Modeling and Analyzing Faults to Improve Election Process Robustness

EVT/WOTE 2010
Washington, D.C.
August 9, 2010

Borislava I. Simidchieva (UMass Amherst),
Sophie J. Engle (UC Davis),
Michael Clifford (UC Davis),
Alicia Clay Jones (Booz Allen),
Sean Peisert (UC Davis, LBNL),
Matt Bishop (UC Davis),
Lori A. Clarke (UMass Amherst), and
Leon J. Osterweil (UMass Amherst)

UMASS AMHERST | UC DAVIS UNIVERSITY OF CALIFORNIA

Motivation

- Elections are more than machines
  - A *process*
- Problems arise in the process
  - Sometimes manifest as machine problems
  - Sometimes not . . .
- Plans for known and anticipated problems
  - But unexpected problems still arise

Example Problem

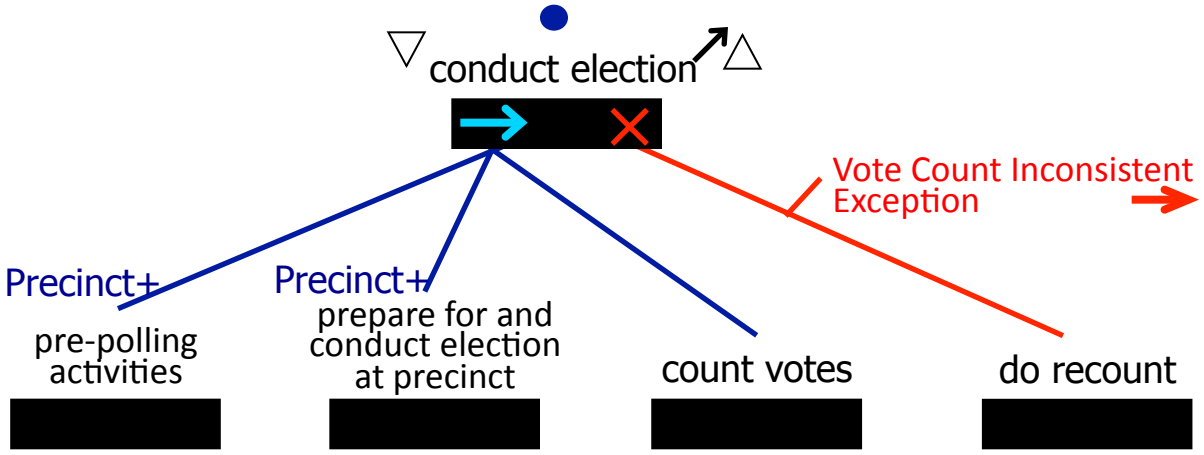
- Election procedures for validating number of ballots
  - Count them at polling station
  - Count them at Election Central
  - A discrepancy: the two ballot counts are different, or the vote counts disagree with the ballot counts
  - *What happened?*

