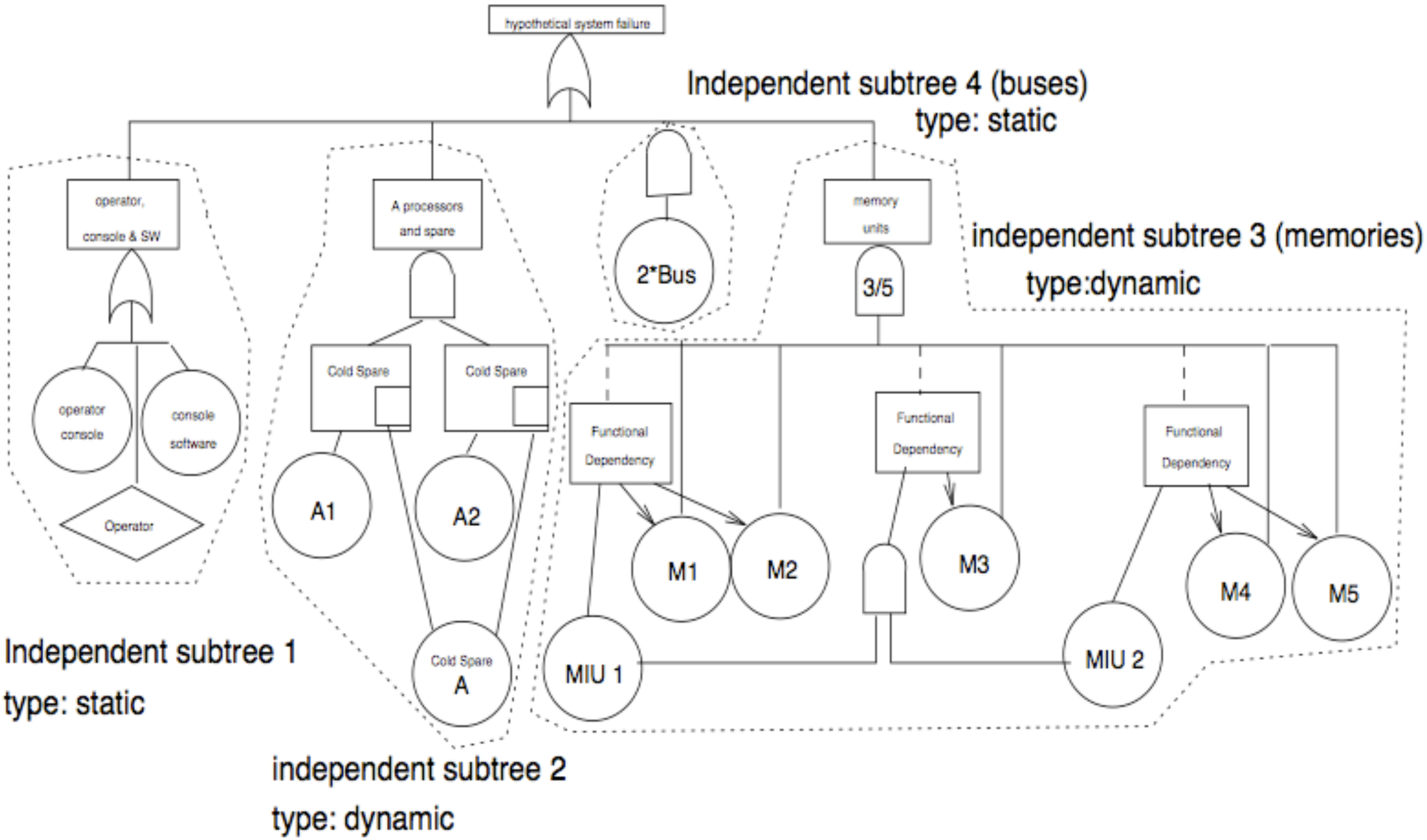


HECS Example



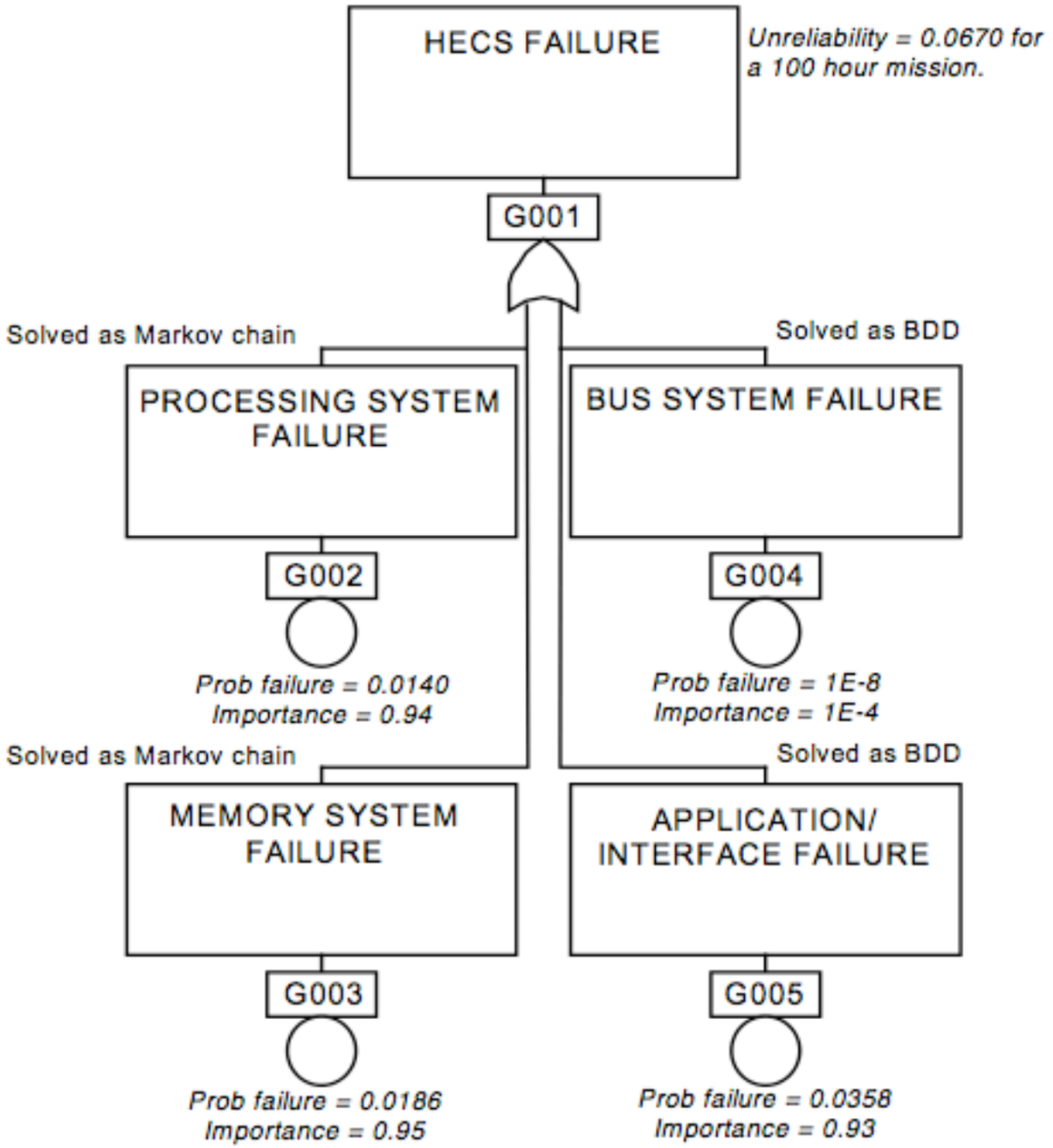


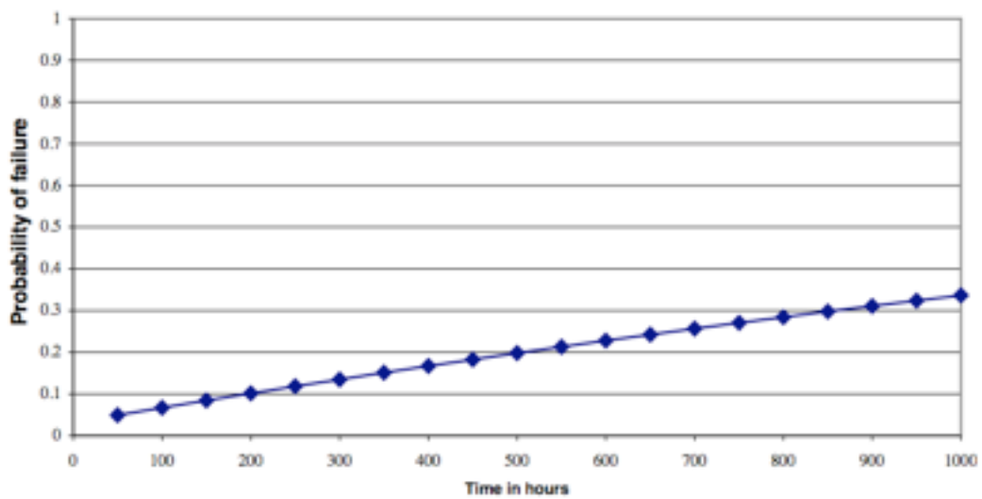
