Adapting Traditional Logic Modeling Techniques to Address Cyberattack

R. Youngblood and K. LeBlanc

Outline

- Original Plan:
  - This project will (1) adapt an existing risk analysis method for application to the problem of identifying vulnerabilities to cyber manipulation in nuclear facilities, (2) demonstrate the modified method on a simple test facility (the Idaho State University, ISU, flow loop), and (3) apply the refined method on a larger scale to a realistic facility, the Human Systems and Simulation Laboratory (HSSL).
- Origins of Fault Tree Analysis
- Lapp-Powers Example
- Idaho State University flow loop
- Human Systems and Simulation Laboratory
- Summary

No Tips on Actual Sabotage Given in This Talk

Decision to Focus on Corruption of Information Flow

- Advantages:
  - It's a more general problem: it includes cyberattack, but is not restricted to cyberattack
  - In some ways, it's easier than first establishing each component's vulnerability to cyberattack, and only then reasoning about attacks that are capable of yielding a particular outcome
- Disadvantages:
  - If we do find an information corruption scenario that is worthy of prevention, we still have work to do to determine whether it is feasible for the attacker: we have to sort out whether there exists a real way to cause the scenario, cyberattack or otherwise

Origins of Fault Tree Analysis

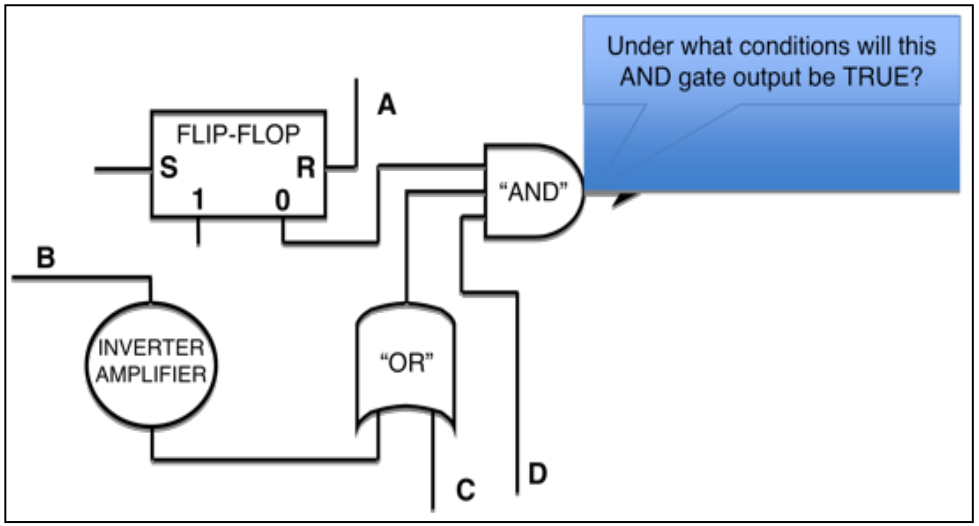
- Clif Ericson, "Fault Tree Analysis – A History," Proceedings of the 17th International System Safety Conference (1999).
  - FTA was conceived circa 1961, and as such is a relatively new tool compared to many other technical tools and disciplines. Special recognition should go to H. A. Watson as the Father of Fault Tree Analysis and Dave Haasl as the God Father [*sic*] of Fault Tree Analysis. Watson of Bell labs invented fault tree analysis (along with assistance from M. A. Mearns [*sic*]). Haasl, while at Boeing saw the benefits of FTA and spearheaded the first major application on the Minuteman program.
- Next several slides based on:
  - A. B. Mearns, "Fault Tree Analysis: the Study of Unlikely Events in Complex Systems," Talk Presented at the System Safety Symposium, Seattle, Washington, June 8-10, 1965.

