A Probabilistic Approach to Estimating
Computer System Reliability
by Robert Apthorpe

Presented by
Richard Martin

Rutgers University
Department of Computer Science

Internet Services
CS 553

1

Author

- MS, Nuclear Engineering Wisconsin
- Quantified reactor safety
- Worked at Excite
  - Large internet service
- Perspectives on Real Engineering™ and computer system administration
  - Understands thought processes of each domain
    - Applies reasoning, paradigms from Real Engineering™ to computer systems
  - Understands computer systems cost arguments against PRA

2

Approach

- Probabilistic Risk Assessment (PRA)
  - How likely are events to occur?
    - Forces designers to reason about space of events
      - What is and is not covered
    - Quantifies importance of events, components
      - Focus attention on what ' s important
- Two Techniques
  - Event Trees
    - Space of events, likelihoods of outcomes
  - Fault Trees
    - Reason about component interactions
- Best used in combination

3

Why Use PRA?

- Often no means to objectively rank risks
  - What ' s worth paying attention ($) to, and what ' s not.
- Not making rational decisions with regards to computational risks
  - Muddles along, locally optimal decisions
  - Buys expensive hardware/software which does little to impove availability and reliability
  - Critical risks overlooked
- PRA proven in other fields
  - defnese, aerospace, nulcear
  - Can it be applied to computer systems with reasonable cost?

4

PRA Utility

- Risk Communication Tools
  - Parties understand what is of concern or not
- Event Trees
  - Provide global view
  - Break up analysis into smaller parts
  - Show consequences
- Fault Trees
  - Low/level, detailed
  - Sequences of events required for failure
  - Rank components by contribution to risk
  - Undeveloped faults show limitations of analysis

5

PRA Limitations

- Cost of analysis is excessive
  - Lapace transforms, Markovian analysis beyond the scope of entry-level education
    - Event trees, FT' s easier to analyze
  - Models can get large and unwieldy
    - Bu event trees, fault trees scale down too
- Not time dependent (but is this a plus?)
- Faith in absolute numbers suspect
  - But provides sensitivity and relative rankings

6

Where to start

- Identify hazards
  - A lot of hazards!
- Identify relevant systems, components, and people
  - Large number of components
  - Not so many systems though
- Bound the analysis
  - PRA
    - Events tree
    - Fault trees

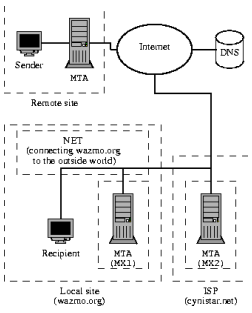
7

Example: Mail Transport

DNS MX (mail) record:

```
IN MX 10 shaft.mx.mazmo.org
IN MX 20 dolomite.mx.wazmo.org
IN MX 30 mta.cynistar.net
```

Remote MTA tries servers in this order



Identify components

- Remote sender (person)
- Remote user host
- Remote MUA (user mail program e.g. pine)
- Remote MTA (transport program e.g. sendmail)
- Remote network
- DNS
- Inter-Networks (many)
- Local ISP
- Local network
- Local MTA
- Local host
- Local MUA
- Local recipient (person)

9

Larger Scope

- Power
  - at all sites along the path?
- On site environment
  - Temperature
- Off-Site environment
  - Fire, floods, snow
- Human activity
  - Unplug cables, power, reboot wrong box
- Political climate
  - Riots, war, lawsuits, strikes

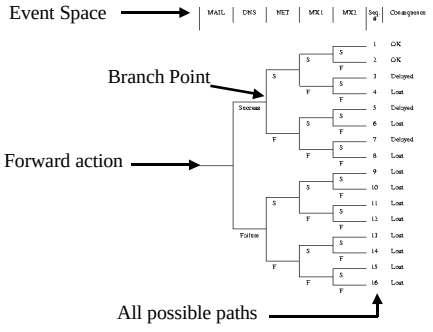
10

Event and Fault Trees

- Event tree specifies all failures/events of interest
  - Good for a high-level view
- Need to know probabilities of each event
- Use fault tree analysis to obtain probabilities for splits in the event tree
  - FTs are a more detailed analysis technique

