Probabilistic Risk Assessment Tutorial

System Safety Conference
Huntsville, AL

September 11, 2001

Todd Paulos, Ph.D.
tpi@ix.netcom.com
(805) 522-9300

AGENDA

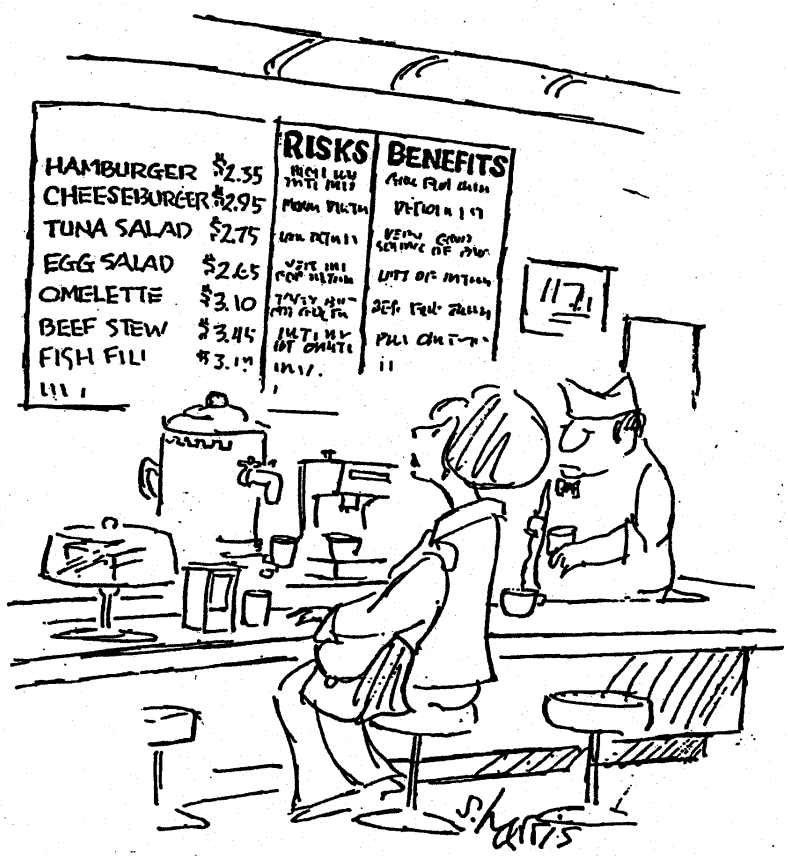
- What is risk?
- What is PRA?
- Introduction to PRA and PRA basics
- PRA Tutorial example
- What about typical reliability analyses?
- Questions

Risks and Benefits

- Risk: Probability distribution of “loss,” e.g., accidents, release of hazardous materials, deaths, environmental contamination, financial, and mission failure.
- Benefits: Those resulting from the activity or systems that poses the risk.

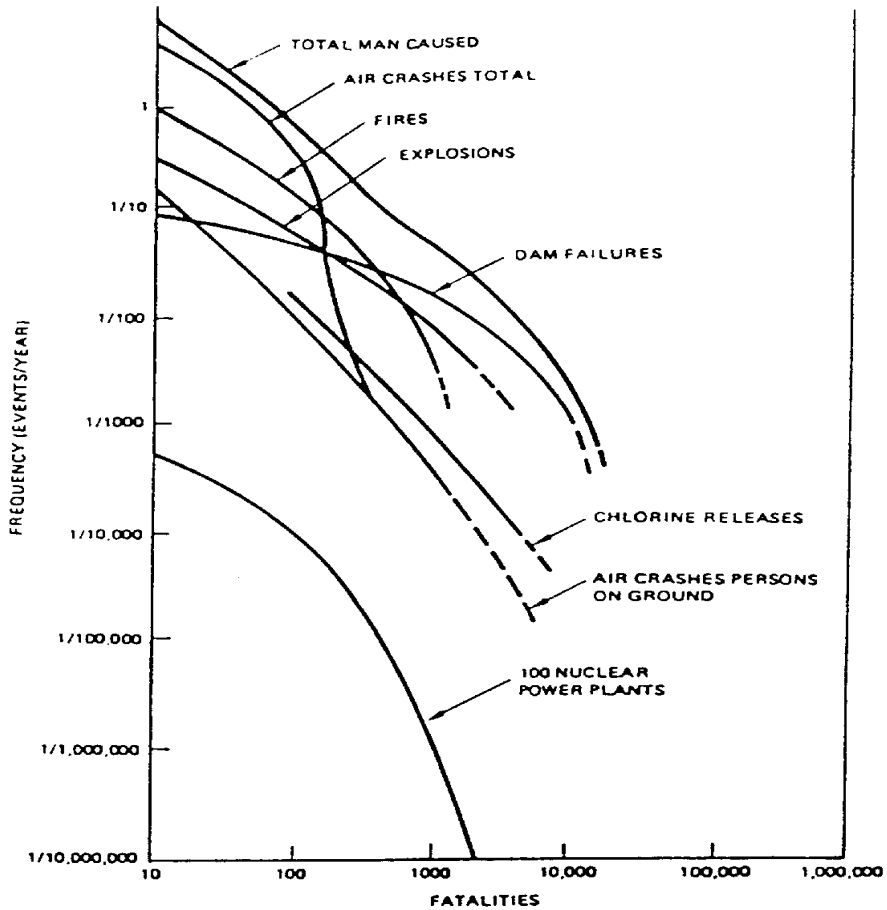
The emphasis is usually on the risks

Are We Too Scared?



Drawing by S. Harris; © 1979 *The New Yorker Magazine.*

The Importance of Risk Communication



Frequency of Fatalities Due to Man-Caused Events (RSS)

The Risks We “Accept”

- *Annual Individual Fatality Risks in Sports*
  - ¨ Hang Gliding: $8x10^{-4}$
  - ¨ Power boat racing: $8x10^{-4}$
  - ¨ Mountaineering: $7x10^{-4}$

The Risks We “Accept” (con’t)

- *Annual Individual Occupational Fatality Risks*
  - Mining: 9×10^{-4}
  - Fire fighting: 8×10^{-4}
  - Police: 2×10^{-4}

The Risks We “Accept” (con’t)

- *Annual Individual Fatality Risks due to Accidents*
  - ¨ Motor vehicles: 2.4×10^{-4}
  - ¨ Falls: 6.2×10^{-5}
- Annual Cancer Fatality Risks
  - ¨ All cancers: 3×10^{-3}

Probabilistic Risk Assessment as a Tool

- PRA is one tool that can be used to help identify risks
- The models generated can be very helpful in communicating risks to the project, engineers, and the outside world

What is Probabilistic Risk Assessment?

- A structured, disciplined approach to analyzing system risk
  - Used for small and simple to large and complex systems, projects and programs
- An investigation into the responses of a system to perturbations or deviations from its normal operation or environment

What is Probabilistic Risk Assessment? (con't)

- A risk model is a system simulation of how a system acts when something goes wrong
  - It captures the knowledge of experts in the system under analysis with respect to how the system should succeed, how it might fail, and how failures may be recovered

The Essence of PRA in Four Questions

- How does the system (process) work?
- What can go wrong? (accident sequences or scenarios)
- How likely are these scenarios?
- What are their consequences?

Risk Assessment and Management

Inputs
- Mission Success Criteria
- Technical Data
- Physical Models
- Cost
- Schedule
- Management Procedures
- Other

Improved Safety & Performance

Risk Assessment
- Initiating Events
- Scenario Modeling
- Risk Quantification
- Uncertainty Evaluation

What Decisions? What Questions?

- Find Best Risk Reduction Strategy
- Go - No Go
- Improve Chance of Successful Mission
- What is the best purchase?
- How can we meet the mission goal?
- Select best Design Process
- How do we improve operation, inspection and maintenance to lower risk?
- What is the Confidence that System will Perform as required?
- Does it meet the Safety Goal?

Milestones

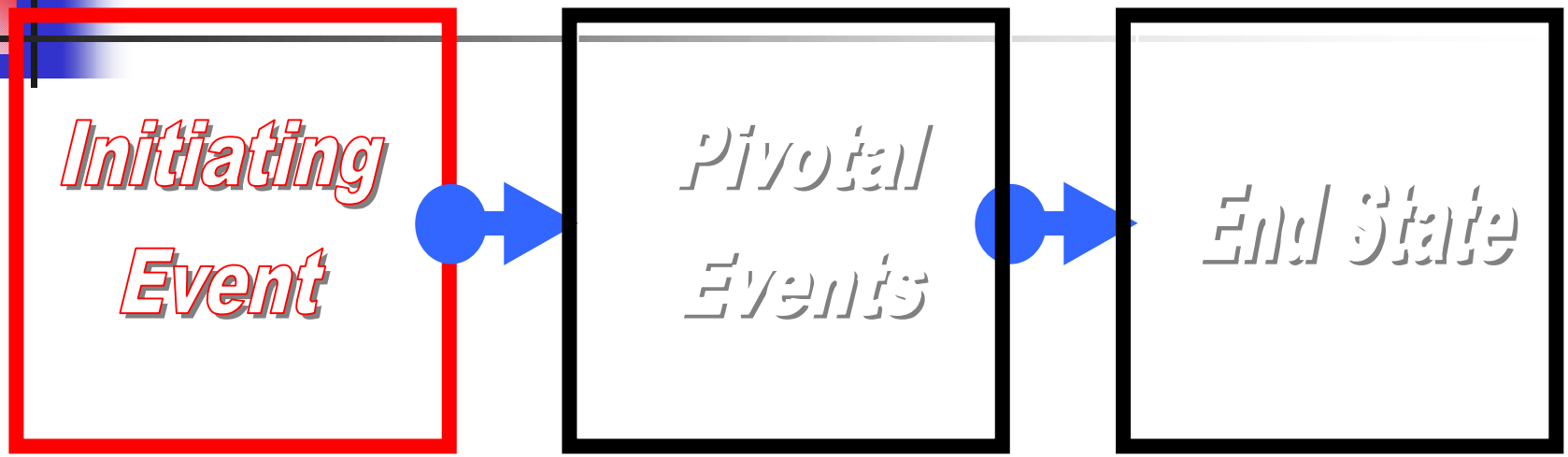
- F.R. Farmer, “Siting Criteria – A New Approach,” International Atomic Energy Agency, Vienna, 1967.
- C. Starr, “Social Benefit versus Technological Risk,” *Science*, 165 (1969) 1232-1238.
- *Reactor Safety Study*, WASH-1400, Nuclear Regulatory Commission, 1975.
- First Modern NASA PRA (Space Shuttle Proof of Concept Study), 1987.

Milestones (con't)

- Weapon System Safety Assessment, BMDO, 1992 to 1999.
- PRA Policy Statement, Nuclear Regulatory Commission, 1995.
- National Research Council, *Understanding Risk*, 1996.
- Tooele Chemical Agent Disposal QRA, US Army, 1996.
- PRA for the International Space Station*, 1999 – present.*

Introductory Concepts of PRA

The Principle of Scenarios

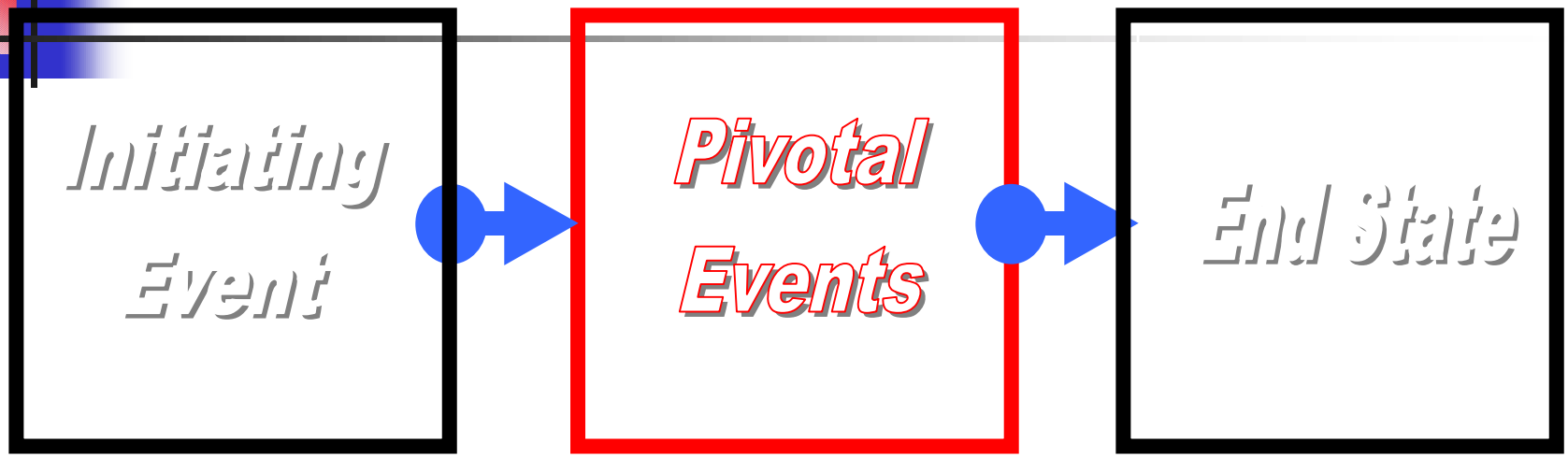


- The Perturbation
- Not always "initiating," may just be an event

- Aggrevative
- Mitigative
- Protective/preventive
- Benign

- Consequence of interest to Decision-Maker

The Principle of Scenarios: What Happens When it Goes Wrong?

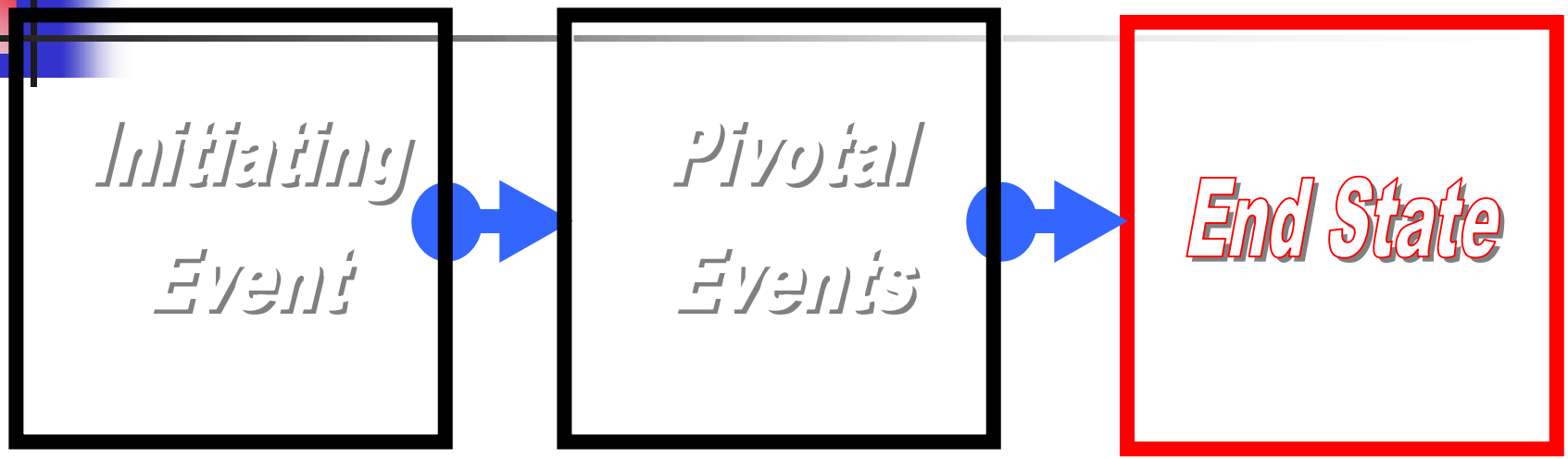


- The Perturbation
- Not always "initiating," may just be an event

- Aggrevative
- Mitigative
- Protective/preventive
- Benign

- Consequence of interest to Decision-Maker

The Principle of Scenarios: What Are the Consequences?



- The Perturbation
- Not always "initiating," may just be an event

- Aggrevative
- Mitigative
- Protective/preventive
- Benign

- Consequence of interest to Decision-Maker

Essential Questions

- “What can go wrong?”
- “What happens when it goes wrong?”
- “How can we recover?”
- “What would be the consequences of things going wrong?”
- “What are the probabilities of things going wrong?”