Origins of Fault Tree Analysis (continued)

- At the time of Mearns' work, he and his coworkers were concerned with failures of "extremely complex" military control systems, including "failures" leading to "inadvertent triggering." Without "fault tree analysis," combining "logical design, symbolic logic, Boolean algebra, reliability analysis, and probability theory," the task of identifying components or combinations of components whose "failure or malfunction would be the most probable cause of [inadvertent triggering] appears to be a formidable task."
- Here, quotation marks denote terminology actually used by Mearns.
  - *It turns out that fault tree analysis was formulated in the first place to model corruption of information flow.*

Origins of Fault Tree Analysis: Inadvertent Triggering

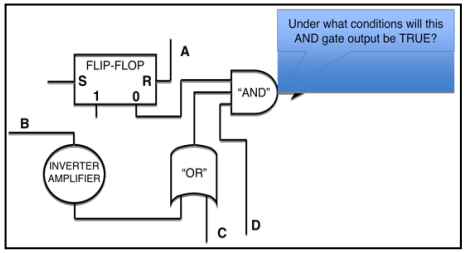
- The figure below shows a portion of a system that Mearns used to illustrate the idea. The event of interest is whether the output of the AND gate is TRUE (whether a triggering signal is present). By inspection, one sees that the design of the AND gate is such that its output will be TRUE by design if A*(/B+C)*D is TRUE. But failures in the electronics – the Flip-Flop, the Inverter, the OR gate, or the AND gate might also lead to this outcome, which is NOT intended by design. How to identify such possibilities?



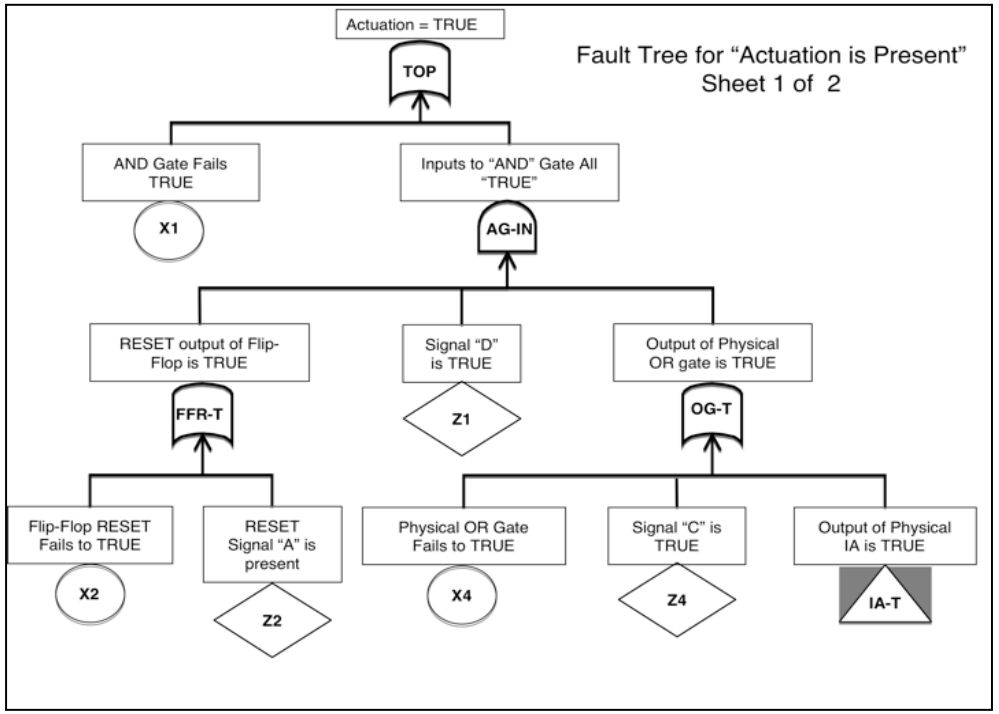
Identifying Failure Modes of a Physical Logic Circuit by developing and using…another logic circuit

- AND, OR, and NOT

Physical Logic Circuit



Logical Model of
Physical Logic Circuit



The Mearns paper then goes on to discuss most of the applications of FTA that we would cite today

- No mention of importance measures (they hadn't been defined yet)
- But from FTA, you get insight into what could cause failure