HECS Example



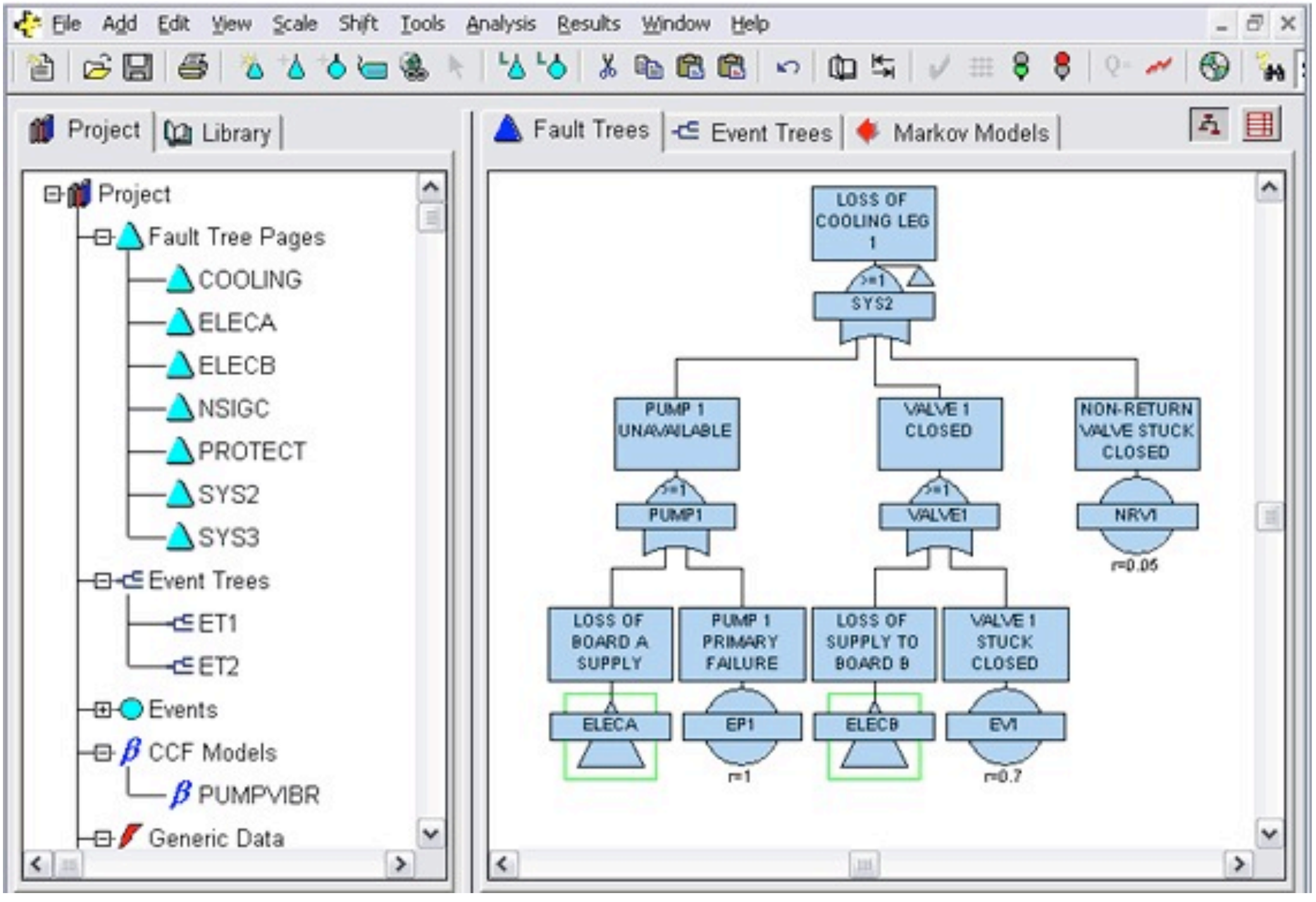
HECS Example

- Analysis with Galileo/ASSAP system for an 100-hour mission
- Processing and memory system analyzed by Markov models
- Importance analysis with Birnbaum method
- Basic assumptions for component failure rates