Our Approach: *Continuous Process Improvement*

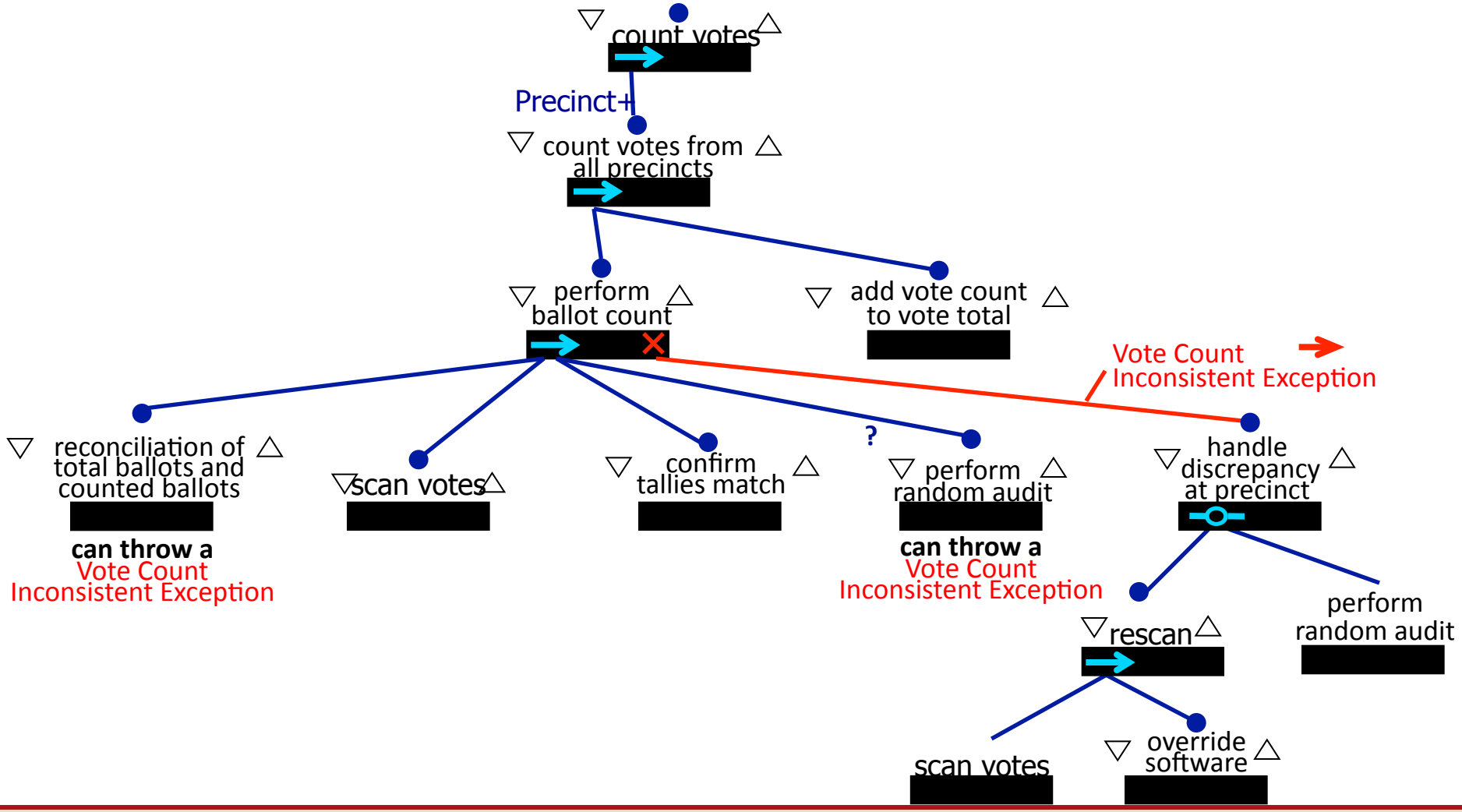
- Create a precise, accurate model of the real-world election process
- Use formal analysis methods to automatically identify potential problems in the model
  - Here, we focus on *single points of failure (SPFs)*
- Modify process model to ameliorate problems
  - Verify the modification makes things better
- Deploy improvements in real-world process
- Repeat

Election Process in Little-JIL

- Graphical process definition language with formal semantics; process represented as a hierarchical decomposition of steps