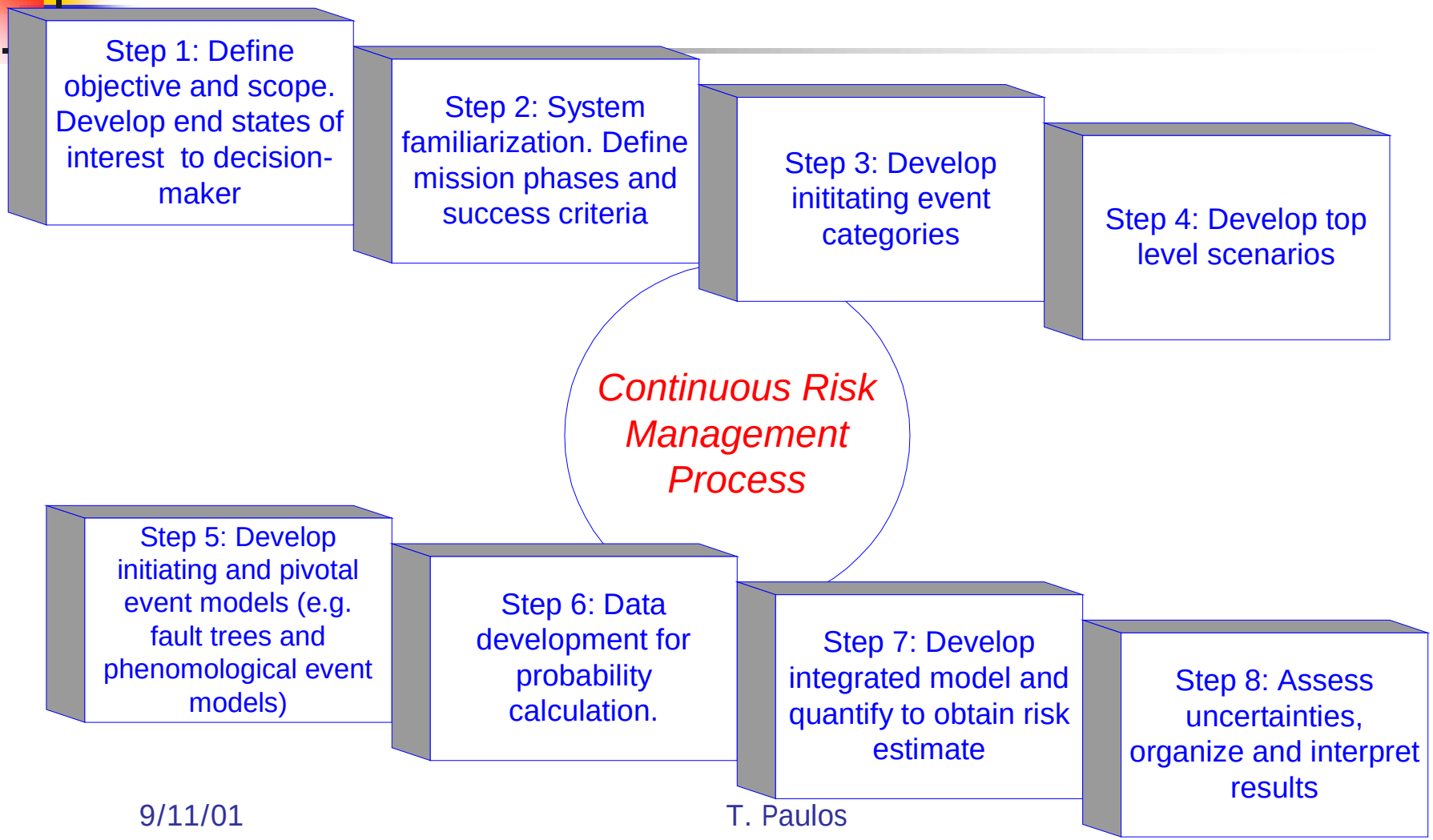
Essential Questions (con't)

"What are the probabilities of the consequences?"

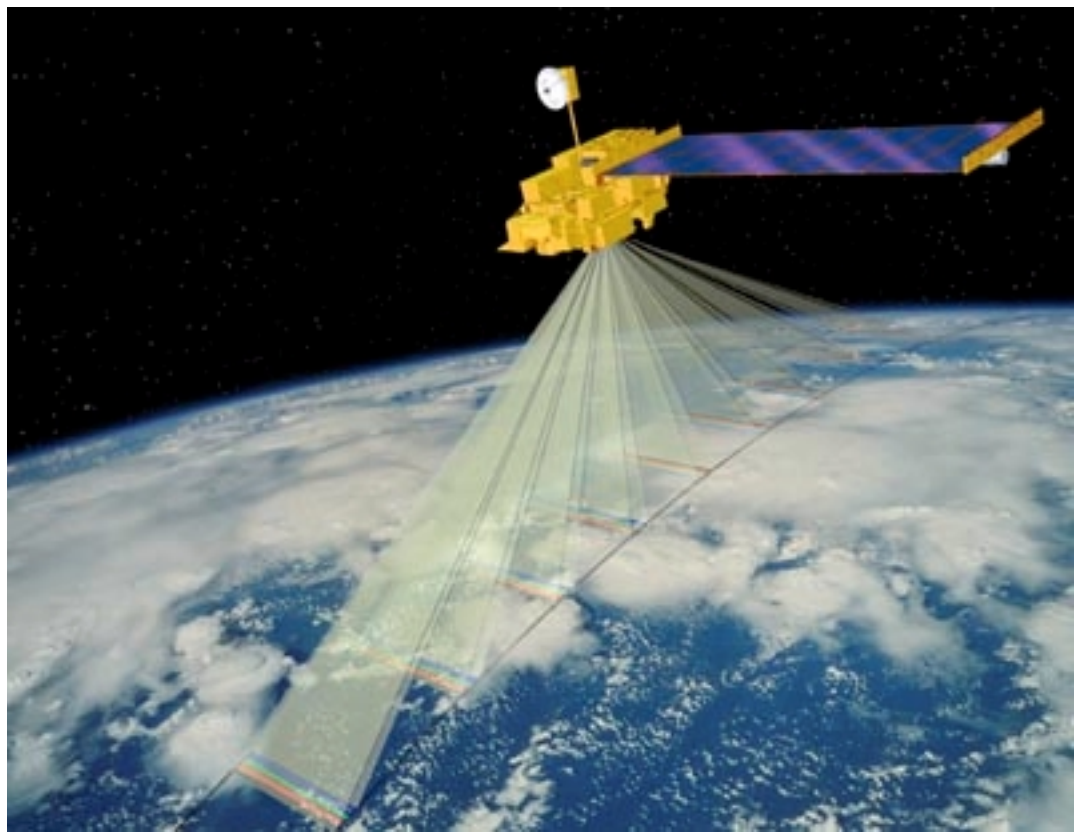
"What are the uncertainties and how do they affect the estimate of consequences and probabilities?"

"What can we do to prevent it from going wrong, or at least reduce the probability or severity of the consequences?"

Eight Steps of a PRA



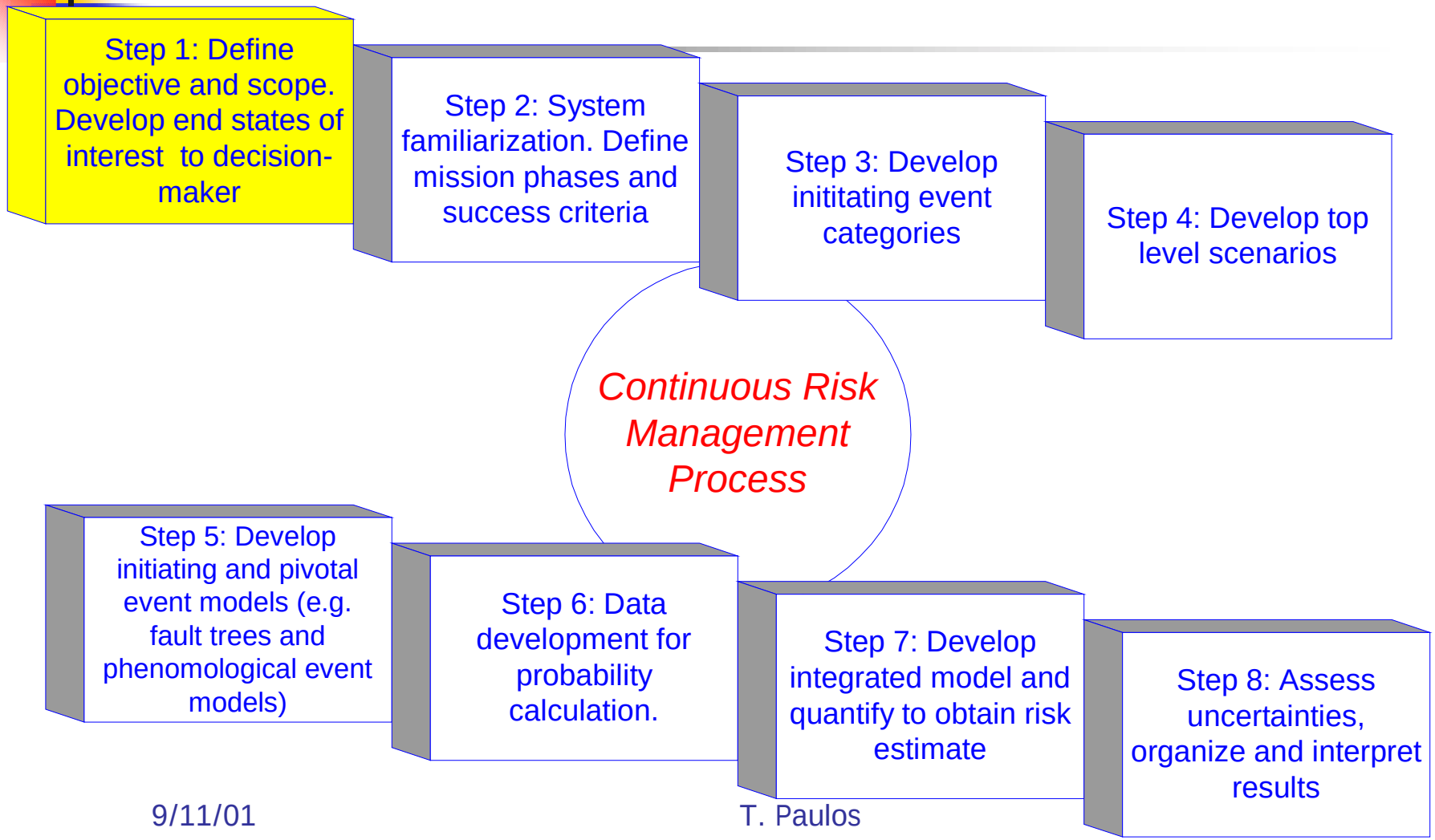
Tutorial Example: Satellite Science Mission



Science Mission

- Place satellite in correct orbit
- Deploy satellite
- Maintain satellite in proper orientation and trajectory
- Perform science
- Transmit science data
- Define 1 year as minimum mission, 5 year possible mission

Step 1: Define Objectives and Scope



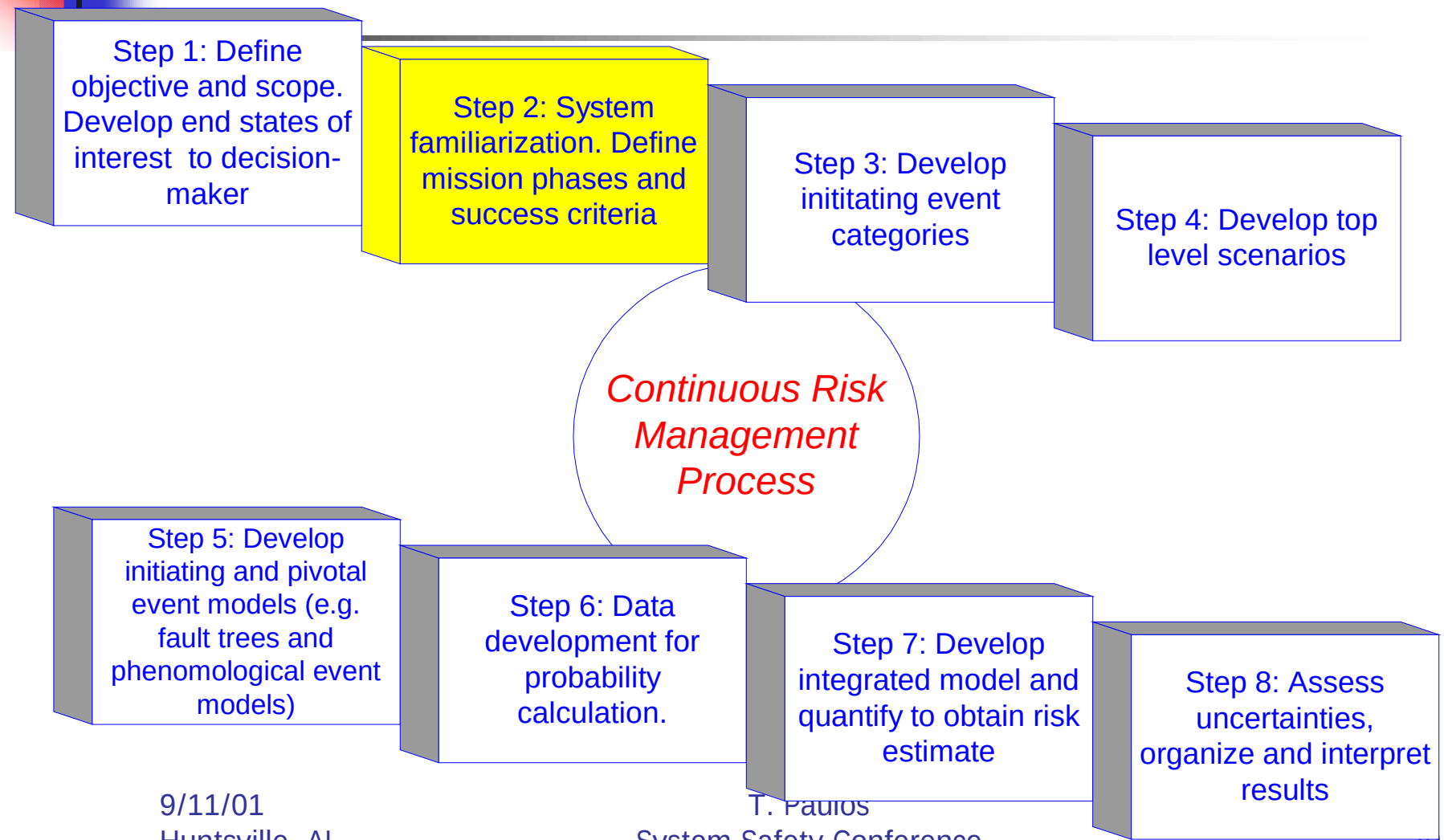
Delta II Launch Vehicle

- Consider launch, separation and correct orbit placement out of scope
  - Historical perspective
  - Plenty of real world data

Objective and Scope

- Include:
  - Satellite subsystems
  - Science instrumentation
  - Minimum mission 1 year, maximum 5 years
- Out of scope
  - Launch vehicle (other than data points)
  - Human reliability
  - Availability of ground stations
  - Software

Step 2: System Familiarization



System Analysis

- Never underestimate the importance of understanding the system
  - What components comprise the system?
  - How do the components and system operate?
  - How does the system interact with other systems?
  - What functions does the system perform?
  - How does the system fail?
    - Hardware
    - Software
    - Human errors
  - What external events is the system susceptible to?

System Familiarization

- Satellite subsystems
  - AOC: Attitude & Orbit Control
  - COM: Communications
  - OBDH: On-Board Data Handling
  - PWR: Power
- Science Instrumentation
  - Radar

Satellite Operations

- Satellite receives guidance, navigation and control information from ground
- Satellite must periodically calibrate radar (every 6 months)
- Little autonomous control (satellite waits for instructions from ground if it senses that something is wrong)
- No need for thermal control subsystem

Dependency Matrix

Each column is supported by the elements below	AOC	COM	OBDH	PWR	RAD
AOC	1			X	
COM	1	X	X	X	2
OBDH	1	X	X	X	2
PWR	1	X	X	X	2
RAD					2

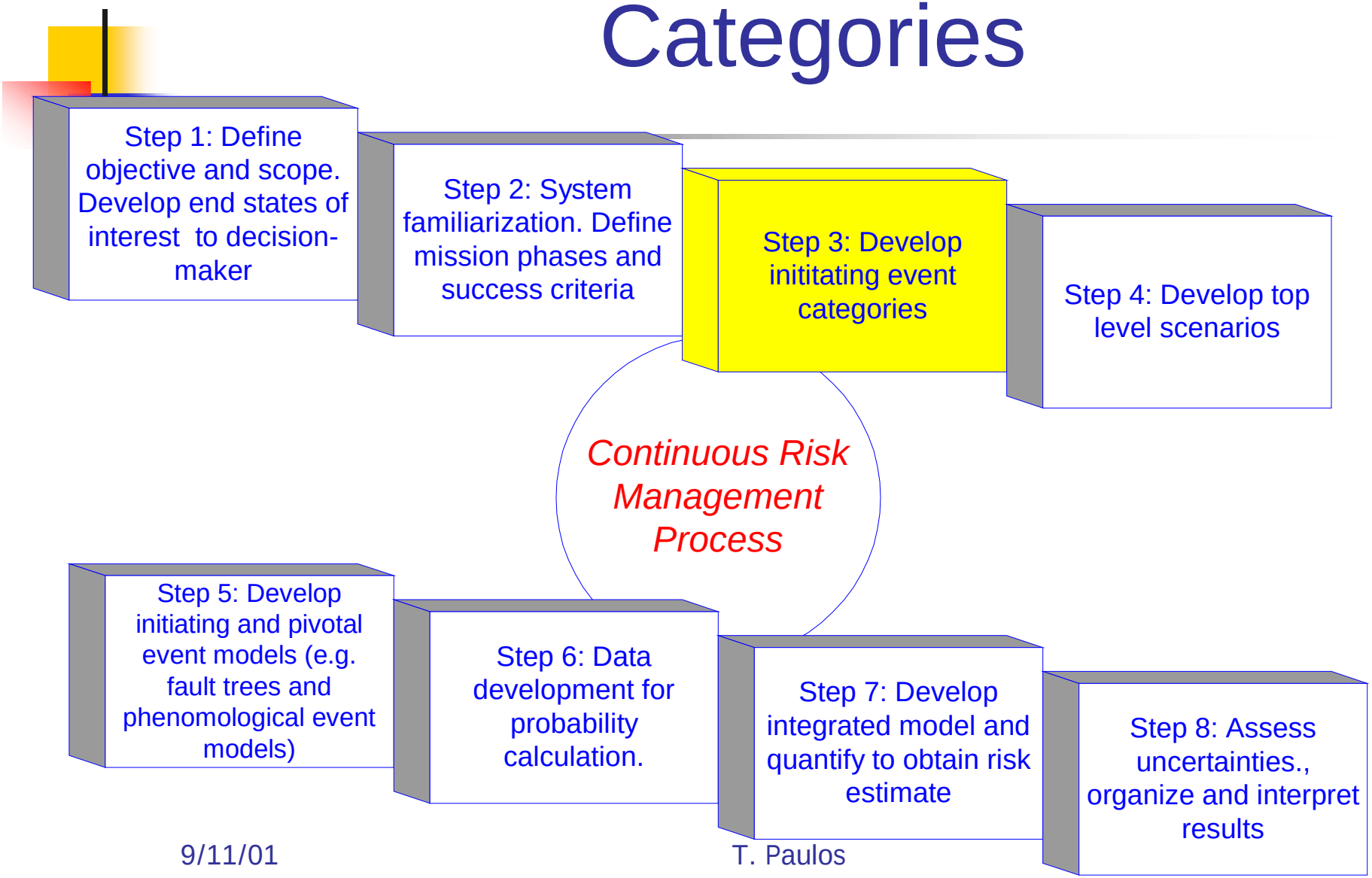
Dependency Matrix Notes (1)