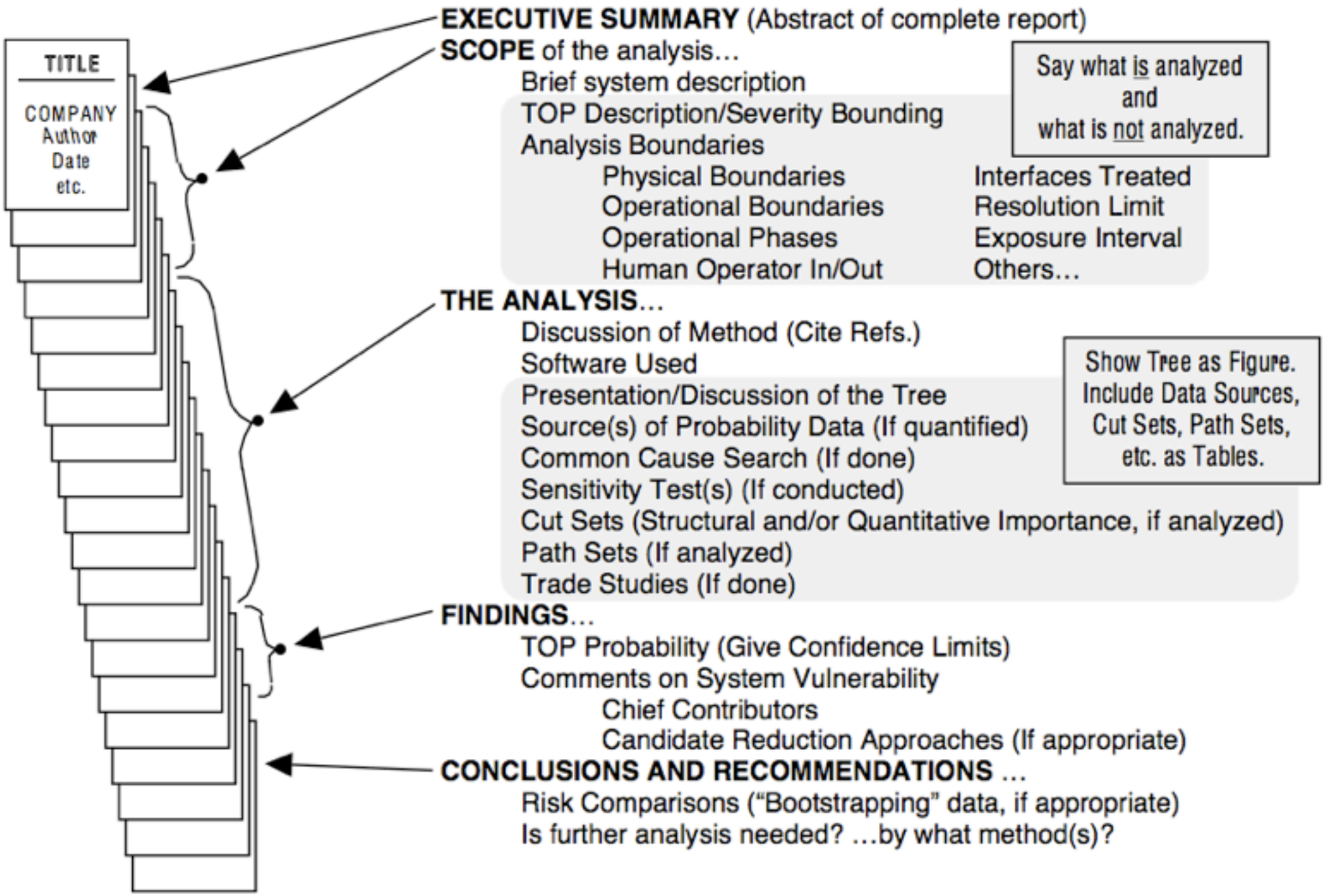
FTA Tools


File Add Edit View Scale Shift Tools Analysis Results Window Help
Project
Library
Fault Trees
Event Trees
Markov Models
Project
Fault Tree Pages
COOLING
ELECA
ELECB
NSIGC
PROTECT
SYS2
SYS3
Event Trees
ET1
ET2
Events
CCF Models
PUMPVIBR
Generic Data
LOSS OF COOLING LEG 1
>=1
SYS2
PUMP 1 UNAVAILABLE
VALVE 1 CLOSED
NON-RETURN VALVE STUCK CLOSED
>=1
PUMP1
>=1
VALVE1
NRV1
r=0.05
LOSS OF BOARD A SUPPLY
PUMP 1 PRIMARY FAILURE
LOSS OF SUPPLY TO BOARD B
VALVE 1 STUCK CLOSED
ELECA
EP1
ELECB
EV1
r=1
r=0.7

Fault Tree Construction [Misra]

- Common errors in construction
  - *Ambiguous TOP event* - Too general TOP event makes FTA unmanageable, too specific TOP event cannot provide a sufficient system analysis with FTA