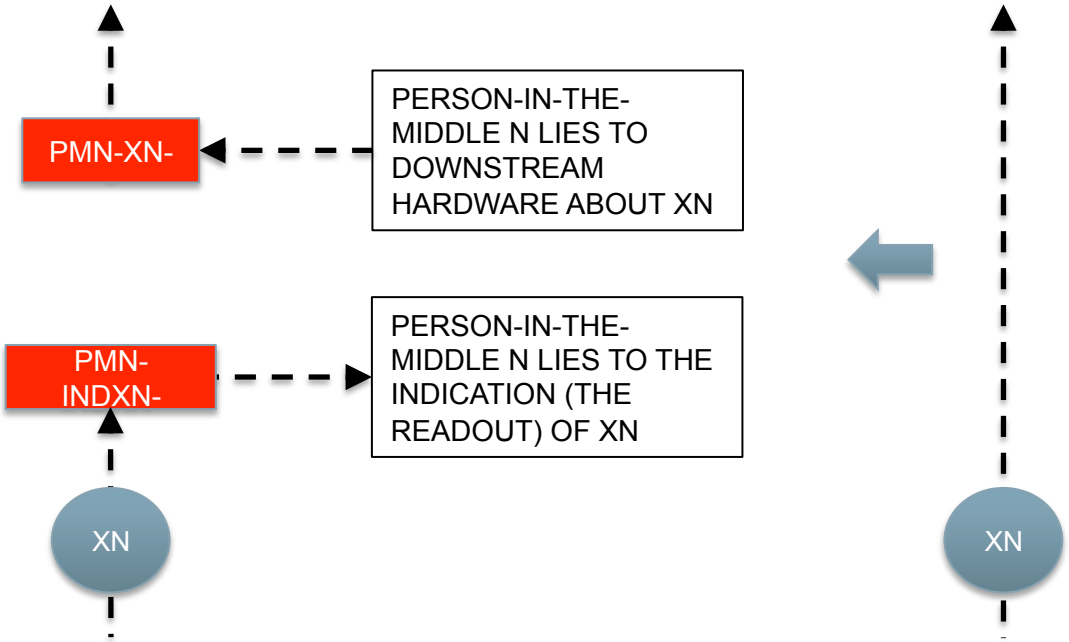
  Many of us learned about FTA in the context of learning about reactor systems, especially fluid systems. But FTA was first used to study the problem of corruption of information, leading to inadvertent triggering

- Back then, interestingly, complicated fault trees were "solved" by "simulation" (MC sampling over basic event probabilities to identify relatively dominant combinations of basic events)

Limitations of Fault-Tree Analysis, Even Then:

- This was a model of a logic network, and logic networks are *supposed* to work with binary variables, but that doesn't mean that there aren't actual failure modes that will yield in-between values of the voltages
  - Open circuits?
- Event Timing: When the order in which basic events occur makes a difference…
  - You can do a certain amount about the timing issues in basic event quantification, but one soon wishes for simulation