- AOC is supported by several subsystems
  - AOC supports itself through attitude determination, propellant storage, propellant regulation, and propellant delivery, and orbit maneuvers
  - Supported by COM since it receives orbit maneuver information from the ground
  - Supported by OBDH since it handles the data received from the ground
  - Supported by PWR since valves are electromechanically operated, and sensors require power

Dependency Matrix Notes (2)

- RAD is supported by the following
  - Itself since the RAD has a radar, antenna, and other components of the system
  - COM since calibration data is uploaded from the ground, as well as data being downloaded
  - OBDH since it stored data going to ground and commands from the ground
  - PWR supplies the necessary power

Step 3: Initiating Event Categories



Initiating Events

- The process of determining Initiating Events (IEs) is very important
  - Necessary to define the scope of scenarios that will be developed
  - Completeness of IEs is key to a successful PRA

Initiating Events (con't)

- IEs can be any system perturbation or performance function, but they will be categorized according to system response
- Events and consequences consider the entire spectrum of severity, unlike FMECAs or Hazard Analyses

Determining Initiating Events

- Experience
  - Individual or a group
  - Background and experience
  - “Brain storming”
  - Expert elicitation
  - Knowledge of past accidents and failures
  - System simulations

Determining Initiating Events (con't)

- Checklist
  - Use of generic checklists, system knowledge or previous mishaps to create a possible list of initiators
- FMEA (MIL-STD-1629)
  - Detailed inductive approach in which the design is reviewed to determine failure modes at a functional level, system level or subsystem level, all the way down to a component

Determining Initiating Events (Cont'd)

- Identifies the worst possible affect on the next higher level of assembly
- Effective for identifying initiating event categories in the PRA
- Ineffective for identifying system interactions, common cause and other dependent failures, human interactions, environmental and external influences

- Hazard analyses can help determine IEs

Internal vs. External Events

- Internal initiators
  - Component failures
  - Human failures
  - Software failures
  - Hazardous conditions

Internal vs. External Events (con't)

- External initiators
  - Things that begin outside the system boundary
    - MMOD
    - Radiation
    - Lightning
    - Etc.
  - Acts of nature or external events which begin outside the system boundary

Screening/Grouping

- Evaluate events deemed out of scope to eliminate them from the analysis
  - Earthquakes
  - Floods
  - Random acts of God
- May be necessary if there are different levels of system responses for similar initiators
- All initiators that have the same response should be grouped together

Initiating Events/Events

- Not all events are “initiating” in the literal sense, some are just events that need to occur (e.g., fails to perform science mission)
  - Functions are not perturbations to the system
- Keep in mind the dynamic nature of missions and platforms

Tutorial Example Events

- Launch vehicle/separation failure
- Deployment failure
- Subsystem failures
  - AOC
  - COMM
  - OBDH
  - PWR
- Loss of science

Master Logic Diagrams

- Master Logic Diagrams are a hierarchical depiction of ways in which system perturbations occur
- Shows the relationship of lower levels of assembly to higher levels of assembly and system functions
- Begin with a top event (end-state of concern)
- Events that are necessary but not sufficient to cause the top event are enumerated in ever more detail as the lower levels of the hierarchy are built

Master Logic Diagram (cont'd)

- Developed to identify Initiating Events in a PRA
- Hierarchical depiction of ways in which system perturbations can occur
- Good sanity check for completeness
- Communication tool between the analyst, the engineer and management that all significant events have been considered

Development

- Begin with a top event that is an end state
  - Loss of Mission
  - Minimum Mission
  - Loss of Vehicle
  - Loss of Crew
  - Cost Overrun

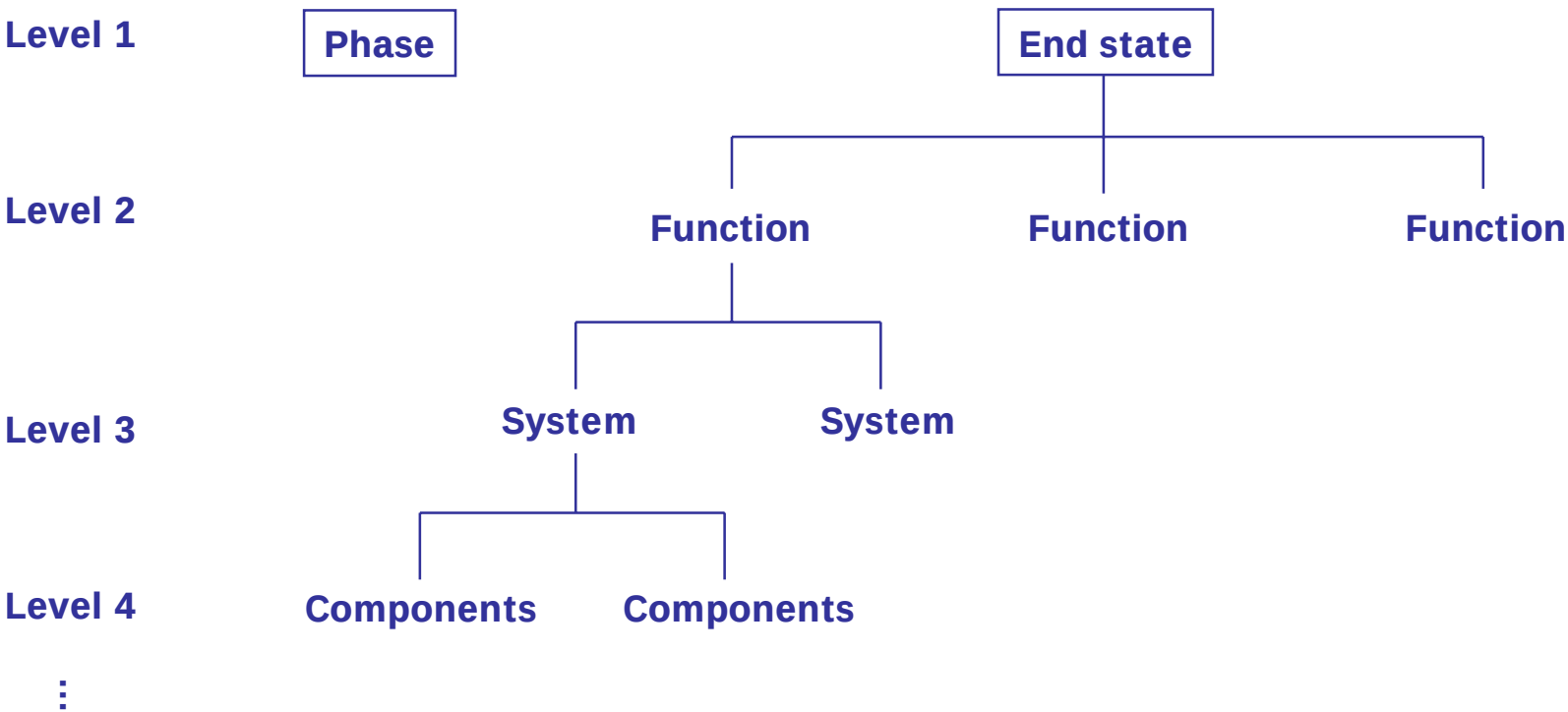
Development (con't)

- The top levels are typically functional
  - Failure to contain
  - Failure to control
  - Failure to cool
- Develop into lower levels of subsystem and component failures

Pinch Point

- The trick in developing a useful MLD is knowing when to stop at a reasonable level
- The “pinch point” occurs when every level below the stopping level has the same consequence as the level above it

Example MLD


Level 1
Phase
End state
Level 2
Function
Function
Function
Level 3
System
System
Level 4
Components
Components
⋮

Tutorial Example of MLD

Orbit Phase

Loss of Mission

Loss of Satellite
- **Subsystem failure**
  - **AOC (expand)**
  - **COM (expand)**
  - **OBDH (expand)**
  - **PWR**
    - **Simple component failure (expand)**
    - **Failure to charge battery**
    - **Battery overcharging (expand)**
    - **Excess power drain (expand)**
    - **Shorts (expand)**
    - **Electro-Static Discharge (ESD) (expand)**
    - **Loss of support from subsys (expand)**
    - **Power regeneration failure (expand)**
    - **Etc.**

Loss of Science
- **Radar system (expand)**

MLD (con't)

Failure to charge battery

- **Battery failure**
- **Solar array misaligned / damaged**
- **Rupture / explosion**
- **Leakage of electrolytes**
- **Low voltage in charging system**
- **Excessive power drain**
- **Battery too hot or too cold**
- **Power spike into system**
- **Etc.**

Example Pinch Point

- For example, consider electrolyte leakage
  - Do we need to consider the reasons why electrolytic leakage occurred, such as overcharging, physical damage or MMOD, case failure, etc.???
  - No, since no matter what the cause, the end result is the same
  - Typically a component and failure mode is a good pinch point

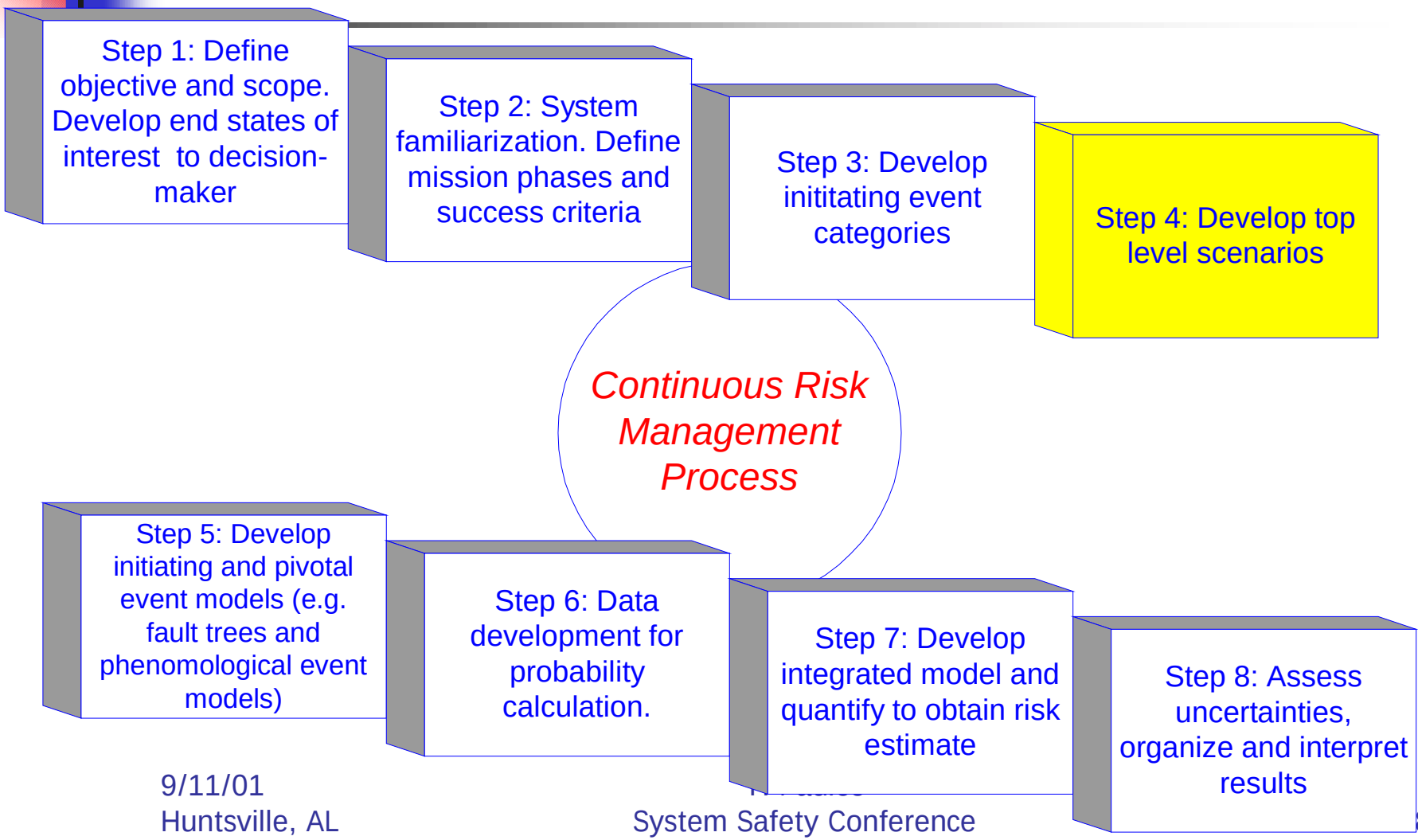
Final Note on MLDs

- MLDs are really brain storming sessions, so try to be as thorough as possible; you can edit and define pinch points later
- Do not forget to consider things such as environmental or external events
- Remember that the subsystems work together and that failures can cross system boundaries
- Understand how the system is operated

Final Word on Initiating Events

- The final list of initiators (the set used to develop event sequences) MUST BE MUTUALLY EXCLUSIVE
  - Most PRA codes treat the IEs in this fashion
  - It is a difficult task, if even possible, to after the fact go back and examine the minimal cut sets to determine what is mutually exclusive, and what is not
  - Binning

Step 4: Top Level Scenarios



What is a Scenario?

- Basically stated, scenarios are generally strings of events which lead to some kind of conclusion
  - The starting point for a scenario is called the initiating event
  - Every scenario ends in an end state or a damage state which are defined by the analyst

What is a Scenario? (con't)

- For example, loss of vehicle or loss of crew are end states and loss of system redundancy could be an intermediate state

– Between the initiating event and the damage state are pivotal events which may be protective, mitigative, aggravative, or benign.

- Scenarios may be documented by the use of Event Sequent Diagrams or event trees / fault tree combinations.

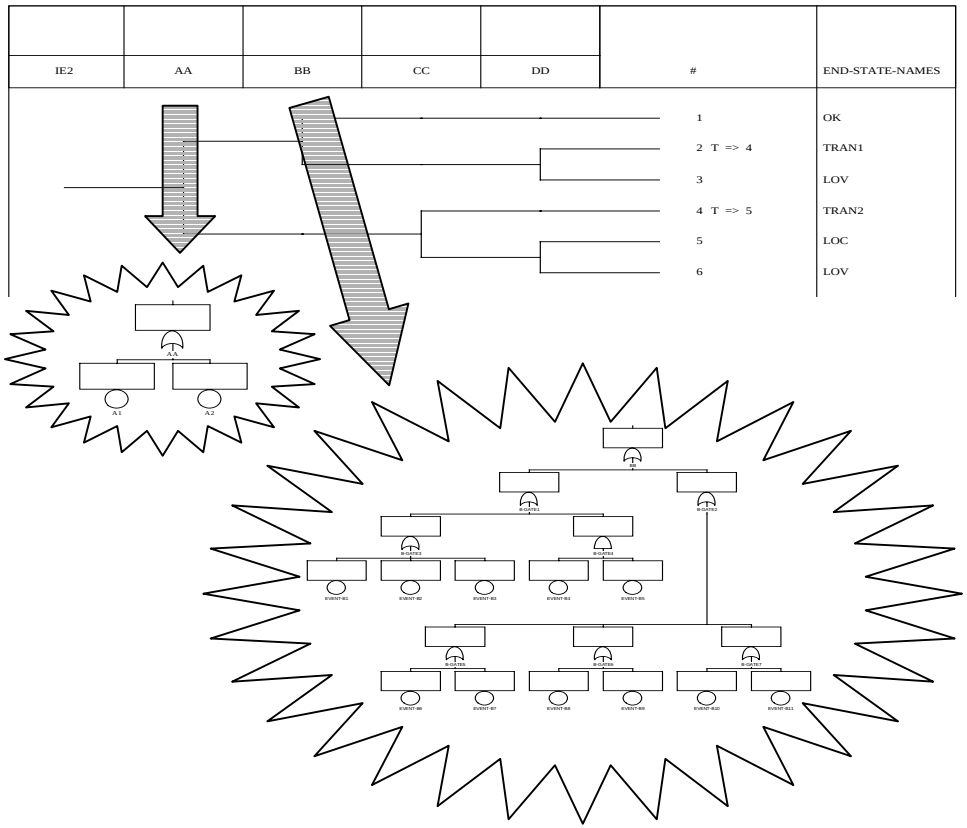
Developing Risk Scenarios

- What are the initiators?
  - Internal vs. External
  - MLD can organize
- What are the systems involved?
  - Dependency Matrix
- What are the success criteria?
  - Functional or phenomenological

Developing Risk Scenarios (con't)

- What are the transition and end states?
  - Loss of Crew
  - Loss of Mission
  - Loss of Vehicle
  - Etc.