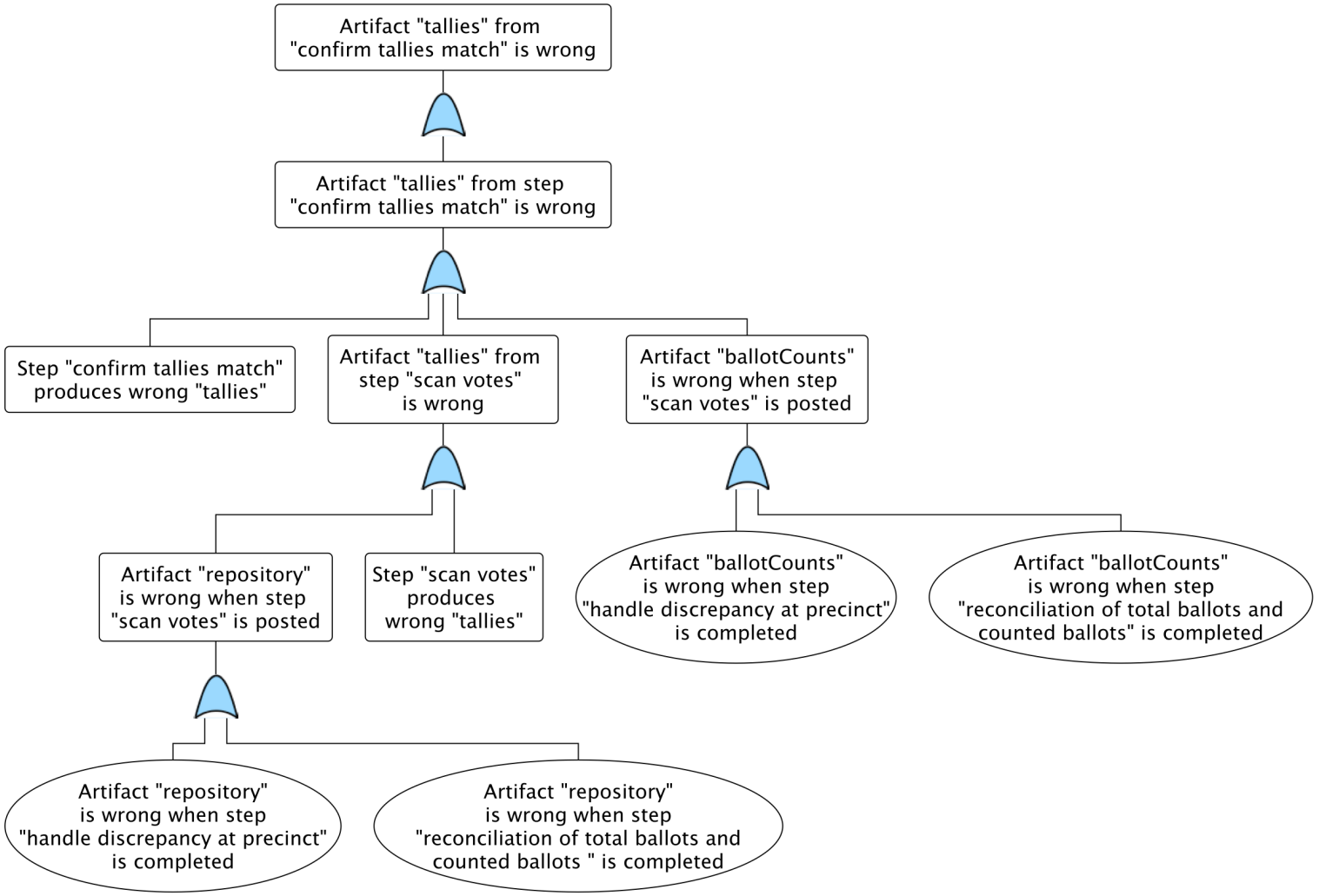
Election Process in Little-JIL (2)



Fault Tree Analysis (FTA)

- Fault trees show how problems could arise
  - Like attack trees but intent is irrelevant
- FTA can automatically generate fault trees from Little-JIL process model and a hazard
- Single Points of Failure (SPFs) can be automatically identified from fault trees