  - *Ignoring significant environment conditions* - External stress might be relevant
  - *Inconsistent fault tree event names* - Same name for same fault event or condition
  - *Inappropriate level of resolution* - Detail level of the fault tree should match the detail level of the available information

FTA Report (Clemens & Sverdrup)



FTA Shortcomings (Clemens & Sverdrup)

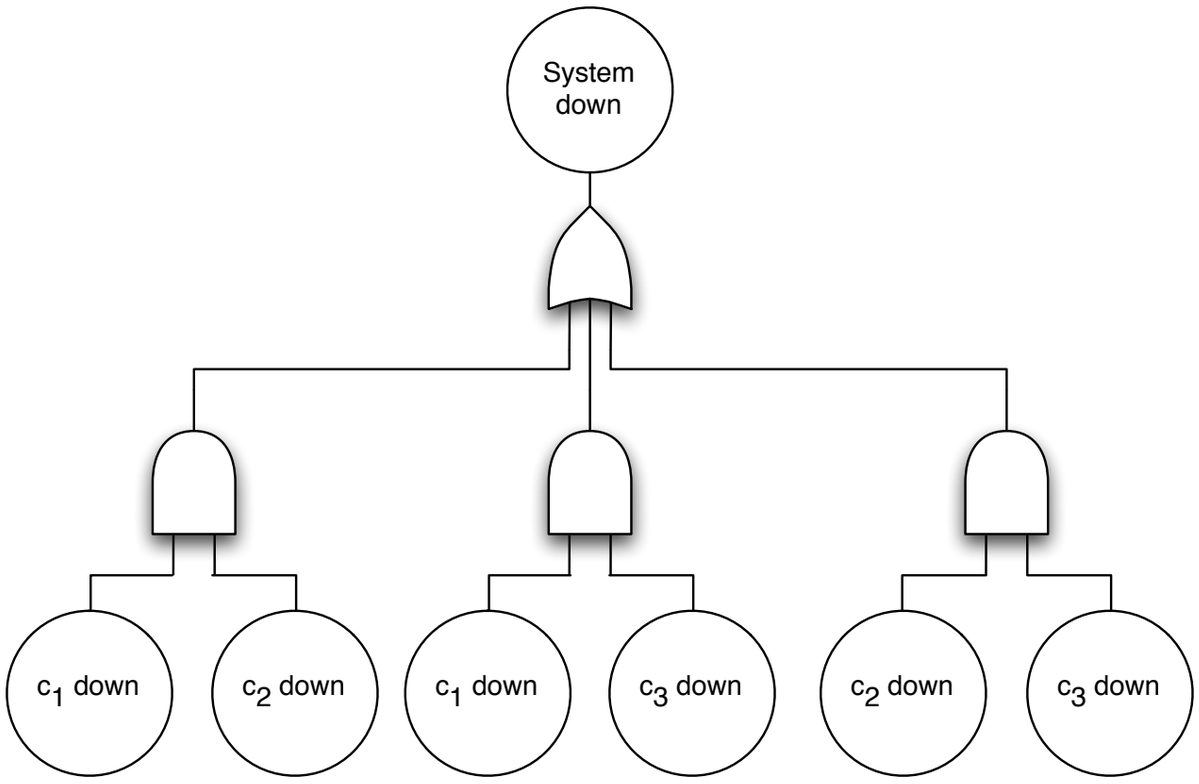
- Undesirable events must be foreseen
- All significant failure contributors must be foreseen
- Each initiator must be constrained to two conditional modes when modeled
- Initiators beneath a common gate must be independent of each other
- Events must be immediate contributors to next higher level (timing)
- Each initiators failure rate must be predictable
  - Amount of data must justify a quantitative analysis !