Risk Models


IE2
AA
BB
CC
DD
#
END-STATE-NAMES
1
OK
2 T => 4
TRAN1
3
LOV
4 T => 5
TRAN2
5
LOC
6
LOV

Risk Models

- The trick is developing the “right” size ETs and FTs for the system at hand
- Steady state systems
  - Probably modeled better using larger fault trees
- Dynamic systems
  - More effective to use larger event trees
- Every application is different, and part of the PRA “art” is to accurately model the world using the combination

What is an Event Tree?

- An Event Tree is an *inductive logic model* used in reliability and risk assessments to provide organized displays of sequences of system and operator failures and successes that can lead to specific outcomes
- An Event Tree model is inductive because it starts with the premise that some failure has already occurred and then maps out what could occur in the future if further systems failed or succeeded

What is an Event Tree? (con't)

- The Event Tree identifies accident sequences (or scenarios) leading to different potential outcomes
- The accident sequences form part of the Boolean logic which allows the systematic quantification of risk

What is a Fault Tree?

- A *Fault Tree* is a *deductive logic model* whereby a system failure is postulated (called the *Top Event*) and reverse paths are developed to gradually link this consequence with all subsystems or components (in order of decreasing generality) that can contribute to the top event down to those whose basic probability of failure (or success) are known and can be directly used for quantification

What is an Event Sequence Diagram?

- For all practical purposes Event Trees and Event Sequence Diagrams are equivalent
  - Event Sequence Diagrams are more graphical in nature and tend to be easier for the engineer to review
  - Event Sequence Diagrams typically contain a lot of detailed explanation for those unfamiliar with the system

What is an Event Sequence Diagram? (con't)

- PRA models use Event Trees for quantification
- Event Trees can be used to group events in the Event Sequence Diagram for simplification
- For every Event Sequence Diagram, there is an equivalent Event Tree, and vice versa

Event Tree and Fault Tree Combinations

- Event Trees
  - Depict a chronological sequence of events, such as a system response
- Fault Trees
  - Analyze higher level events into combinations of component failures

Event Tree and Fault Tree Combinations (con't)

- Why not model a system using only event trees or only fault trees?
  - Eventually the complexity and dynamic elements of a system exceed the ability of only using ETs or FTs to provide an accurate risk model
  - The combination of event trees and fault trees is most effective in modeling complex systems

Developing Event Sequences

- List of initiators (functions, perturbations, failures, etc.)
- For each group of initiators, perform a series of “If-Then” or “Yes-No” statements
  - If this happens, how does the system respond?
  - What are the pivotal (preventative, aggravating or mitigating) events?
  - What are the transition and end states of this system?
  - How does this process continue until all possible scenarios are developed?

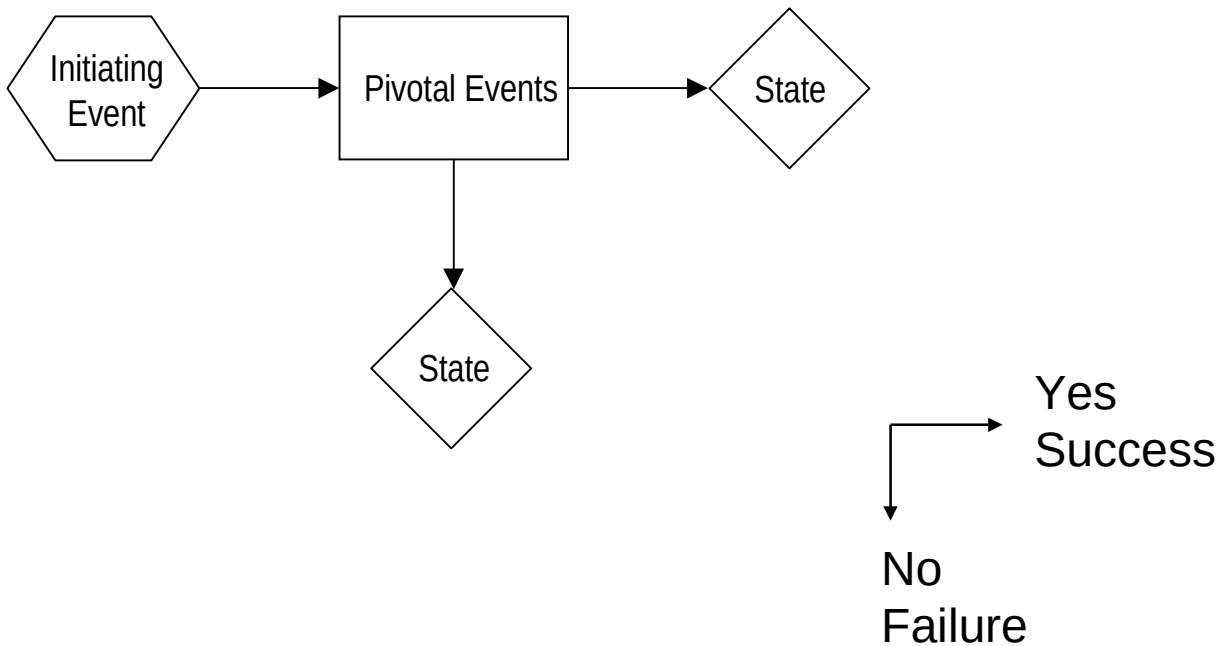
Developing Event Sequences (Cont'd)

- Things to consider in developing event sequences
  - How can the combination of event trees and fault trees and best be used to develop the model of the world?
  - What are the success criteria and how do they affect the model of the world and transition states?
  - What is the chronological order of possible events?
  - What are the pivotal events and recovery actions?

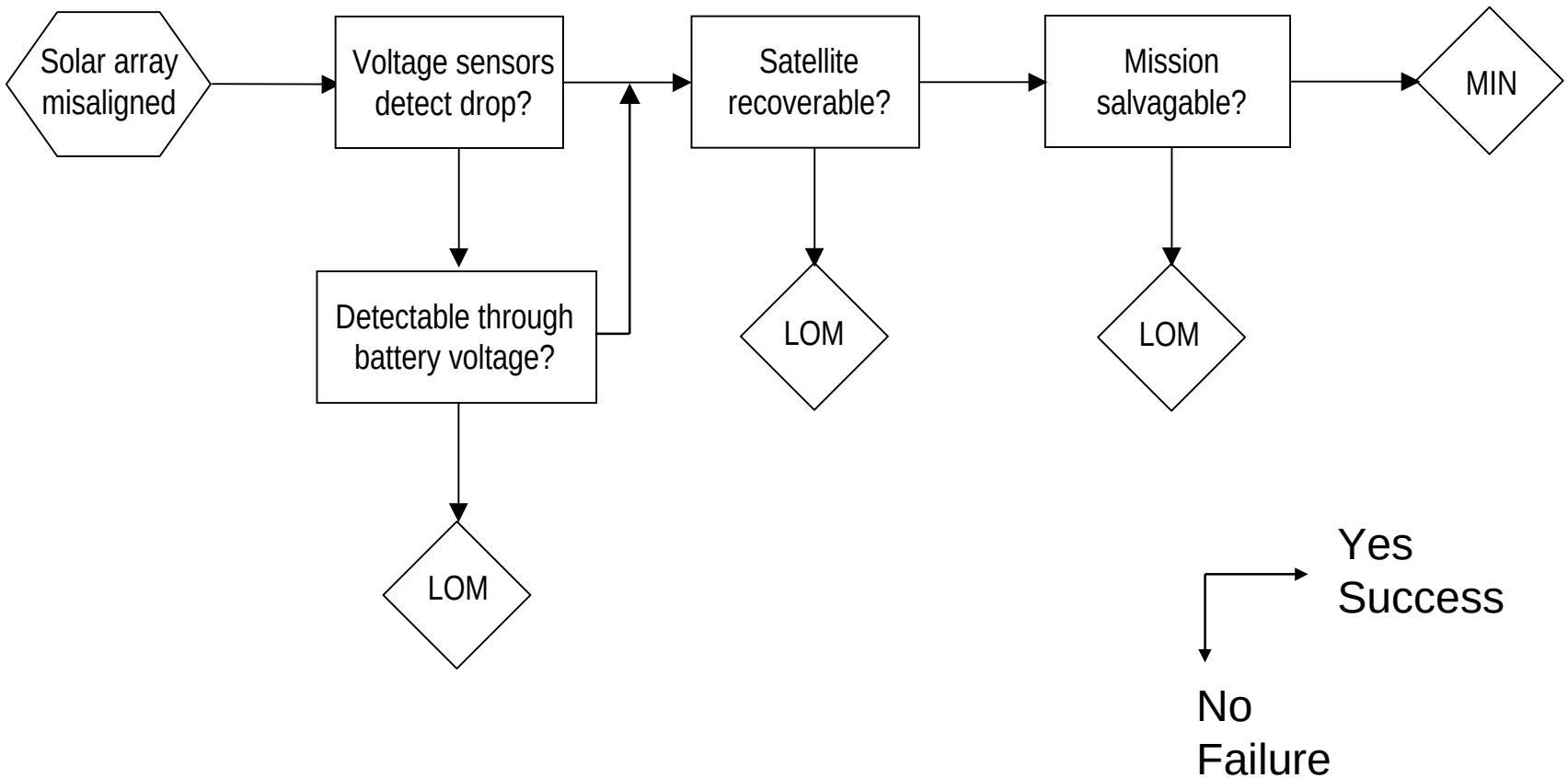
End States

- End states define the outcome (relative to mission success) of each event sequence
- Typically, many different sequences can lead to the same end state
- End states are absorbing states by nature, unlike transition states which depict the various states of the system operation

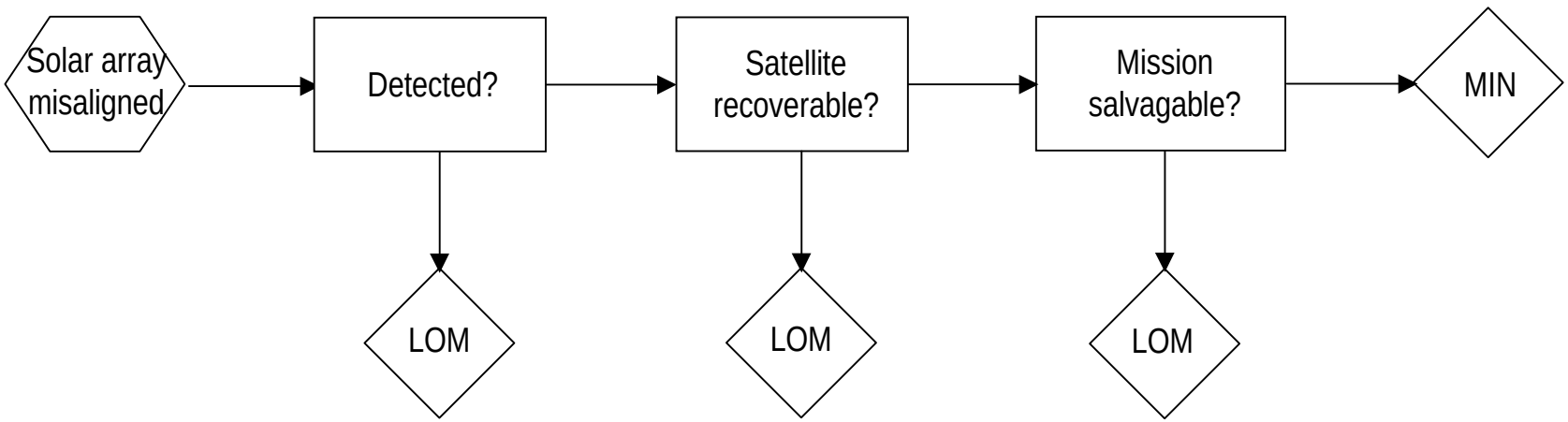
Event Sequence Diagrams



Tutorial Example: Solar Array Misaligned

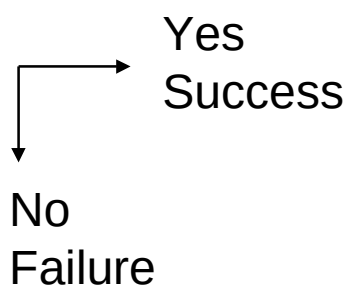


Tutorial Example: Solar Array Misaligned

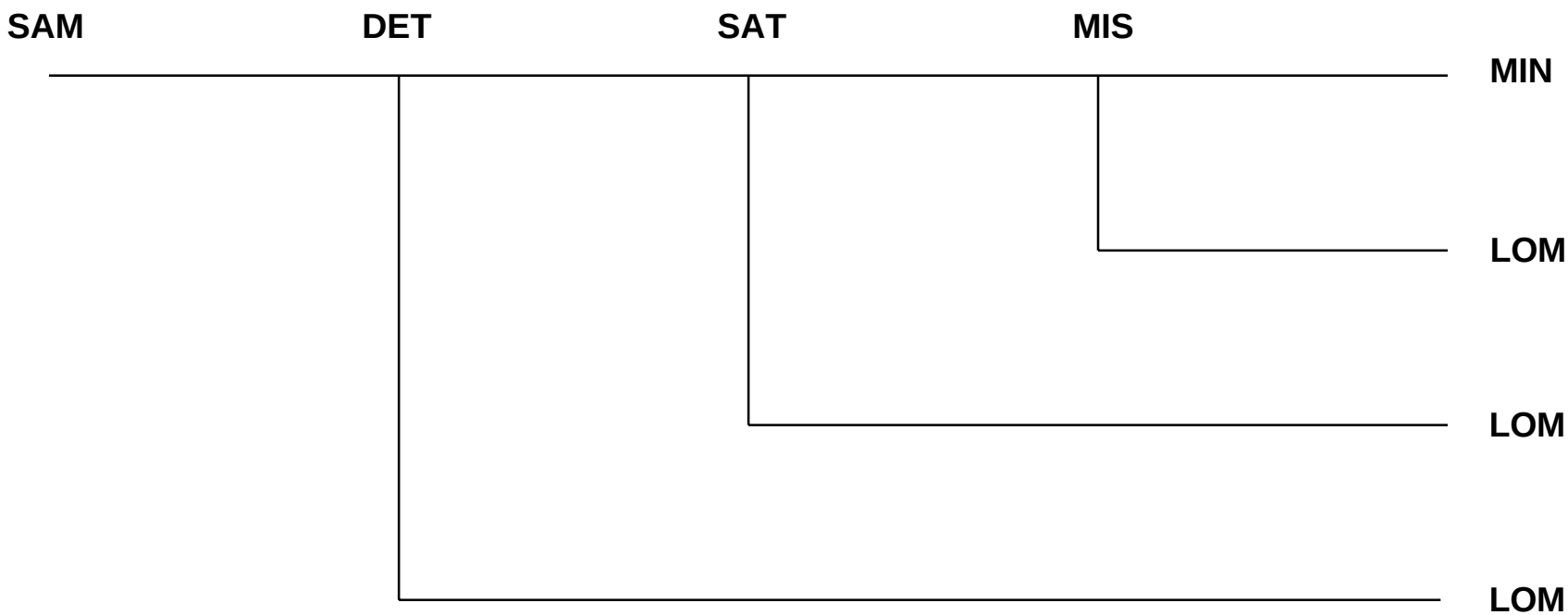


Notes:

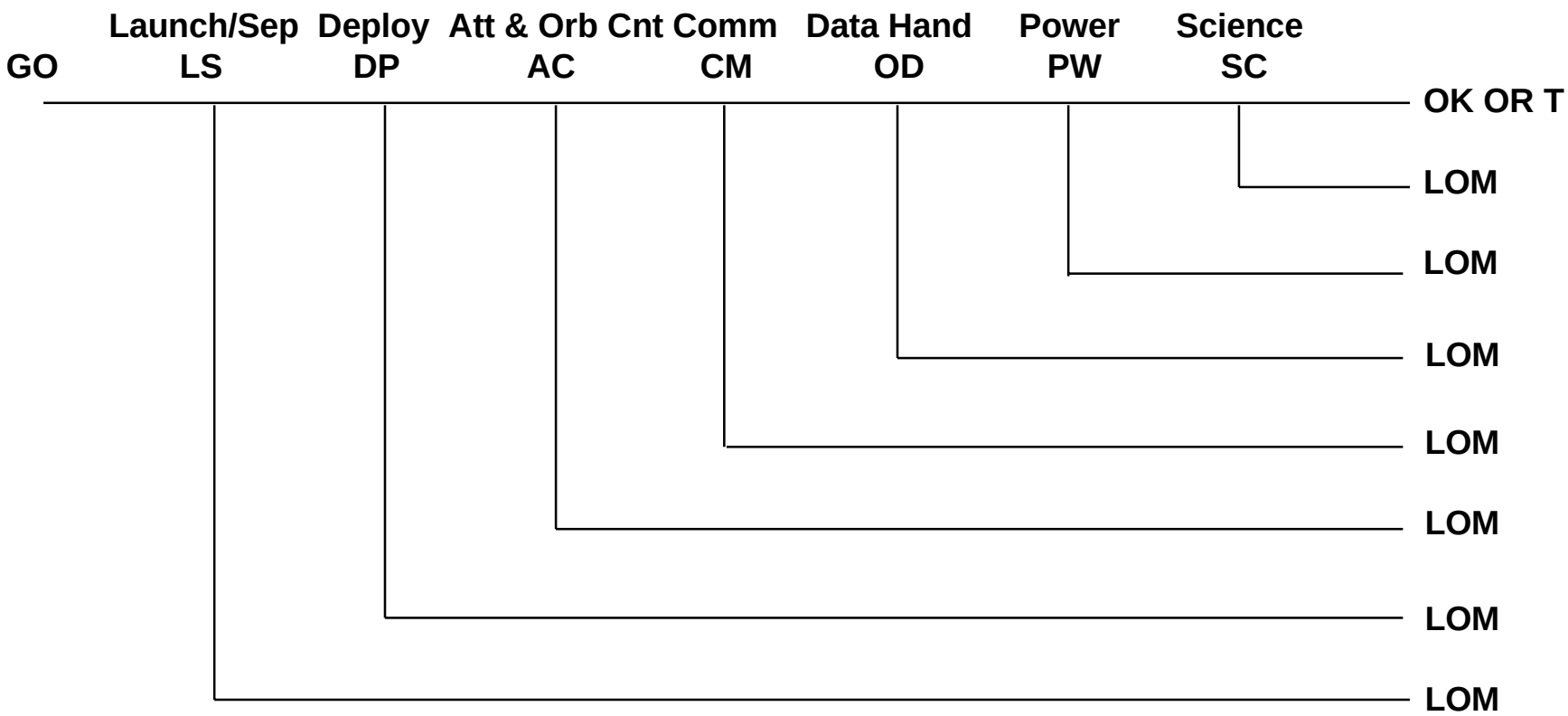
- **Events could be expanded if desired**
- **Can we ensure that this IE is mutually exclusive from other IEs?**



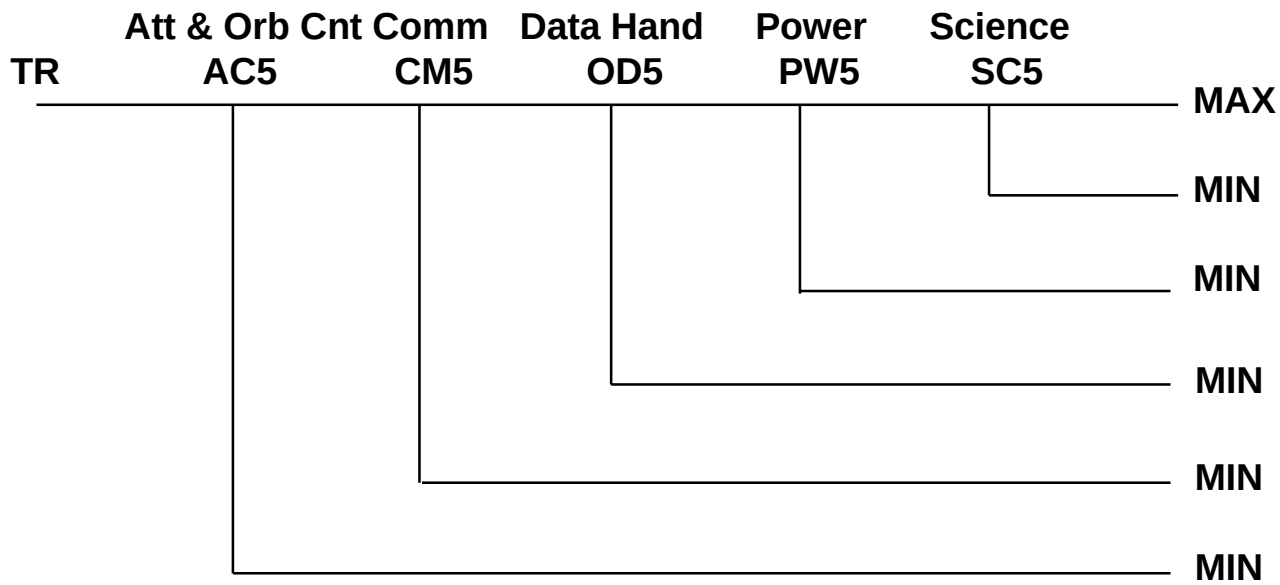
Tutorial Example: ESD to Event Tree



Tutorial Example: Mission Event Tree



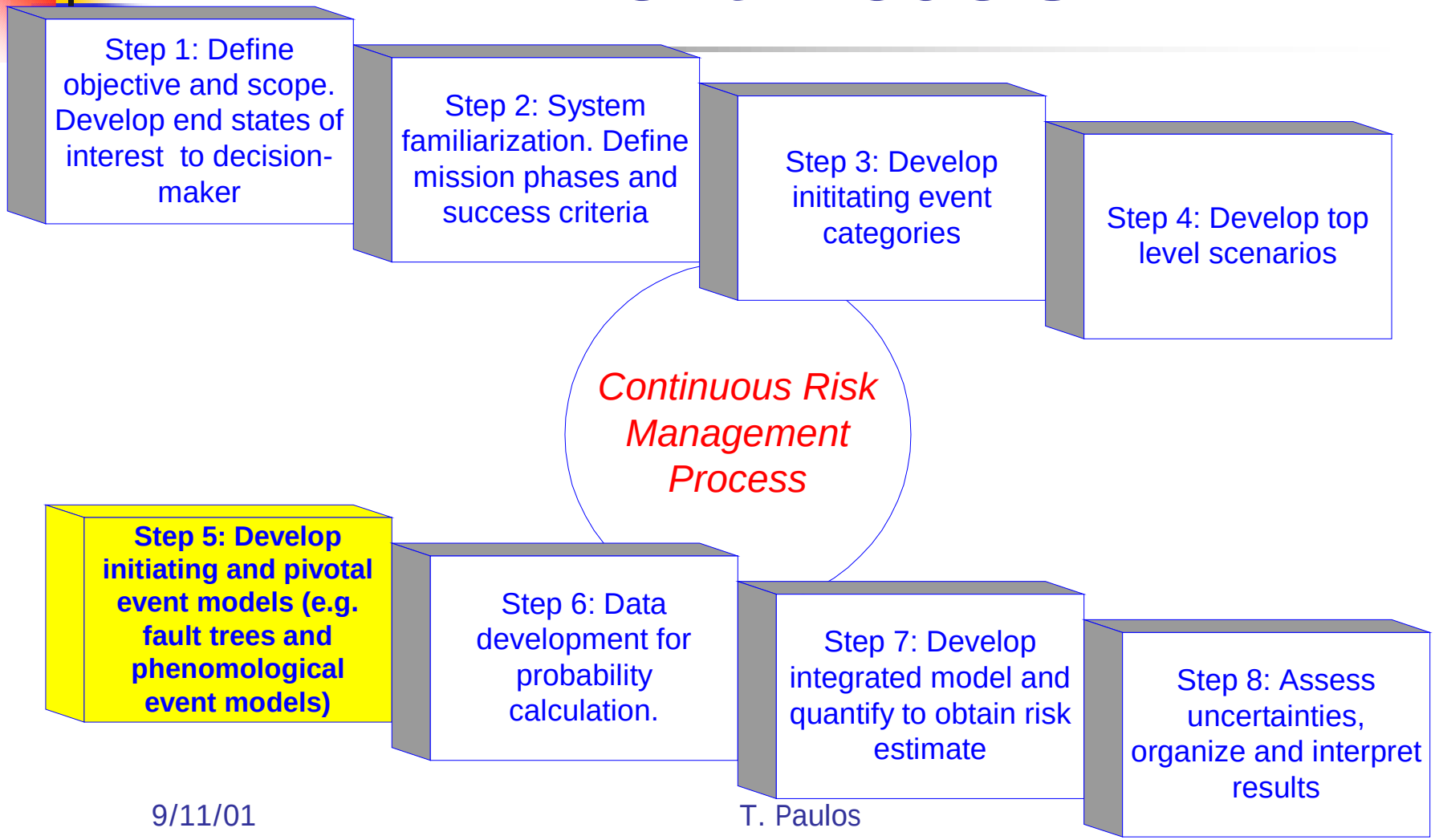
Tutorial Example: Mission Event Tree (con't)



Final Notes on Event Trees

- Do not be afraid of the large event tree approach
  - It may be easier for engineers to understand functional events as opposed to very large fault trees
  - Can collapse events at the end if desired (possibly easier for upper management to understand), but very difficult to expand events later

Step 5: Initiating and Pivotal Event Models



Fault Tree Structure

- Graphically, a fault tree consists of:
  - Rectangles containing descriptions of subsystem or module failure
  - Circles depicting basic unit (or component) failures
  - Binary logic gates (union or intersection) connecting the circles with the rectangles, and the rectangles with each other up to the top event

Fault Tree Guidance

- A good fault tree analyst must be able to
  - Model all important failure modes
  - Only develop detailed models when necessary
- Single point failures are ultimately connected to the top fault tree event through a series of OR gates
- Failure modes that require multiple events are connected to the top fault tree event through a series of AND gates
- The fault trees should be developed down to the level where either data are available, or is required by the project, consistent with data availability

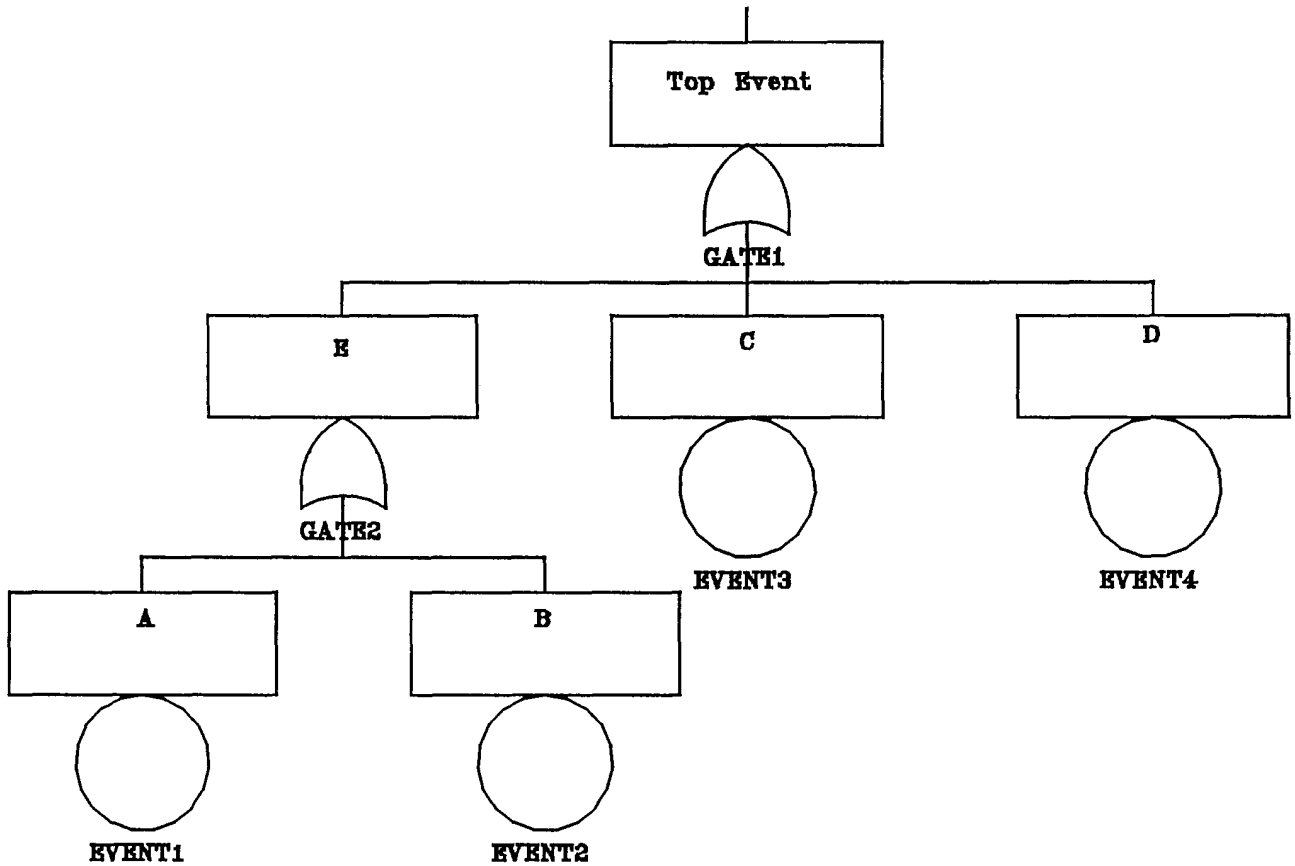
Fault Tree Construction Process

- Fault Trees are constructed to define all possible failure combinations that lead to the “Top Event” -- typically the failure of a particular system to function in performing a particular mission
- A typical mission is defined in terms of success criteria, such as:
  - Satellite needs all support systems for survival
  - Satellite needs 1 of 2 strings in the power distribution subsystem working for survival
  - Satellite needs 1 of 2 antennas working for communications (may need 2 of 2 for certain orientations)
  - Satellite minimum mission is 1 year, maximum is 5

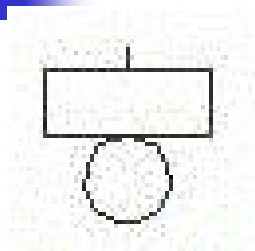
Fault Tree Construction Process (Cont'd)

- Top event failure logic is established from the Boolean complement of the success criteria, e.g.:
  - 2/2 Power distribution subsystems fail
  - 2/2 Antennas fail
- Deductively identify all significant faults leading to the top event
- Basic events are named to facilitate numerical Boolean reduction

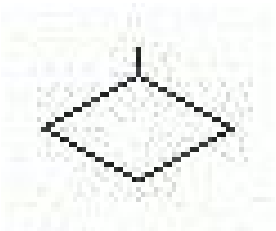
Typical Fault Tree Structure



Typical Fault Tree Symbols

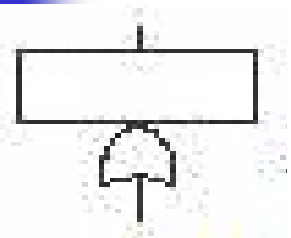


Boxed Basic Event--An alternate symbol for a basic event is a boxed basic event that provides a box to contain the description of the basic event

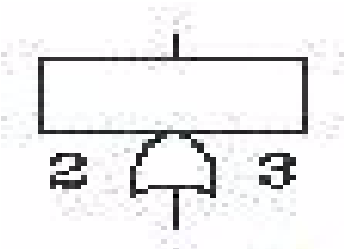


Undeveloped Event--The undeveloped event denotes a basic event that is actually a more complex event that has not been further developed by fault tree logic. IRRAS treats this event as a basic event

Typical Fault Tree Symbols (Cont'd)

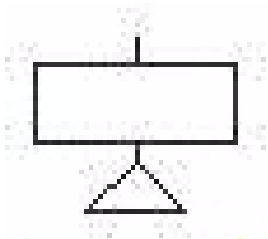


OR Gate--Any one input to the OR gate will cause failure to occur

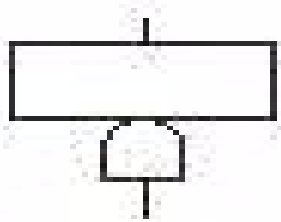


N/M Gate—This gate states that N of M input events must occur for failure to occur. For a 2/3 gate, any combination of 2 out of the 3 input events must occur

Typical Fault Tree Symbols (Cont'd)



Transfer Gate--The transfer gate indicates that logic is continued on a new page, or on the same page. The transfer gate is the same as the gate where the logic continues, and when transferring to another page, the gate being transferred to must be the top gate on the page



AND Gate--All inputs to the AND gate must occur for failure to occur

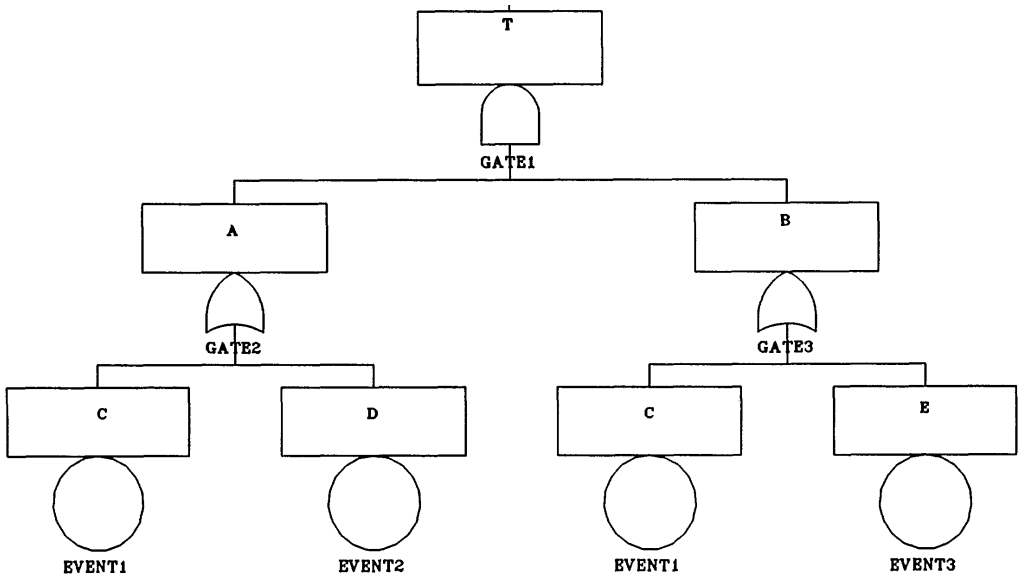
Boolean Reduction and Cut Sets

- The evaluation of a fault tree can be accomplished in two major steps
  - Reduction
  - Quantification
- A collection of primary events (failures) whose simultaneous occurrence guarantees the occurrence of the top event (failure) is called a *cut set*
- *Minimal cut sets* are cut sets containing the minimum subset of primary elements whose simultaneous occurrence guarantees the occurrence of the top event

Boolean Reduction and Cut Sets (Cont'd)

- Boolean (or Logic) Reduction of a fault tree has the objective of reducing the fault tree to an equivalent form which contains only minimal cut sets. This is accomplished by sequential application of the basic laws of Boolean algebra to the original logic expression of the fault tree until the simplest logical expression emerges.
- Quantification of the fault tree is the evaluation of the probability of the top event in terms of the probabilities of the basic events using the reduced Boolean expression as described above.