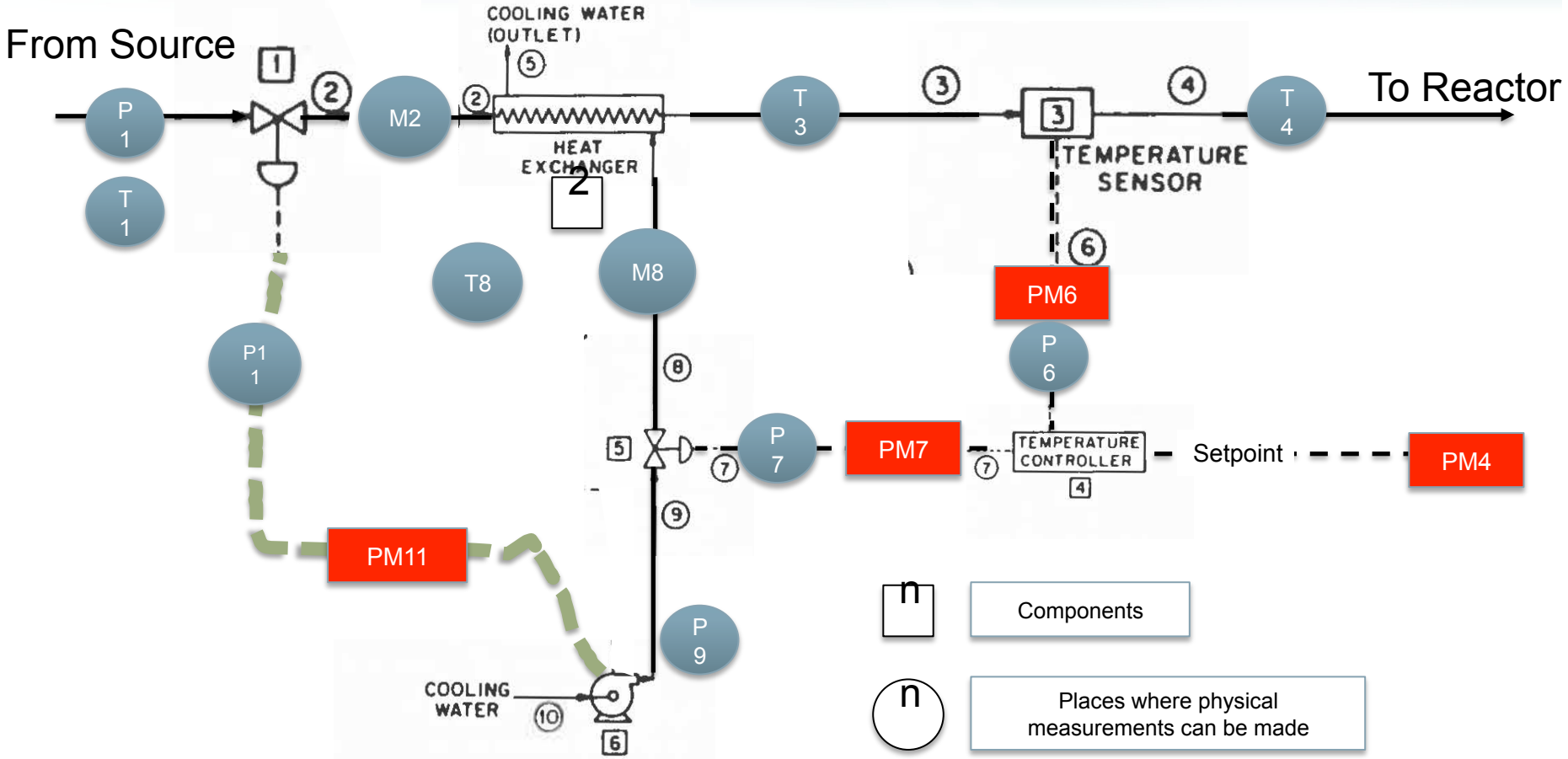
Valve F



<= Dotted Lines Represent Information Flow …

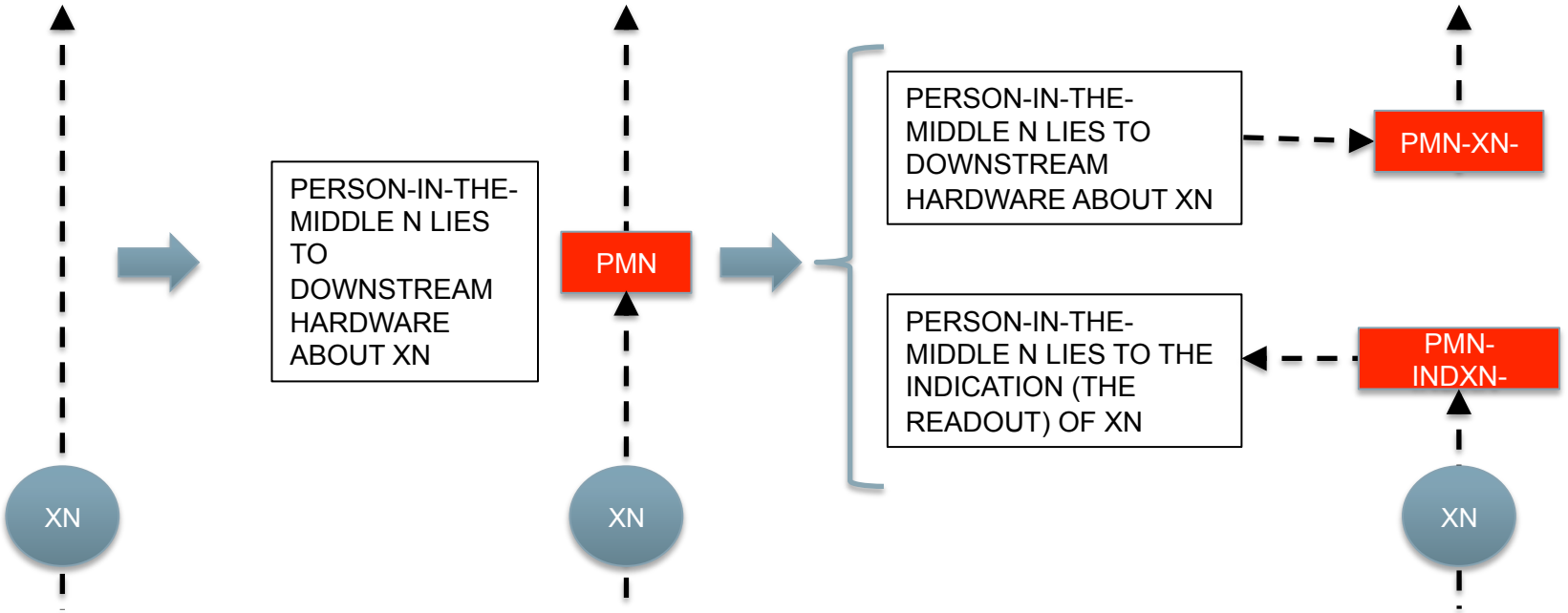
… which may be corruptible

Example Adapted from Lapp-Powers Paper



Dashed Lines Represent Information Flow

Put More Structure into "Person in the Middle"



EXAMPLE: TELL THE CONTROLLER THAT THE TEMP IS LOW (PM6-P6-T3-LOW); TELL THE INDICATION THAT THE TEMP IS NORMAL (PM6-INDP6-T3-NORM).

Interesting Sets of Conditions that Might Be Reachable by Corrupting Information

- System Failure
  - Corresponds approximately to a more usual PRA scope
  - Setting aside whether "usual PRA scope" focuses adequately on cyber-like failure modes: crippling one division in a 2-division system with a ½ success criterion doesn't emerge from such a scope
- Damage or Unavailability
  - You can fail a system completely by damaging it enough
  - But you can also
    - damage a system without completely failing it
    - cause unavailability

  So the "damage" analysis is not the same as the "failure" analysis
- **Corrupt indications to the operators so that**
  - **They cause damage**
  - They don't know that damage is being caused
  - Accomplishing this depends on knowing how damage is being caused <= Simulation (in the sense of simulating system physics)

An interesting one

- Combinations of these:
  - Achieve damage AND hide it from the operators
  - Cause failure AND hide it from the operators
  - Cause failure AND damage AND hide all of it from the operators

Conditions under which there is flow with T> Tmax, AND the indications available to operators are spoofed so as to conceal that fact. (1 of 2)