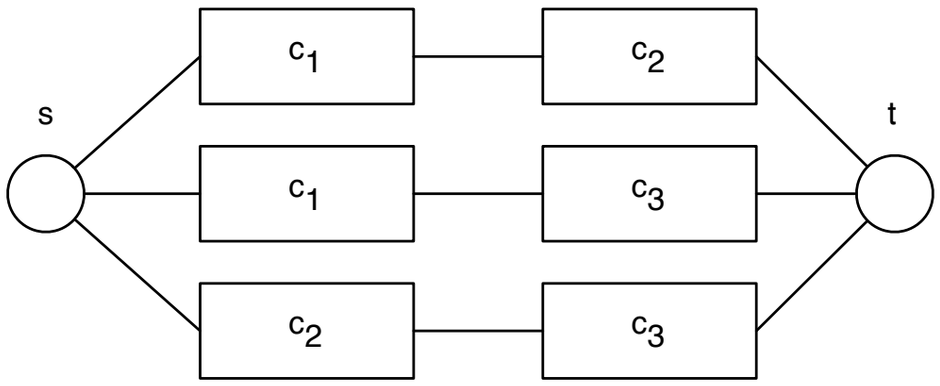
FTA Remarks

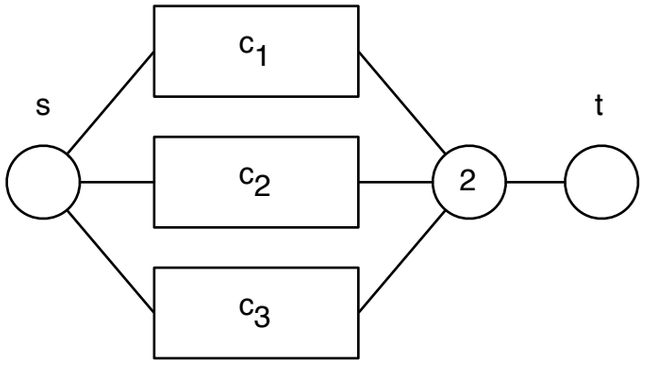
- Proper and consistent naming is very important
  - Per event: WHAT failed and HOW
- Initiators must be statistically independent of one another
- Contributing elements must be an IMMEDIATE contributor
- At a given level under a given gate, each fault must be independent from all others
- Logic can be tested in SUCCESS DOMAIN - invert all statements and gates and check correctness
- Analyze no further down than is necessary to enter probability data with confidence

Reliability Block Diagrams (RBD)

- Alternative deductive way to model logical interaction - success-oriented analysis
- System is available only if there is a path between s and t
- Structure formula can be obtained by identifying all minimal cut sets
  - Subset of edges that disconnects s from t if removed