Example of Fault Tree Reduction



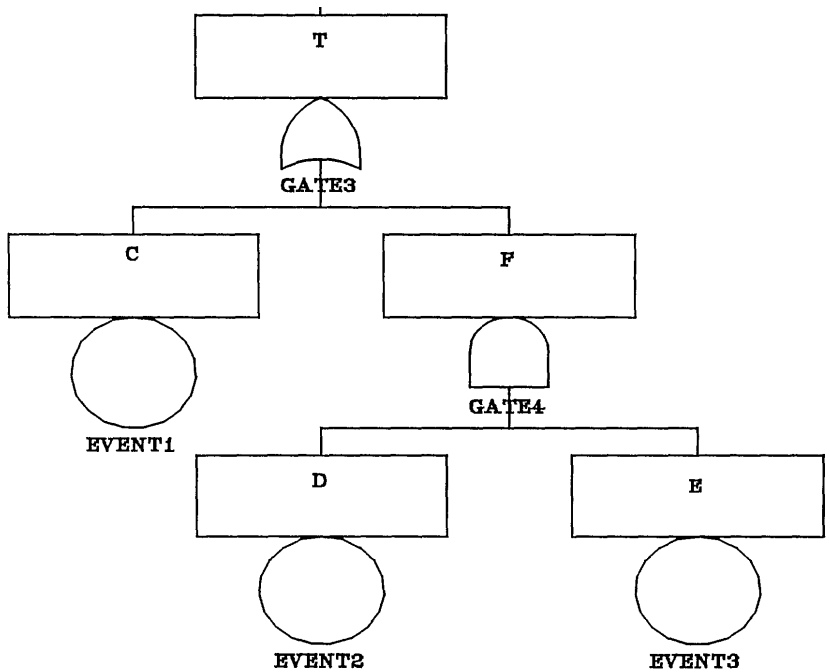
$T = A \cap B = (C \cup D) \cap (C \cup E) = C \cup (D \cap E)$

using the Distributive Law

The cut sets are: (C,D), (C,E), (C,D,E)

Example of Fault Tree Reduction (Cont'd)

- The reduced fault tree is:



- The minimal cut sets are: (C) and (D,E)

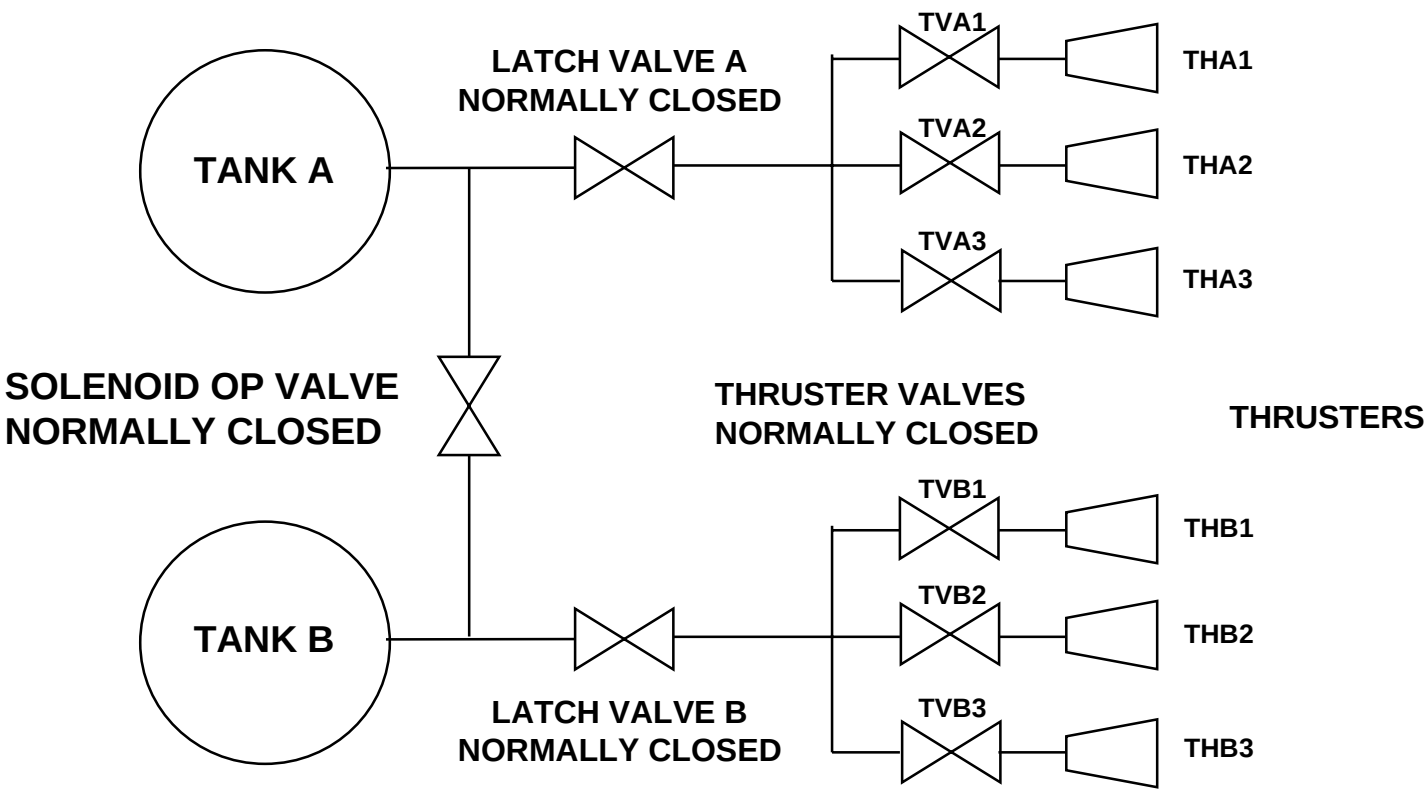
Why Not A Single Fault Tree or Event Tree?

- In practice, PRAs are modeled using both event trees and fault trees
- Event Trees are very efficient at sorting out the specific consequences of combinations of failures and successes of specific systems or operator actions
- Event Trees can lay out the Boolean logic necessary to perform probabilistic calculations (by defining in Boolean logic the combinations of successes and failures of systems and operator actions leading to specific outcomes)
- Fault Trees are very efficient at logically defining the specific combinations of component level failures which can lead to system failure

Why Not A Single Fault Tree or Event Tree? (Cont'd)

- Event Trees and Fault Trees both produce Boolean logic expressions essential for probabilistic quantification
  - Realize that once the Boolean expressions are developed, it is impossible to back out the sequence of events and model logic as it was originally developed
  - The purpose of performing Boolean logic reductions is only for model quantification
- Understand that although Reliability Block Diagrams, Event Trees and Fault Trees can produce similar cut set equations, they each have a place and are NOT interchangeable

Tutorial Fault Tree Example: Attitude & Orbit Control Subsystem



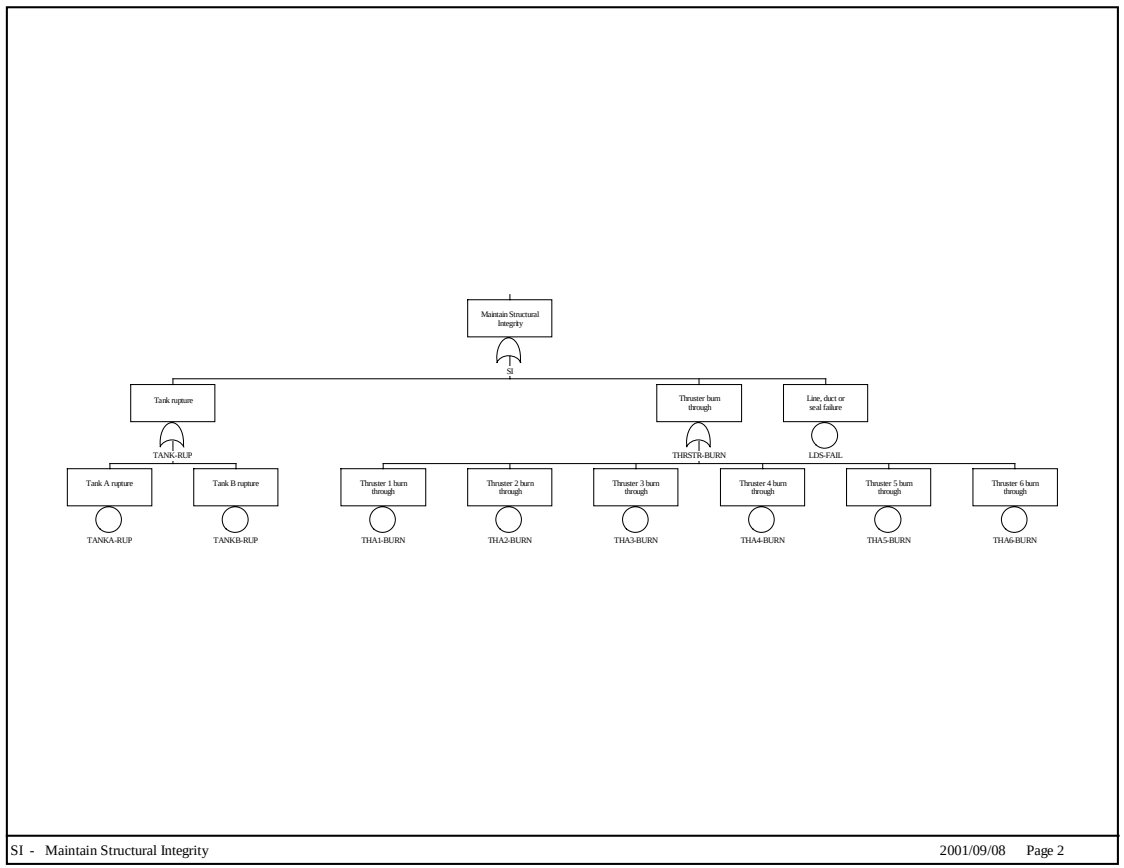
Tutorial Fault Tree Example: Attitude & Orbit Control Subsystem (con't)

- Consider the Attitude Control Subsystem only (ignore determination in this example)
- AOC functions
  - Maintain structural integrity
  - Maintain propellant load
  - Perform orbit maintenance maneuvers
  - Maintain center of mass functions

Tutorial Fault Tree Example: Attitude & Orbit Control Subsystem (con't)

- Tree logic: maintain structural integrity
  - SI OR
    - Tank A ruptures
    - Tank B ruptures
    - Line, duct or seal failure
    - Thruster burn through OR
      - THA1 fail
      - THA2 fail
      - THA3 fail
      - THB1 fail
      - THB2 fail
      - THB3 fail

Tutorial Fault Tree Example: Attitude & Orbit Control Subsystem (con't)

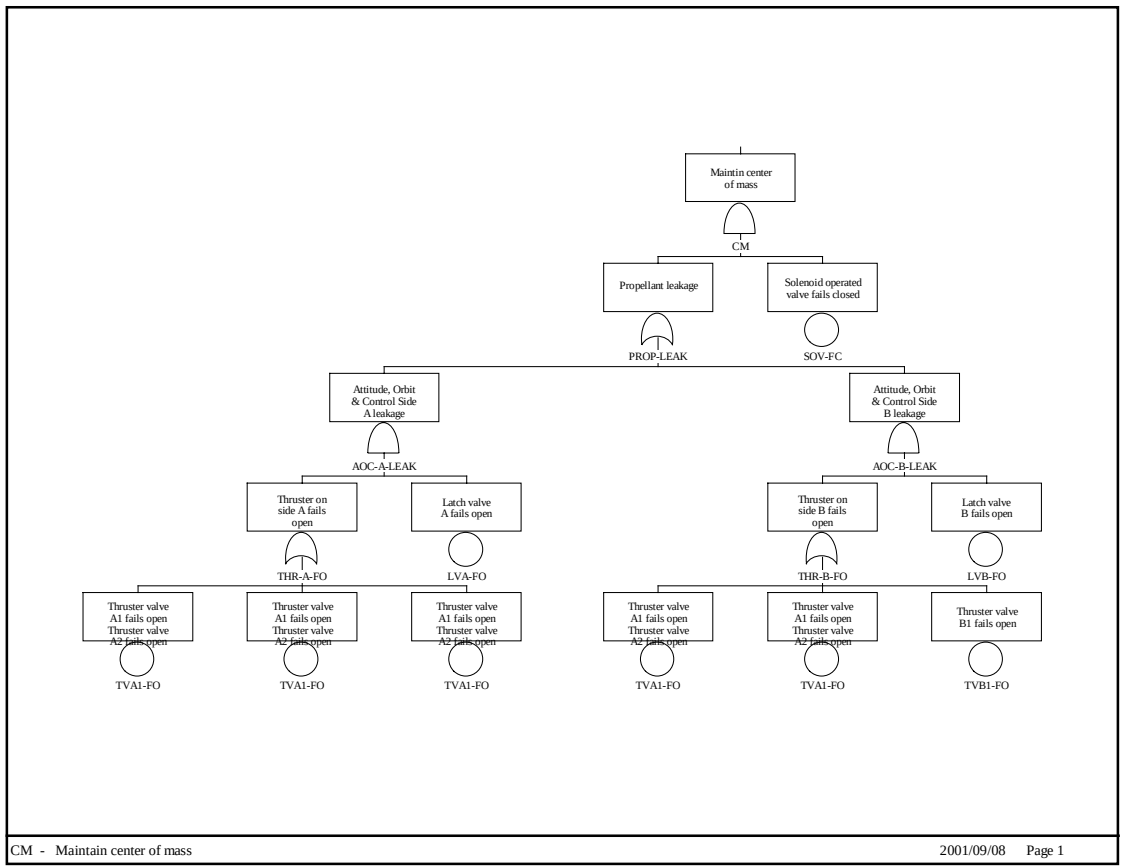


Tutorial Fault Tree Example: Attitude & Orbit Control Subsystem (con't)

Tree logic: maintain center of mass capability

- CM AND
  - SOV fail close
  - Propellant leakage on one side OR
    - Side A leakage AND
      - Latch valve A fails open
      - Thruster side A fails open OR
        - » TVA1 fails open
        - » TVA2 fails open
        - » TVA3 fails open
    - Side B leakage (Similar to Side A)

Tutorial Fault Tree Example: Attitude & Orbit Control Subsystem (con't)