	Failure or Causal Spoof	Indication Spoof	Indication Spoof	Indication Spoof
P11-ON	**V5-INT**	**PM6-IND-T3-NORM**		
	(Failure) Coolant control valve just fails	PM indicates outlet temperature is normal		
P11-ON	**PM7-V5-CL**	**PM6-IND-T3-NORM**	**PM7-IND-P7-NORM**	
	(Spoof) PM tells coolant control valve to close	PM indicates outlet temperature is normal	PM tells the observer that the control valve is getting a “normal” signal	
P11-ON	**TC4-V5-CL**	**PM6-IND-T3-NORM**	**PM7-IND-P7-NORM**	
	(Failure) Temperature controller just tells control valve to close	PM indicates outlet temperature is normal	PM tells the observer that the control valve is getting a “normal” signal	

From Sol

Conditions under which there is flow with T> Tmax, AND the indications available to operators are spoofed so as to conceal that fact. (2 of 2)

	Failure or Causal Spoof	Indication Spoof	Indication Spoof	Indication Spoof
P11-ON	**PM6-TC4-LO-T3**	**PM6-IND-T3-NORM**	**PM7-IND-P7-NORM**	
	(Spoof) PM tells the controller that outlet temperature is LOW	PM indicates outlet temperature is normal	PM tells the observer that the control valve is getting a “normal” signal	
P11-ON	**SENS-3-LOW-T3**	**PM6-IND-T3-NORM**	**PM7-IND-P7-NORM**	
	(Failure) The Sensor reads a low outlet temperature	PM indicates outlet temperature is normal	PM tells the observer that the control valve is getting a “normal” signal	
P11-ON	**PM4-TC4-STPT-HI**	**PM4-IND-TC4-NORM**	**PM6-IND-T3-NORM**	**PM7-IND-P7-NORM**
	(Spoof) PM feeds the temperature controller a high setpoint	PM indication to the observer is a normal setpoint	PM indicates outlet temperature is normal	PM tells the observer that the control valve is getting a “normal” signal