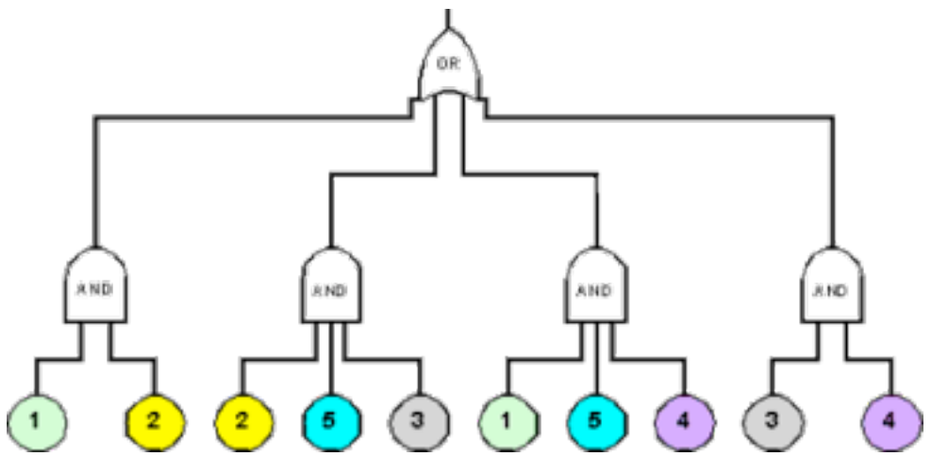


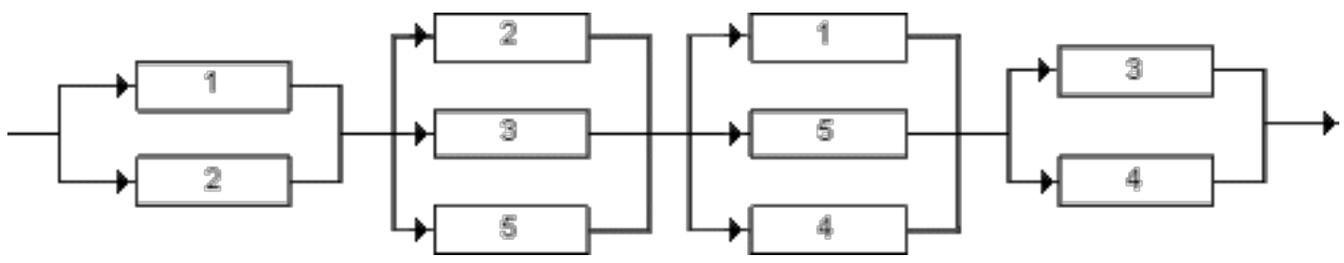




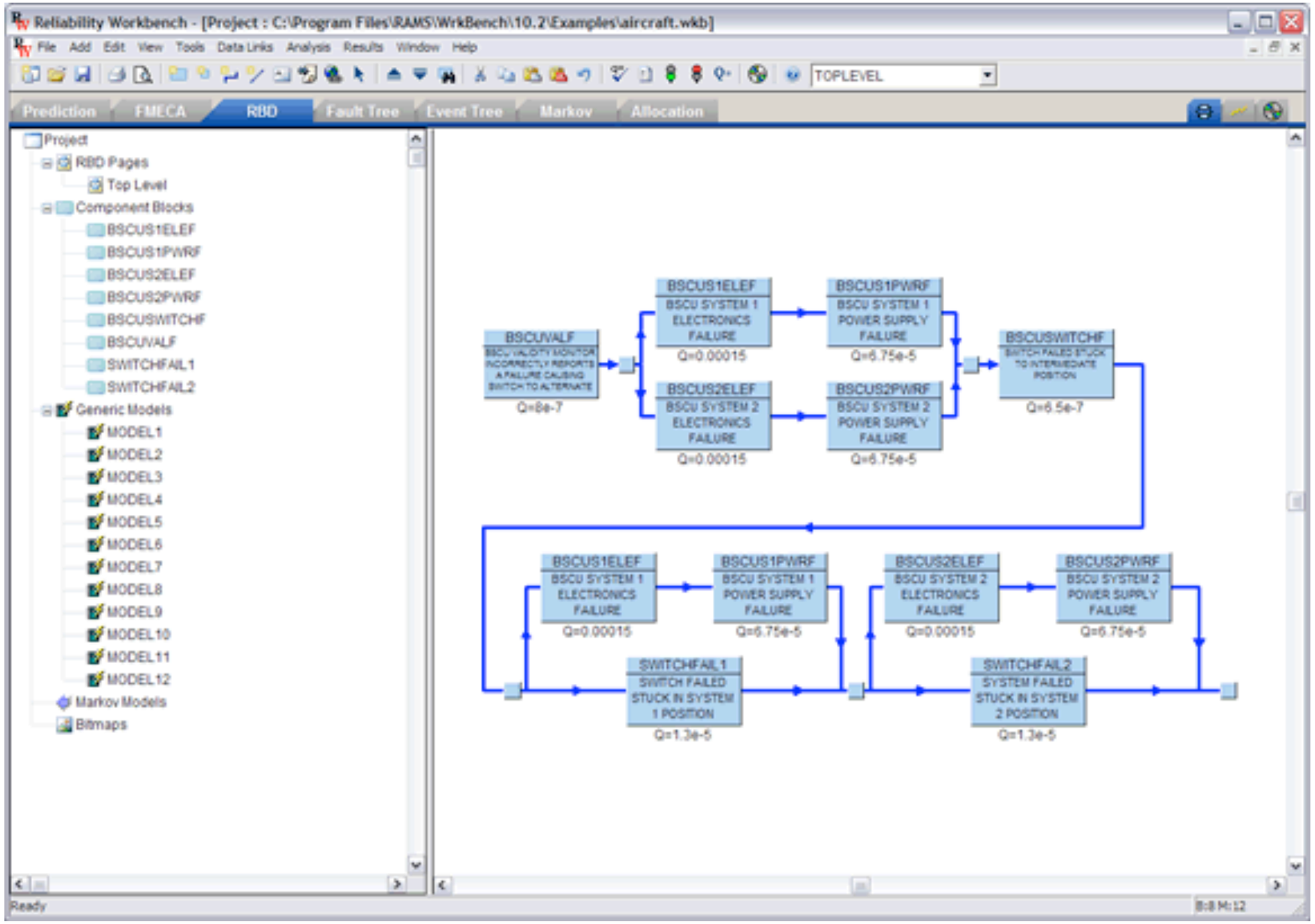
Reliability Block Diagrams (RBD)

- Convert fault tree to reliability block diagram
  - Start from TOP event, replace gates successively
  - Logical AND gate <-> parallel structure of the inputs of the gate
  - Logical OR gate <-> serial structure of the inputs of the gate
  - Elements in the fault tree: Failure events, blocks in the RBD: Functioning blocks
- Some FTA and RBD extensions are not convertible
  - Example: Sequence-dependent gates in fault trees





RBD Tools


Reliability Workbench - [Project : C:\Program Files\RAMS\WrkBench\10.2\Examples\aircraft.wkb]
File Add Edit View Tools Data Links Analysis Results Window Help