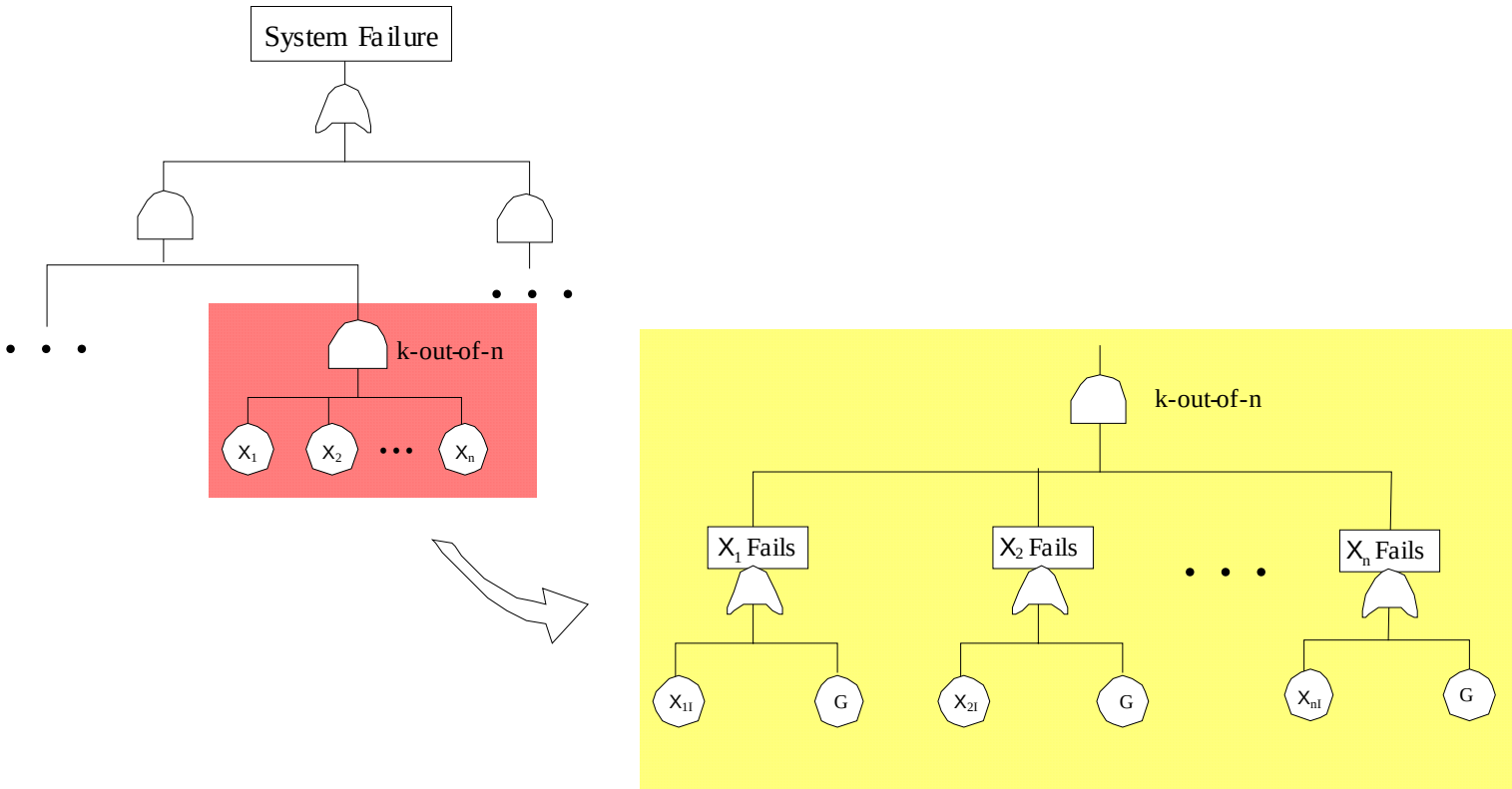
Common Cause Failures

- No PRA would be complete without including common cause failures
  - Intrinsic: Dependencies where the functional status of one component/element is affected by the functional status of another
  - Extrinsic: Dependencies where the couplings are not inherent and intended in the functional and physical design of the system

Common Cause Failures (con't)

- Extrinsic common cause failures are more difficult to grasp since they depend upon external factors
  - Environment
  - Handling
  - Maintenance
  - Design
  - Etc.

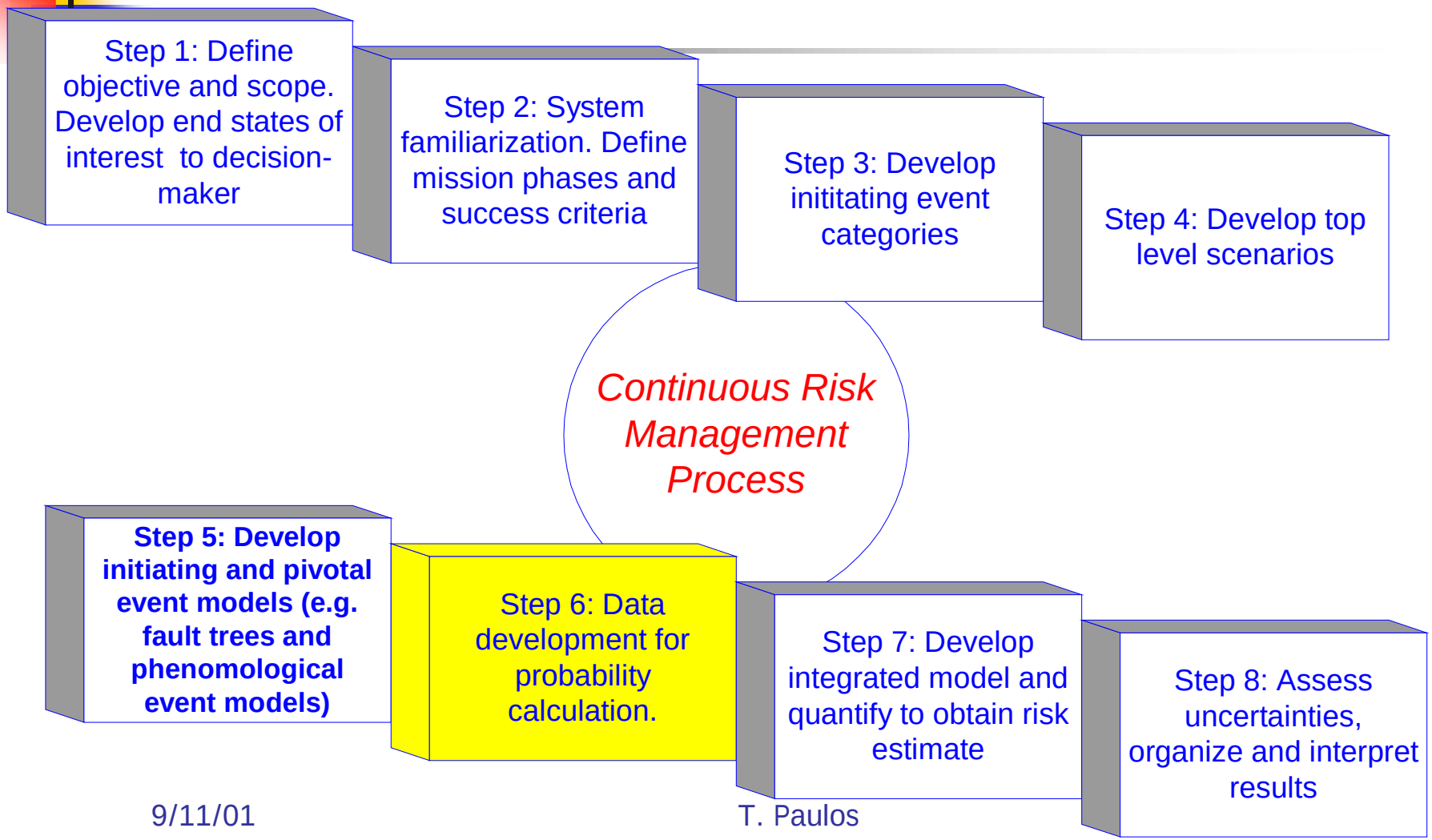
Common Cause Failures (con't)



Final Notes on Fault Trees

- Understand how the system is operated before you begin developing fault trees
- Understand ways in which the system can fail (some failures not always obvious)
- Use the Dependency Matrix to make sure that the cross system dependencies are taken into account
- Inclusion of common cause failures is IMPORTANT
- Are you really a fault tree expert?

Step 6: Data Development



Data Development

- PRA data analysis refers to the process of collecting and analyzing information in order to estimate various parameters of the PRA models
- Typical quantities of interest are:
  - Internal Initiating Events Frequencies
  - Component Failure Frequencies
  - Component Test and Maintenance Unavailability
  - Common Cause Failure Probabilities
  - Human Error Rates
  - Software Failure Probabilities

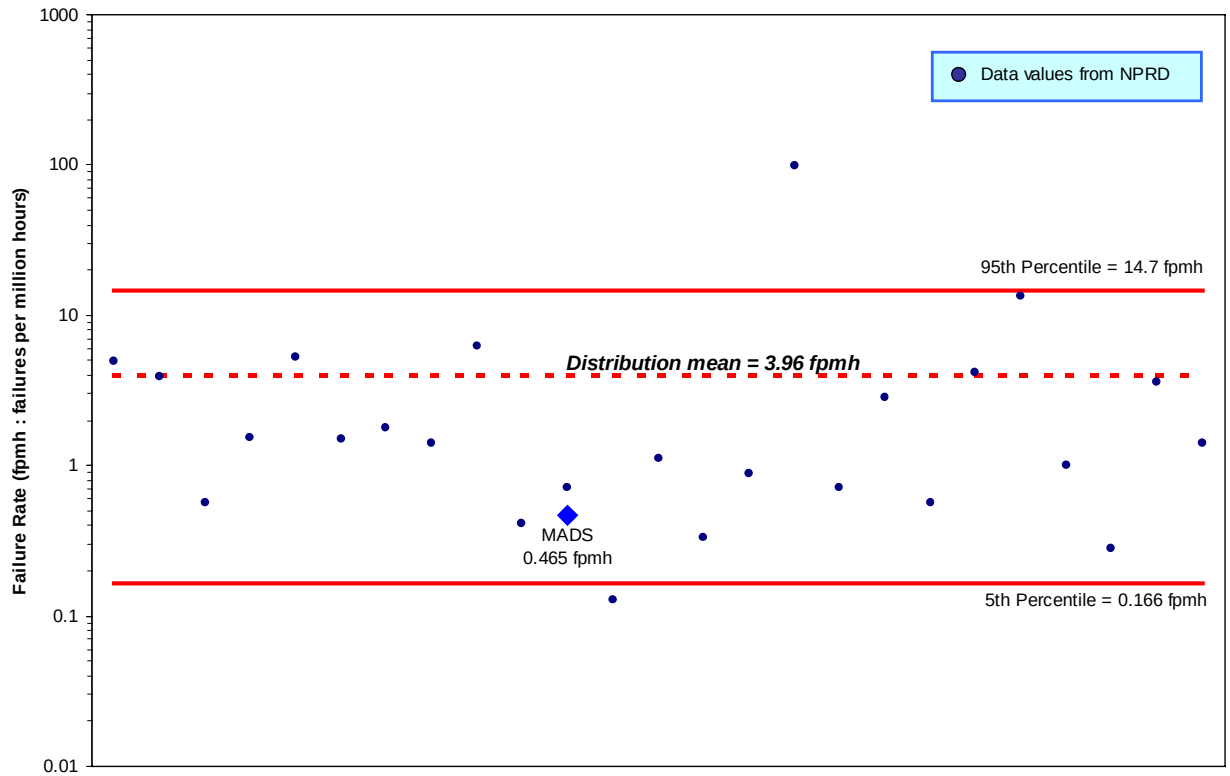
Data Sources

- Equipment Failure Rates
  - Modeling Analysis Data Sets (MADS)
  - Reliability & Maintainability reports
  - Non-electronic Parts Reliability Database 1995 (NPRD)
  - Electronic Parts Reliability Database 1997 (EPRD)
  - Failure Mode Distribution 1997 (FMD)

Data Sources (Cont'd)

  - Bellcore TR-332: Reliability Prediction Procedure for Electronic Equipment
  - Problem Reporting and Corrective Action (PRACA) Data System
- System-specific Information
  - Maintenance Logs
  - Test Logs
  - Operation Records
- Expert Elicitation

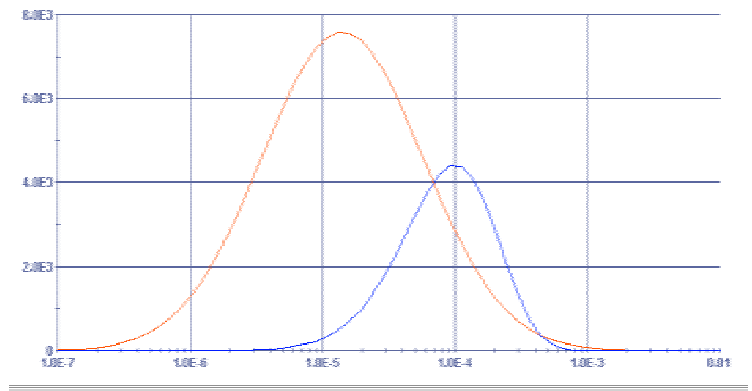
Example of Data Development



Bayesian Estimation (Continuous Random Variable)

- The *prior* probability distribution of a continuous unknown quantity, $Pr_o(x)$ can be updated to a incorporate new evidence E as follows:

$$Pr(x \mid E) = \frac{L(E \mid x)\, Pr_0(x)}{\int L(E \mid x)\, Pr_0(x)\, dx}$$



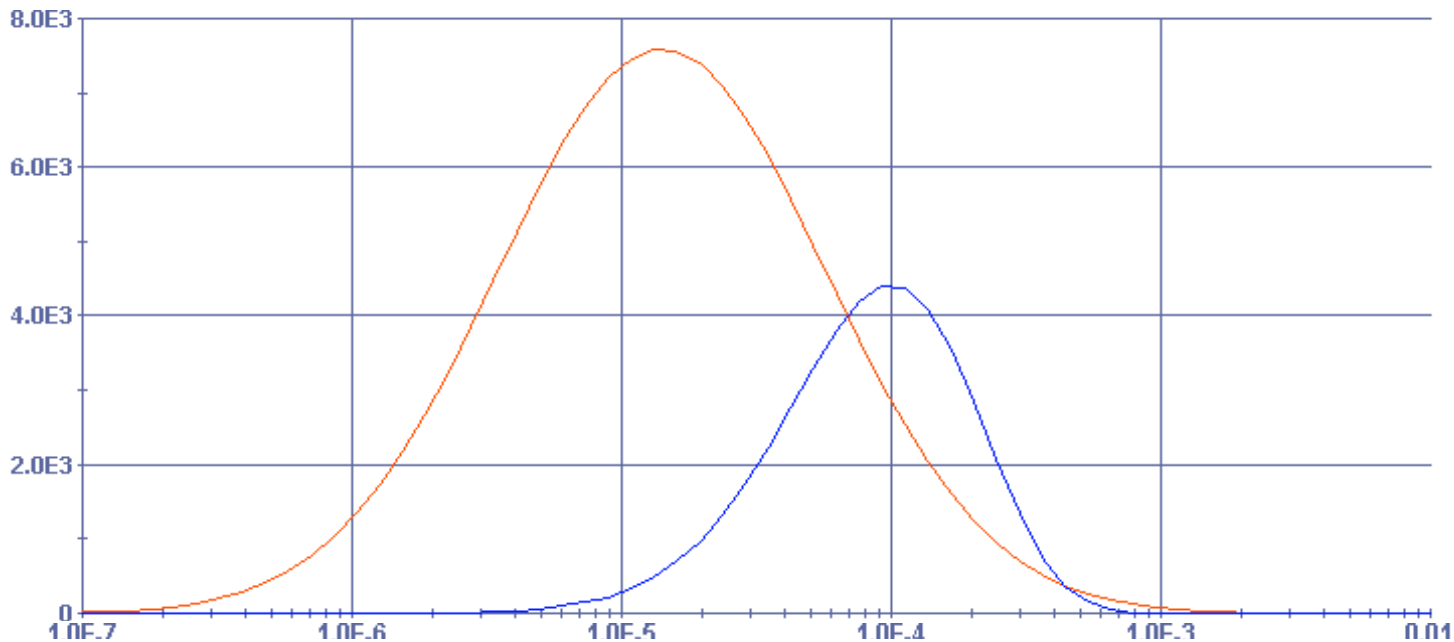
where

Pr(x|E) is the *posterior* or updated probability distribution of the unknown quantity X given evidence E (occurrence of event E),

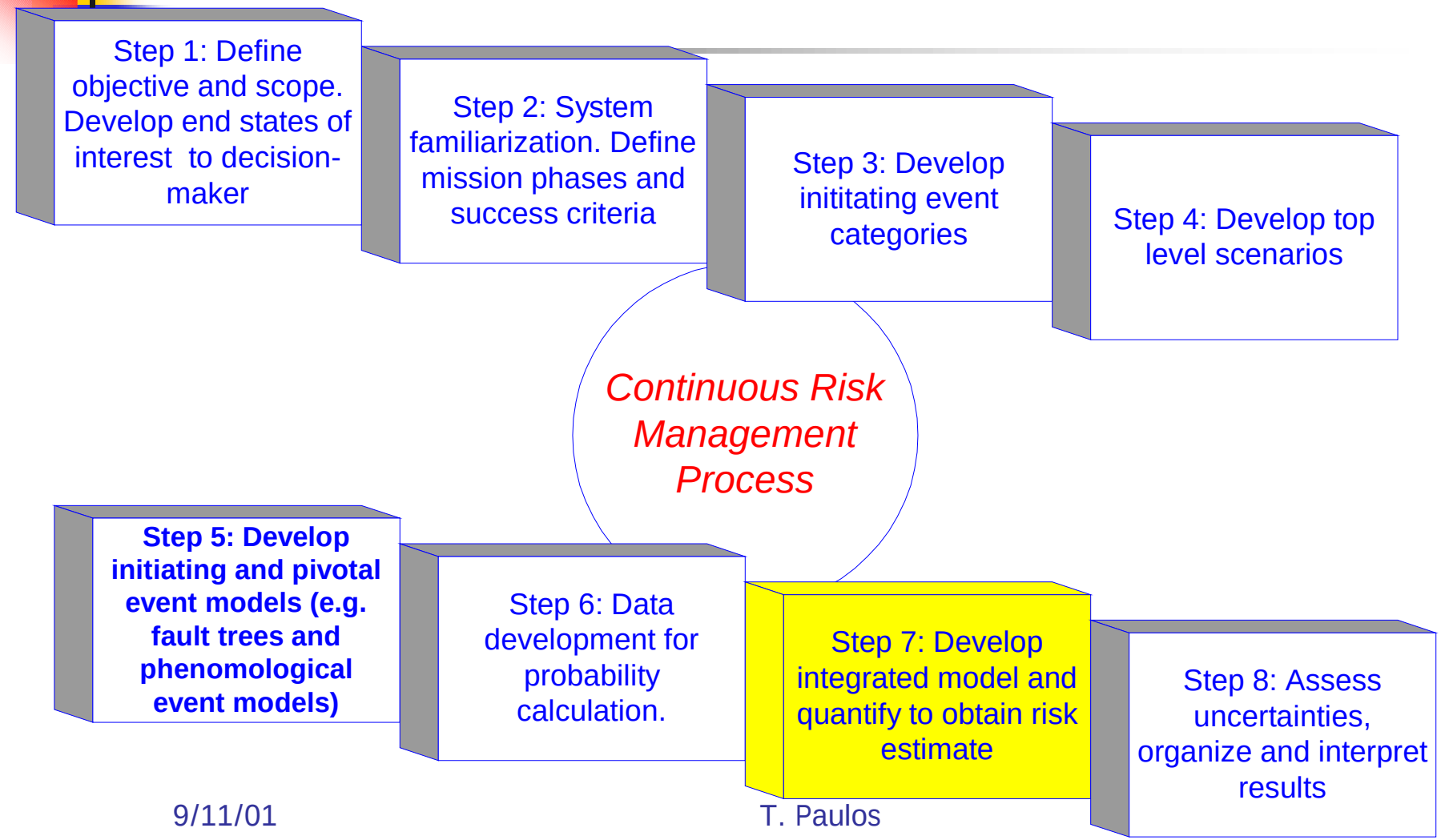
L(E|x) is the *likelihood* function, i.e., probability of the evidence E assuming the value of the unknown quantity is x,