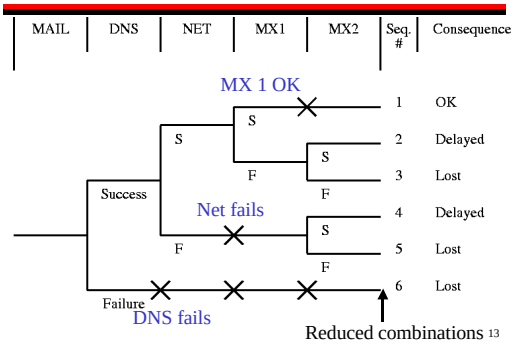
11

Event Tree Example



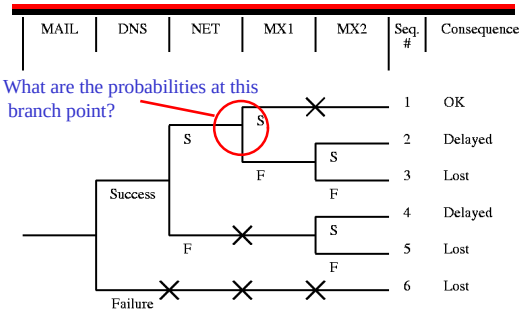
12

Reduced Event Tree



Reduced combinations

Using Fault Tree Analysis



Fault Tree Analysis Overview

- Select Top level Event
  - Root of the tree
- Decompose into more specific events
  - Decomposition rules
- Stop when appropriate level of detail is reached
  - More art than science
- Convert logic trees into equations based on leafs
- Find minimal cutsets
- Apply probability theory and failure rate data to find liklihood of each cutset
- Use metrics to find component importance

Fault Categories

- Faults vs. Failures
- State-of-system faults
  - Cause is outside of component in question
  - Futher modeling necessary
- State-of-component faults
  - Primary (within operating range)
  - Secondary (outside range)
  - Command (timing or action faults)

FT event categories

- Top
  - Failures/events of interest
  - Short,unambiguous
- Intermediate
- Primary (leaf)
  - Basic
    - primary/secondary/command
  - Undeveloped
    - too complicated/uncertain
  - Conditioning
    - restrictions or necessary conditions
  - External
    - "House", normal expected event

17

Decomposition Rules

- Write statements in event nodes as faults
  - State what fault is, and when it occures
- If a state-of-component fault:
  - classify as add "or" gate and look for primary, secondary and command modes to add
- If a state-of-system fault:
  - look for **minumum, necessary**, and **immediate** conditions
  - add conditionals as necessary
- Faults can propagate through working components
- Breath-first traversal
- Any sub-tree is a well formed tree (no gate-to-gate connections)

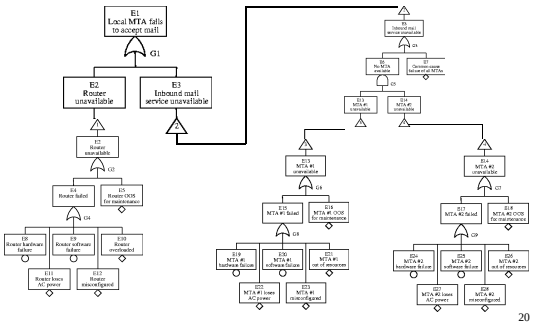
18

Power Failure Example

- Immediate, necessary and sufficient
- A machine with two UPSes
- One event is loss of power to computer
  - Second event loss of power in UPS 1
  - Third event is loss of power in UPS 2
- Loss of Grid power
  - Conditional is > 20 minutes
  - Second conditional is > 40 minutes
- What would this tree look like?

19

Fault Tree for Event MX1



20

Cutsets

- Set of events which cuts the root from the rest of the tree
- Minimal cutset:
  - Set of events for which any one events cuts the root
- Goal of FTA:
  - Find the set of minimal cutsets

21

Risk Metrics

- Birnbaum
  - sensitivity of component failure to entire range of risk
- Fussell-Vessely
  - sum of all cutsets containing that component to entire risk
- Risk Achievement Worth (RAW)
  - Fractional increase in risk assuming event always occurs
- Risk Reduction Worth (RRW)
  - Factional decrease in risk assuming event never occurs

22

Metric Equations

$$RAW_i = \frac{Pr(T, Pr(e_i) = 1)}{Pr(T)}$$

$$RRW_i = \frac{Pr(T)}{Pr(T, Pr(e_i) = 0)}$$

$$I_B^j = \left(RAW_j - \frac{1}{RRW_j} \right) Pr(T)$$

$$I_{FV}^j = \left(1 - \frac{1}{RRW_j} \right)$$

23

Comparison to other fields

- Computer Systems change frequently
  - No culture of comfiguration management (CM)
    - E.g. how to change specifications and realizations?
    - Often no " specification" at all!
- Possible to automate many of these functions
  - Monitor system status
  - Derive low-level details of the model automaticallty from observations.
- How to integrate CM, PRA and monitoring in the computer systems context?

24

Summary

- What we can do:
  - ordered lists of credible risks
  - consistent definition of " credible"
  - quantitiative measure of component importance
  - Low and high-level representations (to assign work)
- Limitations
  - Still requires knowledge of PRA (but not too complex)
  - Absolute numbers are suspect (meaningless?)
  - Can' t model everything
  - Many events are still hard to model in PRA
    - Repair
    - Common-cause failues
    - Humans

25