TOPLEVEL
Prediction FMECA RBD Fault Tree Event Tree Markov Allocation
Project
RBD Pages
Top Level
Component Blocks
BSCUS1ELEF
BSCUS1PWRF
BSCUS2ELEF
BSCUS2PWRF
BSCUSWITCHF
BSCUVALF
SWITCHFAIL1
SWITCHFAIL2
Generic Models
MODEL1
MODEL2
MODEL3
MODEL4
MODEL5
MODEL6
MODEL7
MODEL8
MODEL9
MODEL10
MODEL11
MODEL12
Markov Models
Bitmaps
BSCUVALF
Q=8e-7
BSCUS1ELEF
BSCU SYSTEM 1 ELECTRONICS FAILURE
Q=0.00015
BSCUS1PWRF
BSCU SYSTEM 1 POWER SUPPLY FAILURE
Q=6.75e-5
BSCUS2ELEF
BSCU SYSTEM 2 ELECTRONICS FAILURE
Q=0.00015
BSCUS2PWRF
BSCU SYSTEM 2 POWER SUPPLY FAILURE
Q=6.75e-5
BSCUSWITCHF
Q=6.5e-7
SWITCHFAIL1
SWITCH FAILED STUCK IN SYSTEM 1 POSITION
Q=1.3e-5
SWITCHFAIL2
SYSTEM FAILED STUCK IN SYSTEM 2 POSITION
Q=1.3e-5
Ready