Bayesian Updating Example

- Prior distribution
- Evidence
- Posterior distribution numerically



Step 7: Integrate and Quantify Risk Model



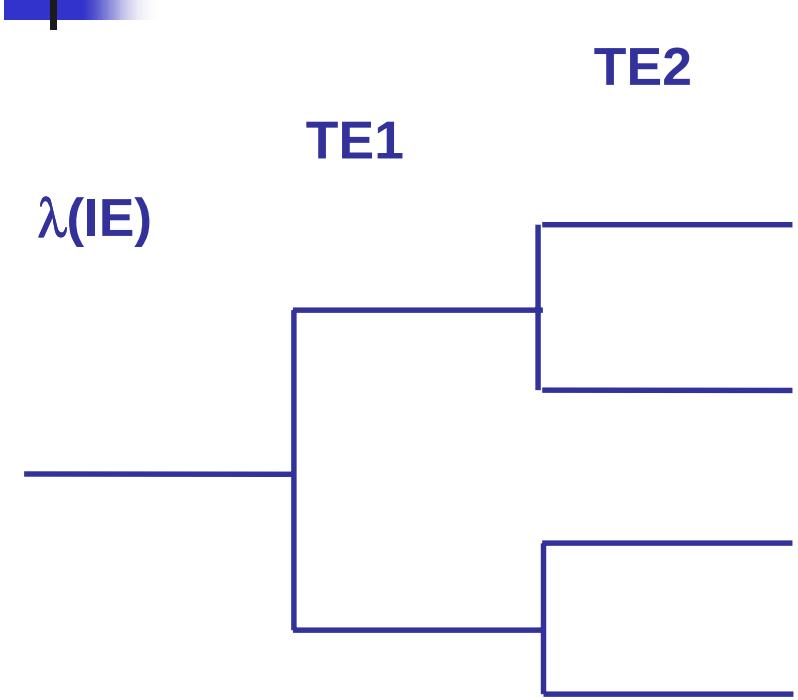
Quantification of Linked Event Tree/Fault Tree Models

- If only a limited scope reliability analysis of one system is being performed, numerical fault tree reduction is performed to obtain the minimal cut sets
  - Quantified as the sum of the probabilities of the individual cut sets

Quantification of Linked Event Tree/Fault Tree Models (con't)

- If a complete PRA quantification is desired, the minimal cut sets for individual top events (systems) are obtained numerically and stored
  - The event tree sequence logic is then used to define sequence cut sets by combining the logic expressions for the various top events according to the event tree logic
  - End state logic can be obtained from collecting the logic of all the sequences that end in that particular state

Model Integration



Sequence No.	Boolean Logic Expression
1	$\lambda(IE) \bullet \overline{TE1} \bullet \overline{TE2}$
2	$\lambda(IE) \bullet \overline{TE1} \bullet TE2$
3	$\lambda(IE) \bullet TE1 \bullet \overline{TE2}$
4	$\lambda(IE) \bullet TE1 \bullet TE2$

Quantification of Linked Event Tree/Fault Tree Models (Cont'd)

- Top Event Cut Set Logic:

  TE1 = A•B + A•C + A•D

  TE2 = K•A + K•D

- Boolean Expression for Sequence 4 would be:

  TE1•TE2 = A•B•K + A•B•K•D + A•C•K +
  A•C•K•D + A•D•K + A•D•K

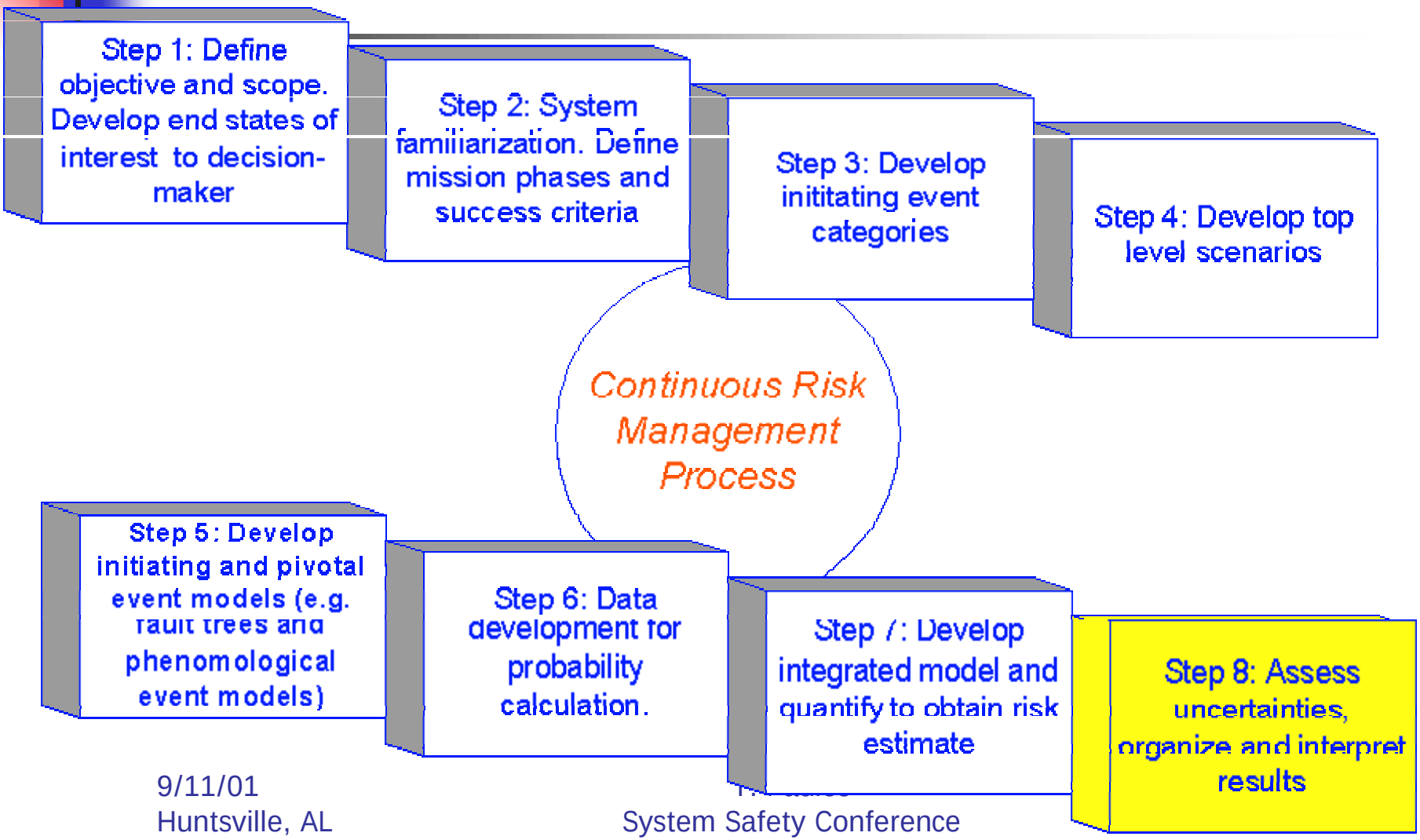
  TE1•TE2 = A•B•K + A•C•K + A•D•K

- Note that cut set A•D•K is generated twice but drops out because A•D•K + A•D•K = A•D•K
- Also note that A•B•K•D and A•C•K•D drop out (absorbed into A•D•K)

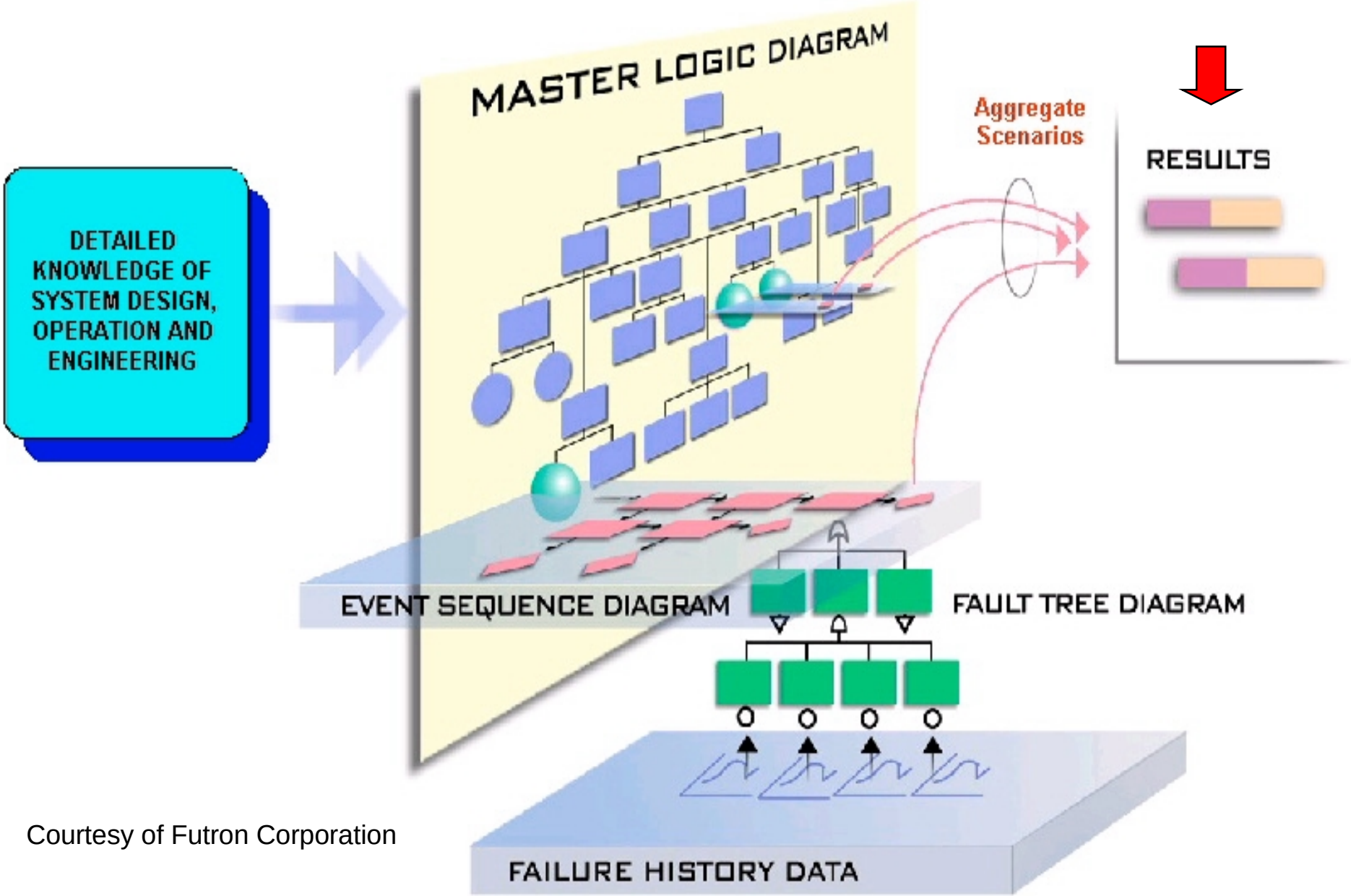
Final Notes on Model Integration

- System models can generate cut sets at various levels:
  - Fault Tree
  - Sequence
  - End State
- Not all cut sets are alike!
  - Minimal cut sets are generated for quantification purposes; once they are generated, the logic behind developing the models is lost

Step 8: Organize and Interpret Results

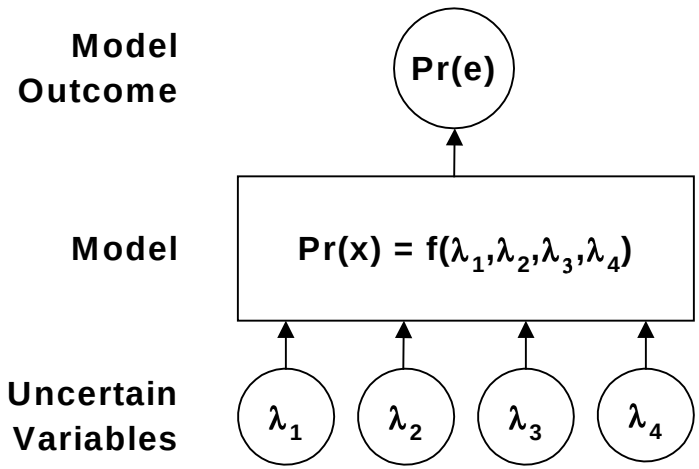


How Do Uncertainties Affect the Probabilities?



Courtesy of Futron Corporation

The Uncertainty Propagation Problem

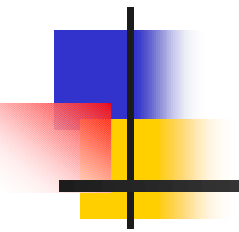


- In case the failure rates $\lambda_1,...,\lambda 4$ are known exactly, an exact value for Pr(e) can be easily obtained, given a deterministic model Pr(e) = $f(\lambda_1,...,\lambda_4)$.
- If the failure rates are uncertain, these uncertainties need to be accounted for when estimating Pr(e).
- Pr(e) itself will then also be uncertain.

Quick Note on Uncertainty

- Uncertainty calculations have become quite easy with the modern computer
  - Minimal cut sets generated by risk models
  - Data distribution inputs
  - Monte Carlo or LHS simulations at any level
    - Fault Tree
    - Sequence
    - End State

The Role of Traditional Reliability Engineering Analyses



What About FMEAs and Hazard Analyses?

- In a word, they are useful inputs
  - Each is incomplete with respect to PRA requirements
    - Lack dependencies
    - Lack multiple failures
    - Worst case consequences only
    - Do not obtain total probability of end states with uncertainties

What About FMEAs and Hazard Analyses? (con't)

- If already available, hazard analyses useful as input for identifying Initiating Events and Scenarios
- If already available, FMEAs are useful in checking fault tree basic events; interface FMEAs are useful in checking functions that need to occur for system success

What About FMEAs and Hazard Analyses? (con't)

- If FMEAs or Hazard Analyses are not available, a PRA can substitute for them
  - Scenarios identify hazards
  - Fault tree basic events identify failure modes
  - Identify system functions and/or scenario pivotal events

What About FMEAs and Hazard Analyses? (con't)

- The information is in a different form, but will be there if the analysis is complete
- Not always easy or possible to make a PRA look like another type of analysis

What About Fault Tree Analyses?

- PRAs are essentially linked Fault Trees
  - If appropriate, portions of a FTA can be used as part of the PRA
    - Difficult to break a single, mission fault tree into many different trees with different top events
    - Qualitative fault trees are much different than quantitative fault trees

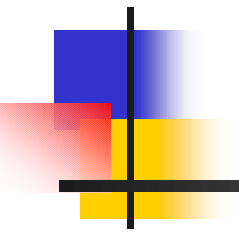
What About Fault Tree Analyses? (con't)

- Need to understand concepts of basic events, minimal cut sets, and quantification
- Need to understand that events in the event sequence are *conditional*
  - Fault trees do not show time or sequences

- The FTA supports the PRA, not vice versa

In Conclusion
(Yes, it is almost over …)

- The “P” is important, but do not over emphasize it
- The power of the PRA process lies in its ability to prioritize the risks, not in quantifying the bottom line number
  - Helps with risk management
  - The earlier these tasks are begun in the project, the better
- Understand the limitations of uncertainty
- PRA pays for itself many times over
  - Helps reduce costly redesigns (if possible)
  - How much did that vehicle cost anyway?



SAPHIRE Demonstration

We have time left?

QUESTIONS?????

IS ANY BODY AWAKE?