Fault Tree Generated from Model



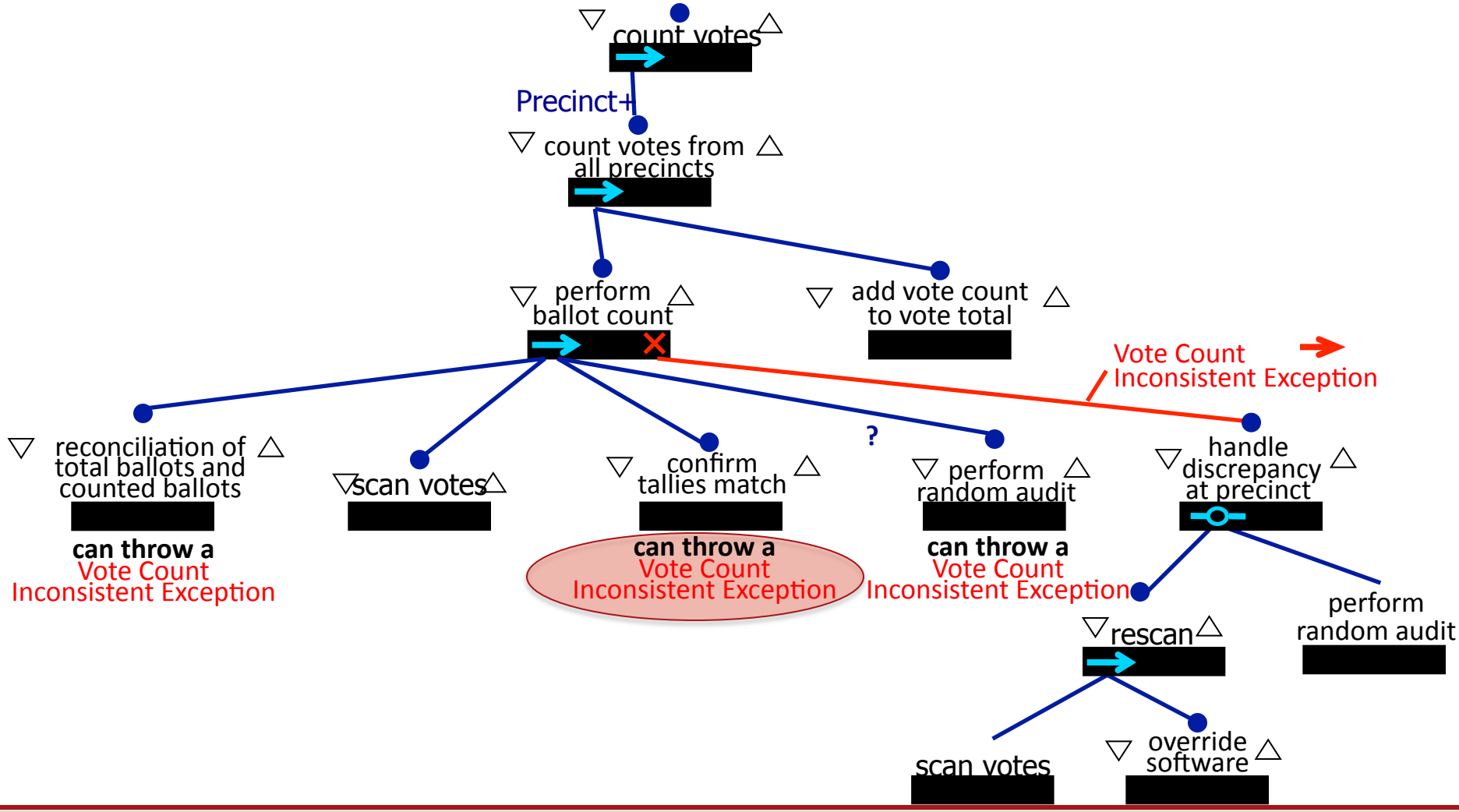
Cut Sets Computed from Fault Tree

- Combination of events such that, if all events in the cut set occur, the hazard occurs
  - *Minimal* if removal of any event causes the resulting set not to be a cut set
- Can be computed automatically from the fault tree