Single Top Event versus Multiple Top Events

Idaho National Laboratory

- In many applications of logic modeling, there is one top event, often related to high consequences
  - Inadvertent triggering of missile launch
  - Inadvertent detonation of nuclear weapon
  - Damage to a nuclear reactor core
  - Large Hydrocarbon Release
  - …
- In those facilities, specific functions are designed to prevent those specific consequence types
- In analyzing those facilities, it's natural to build a logic model top down, focusing on possible causes of the specific events of concern, addressing failures of the specific functions
- However, consider how to identify ways of damaging or inconveniencing the facility
- For a large, complex facility, there are potentially many ways to damage or inconvenience facilities, involving different mechanisms operating at different places on different components, having different types of effects.
- Formally, we can imagine an OR gate at the top, with many, many inputs corresponding to different ways of causing damage. That's a very flat model.
- This seems like a formidable scope problem …

Summary

- Thinking in terms of "corruption of information flow" seems like a good idea
- The HSSL is a key tool for us. It IS a plant simulator, and we can test hypotheses about plant responses to corrupt inputs
- We *have* found ways to adversely affect plants by corrupting information with fairly minimal effort, though we have not explicitly mapped those ways into feasible cyberattacks
- Logic modeling has the usual pluses and minuses …
  - One plus is that given a logic model, one can do Top Event Prevention Analysis, which often improves insight in several ways
  - Minuses:
    - Simulation is needed
      - In order to address timing
      - In order for the analyst to be sure that assumptions about plant response to a given attack are correct
      - In order to determine system response to gray-area (not 1/0) signals
- ... With the additional practical "minus" that for "damage" or "inconvenience" - as opposed to "outright system failure" - a *logic* model of a complex facility may be very flat
- We need something like an information-oriented variant of HAZOP

References

- Clif Ericson, “Fault Tree Analysis – A History,“ Proceedings of the 17th International System Safety Conference (1999).
- A. B. Mearns, “Fault Tree Analysis: the Study of Unlikely Events in Complex Systems,” Talk Presented at the System Safety Symposium, Seattle, Washington, June 8-10, 1965.
- Steven A. Lapp and Gary J. Powers, Computer-aided Synthesis of Fault-trees, IEEE Transactions on Reliability, April 1977.
- Lapp S. A. and G. J. Powers, Computer-Aided Synthesis of Fault Tree, IEEE Transactions on Reliability, Vol. R-26, No. 1, pp. 2-12, Apr. 1977.
- R. Youngblood and L. Oliveira, "Application of an Allocation Methodology," Proceedings of "PSA '89 / International Topical Meeting / Probability, Reliability, and Safety Assessment," April 2-7, 1989, Pittsburgh, Pennsylvania (American Nuclear Society, Inc., La Grange Park, Illinois, 1989).
- R. W. Youngblood and R. B. Worrell, "Top Event Prevention in Complex Systems," Proceedings of the 1995 Joint ASME/JSME Pressure Vessels and Piping Conference, PVP-Vol. 296, SERA-Vol. 3, "Risk and Safety Assessments: Where Is The Balance?” July 1995 (The American Society of Mechanical Engineers, New York, New York 10017, 1995).
- D. D. Boozer and R.B. Worrell, 'A Method for Determining the Susceptibility of a Facility to Sensor System Nullification by Insiders,' SAND77-1916C (February 1978).
- Risk-Informed Safety Margin Characterization Case Study: Use of Prevention Analysis in the Selection of Electrical Equipment to Be Subjected to Environmental Qualification, D. P. Blanchard and R. W. Youngblood, Proceedings of PSAM-12 (Probabilistic Safety Assessment and Management), 22-27 June 2014.
- R. B. Worrell and D. P. Blanchard, “Top Event Prevention Analysis to Eliminate Requirements Marginal to Safety,” Proceedings of the 1995 Joint ASME/JSME Pressure Vessels and Piping Conference, June 1995.
- Yadav, V., Youngblood, R. W., Le Blanc, K. L., Perschon, J., Pitcher, R., “Fault-Tree Based Prevention Analysis of Cyber-Attack Scenarios for PRA Applications,” In Proceedings of the Annual Reliability and Maintainability Symposium (RAMS) (IEEE, 2019).