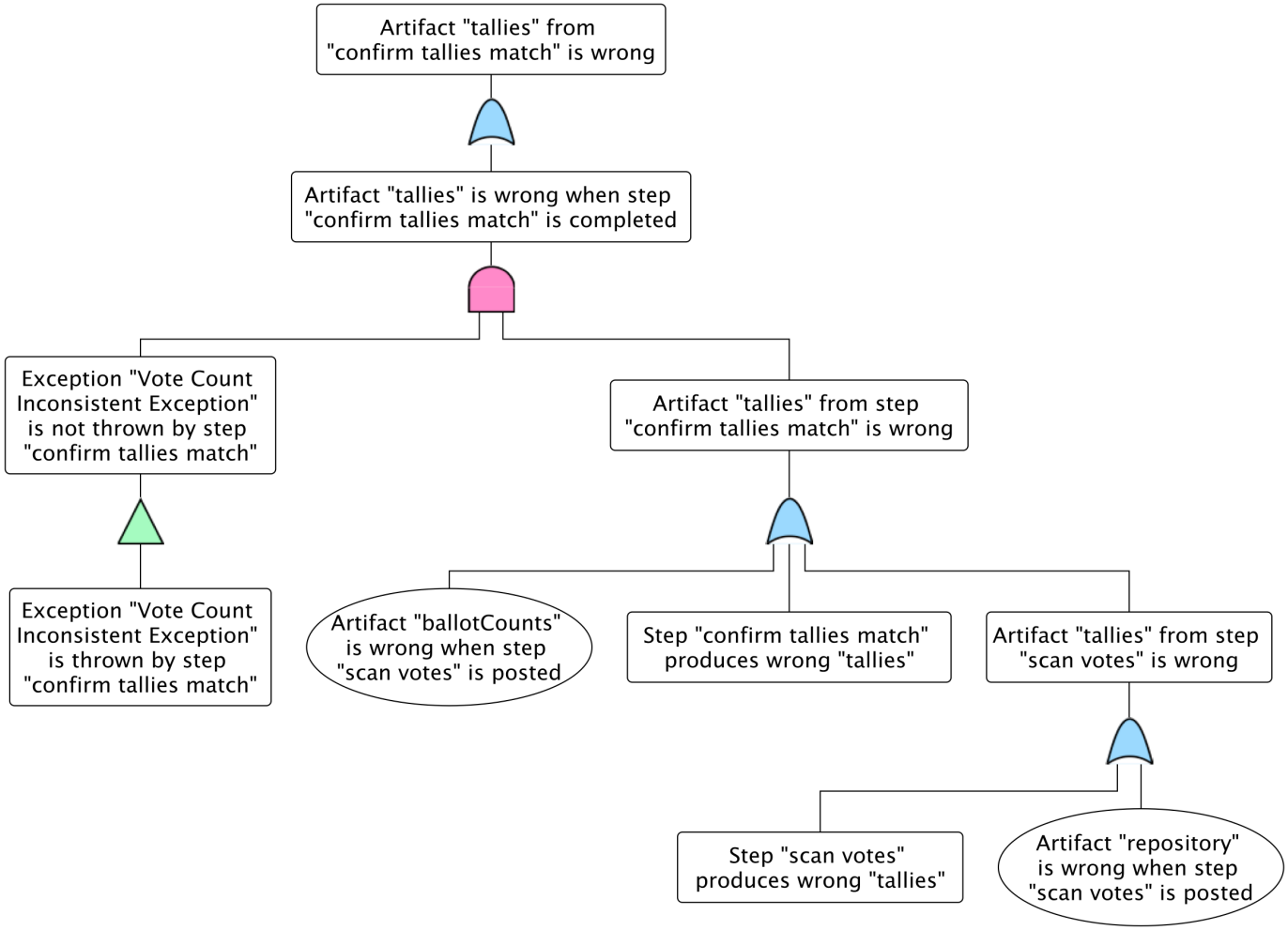
Our Original Process Model MCSs

- MCS #1 (SPF): Step **`scan votes`** produces **`wrong tallies`**
- MCS #2 (SPF): Step **`confirm tallies match`** produces **`wrong tallies`**
- Total 16 MCSs
  - 10 of size 2 or less

Add Exception Declaration to Model



And the Resulting Fault Tree



Our Revised Process Model MCSs

- MCS #1′: Step **`scan votes`** produces **`wrong tallies`**; **`Vote Count Inconsistent Exception`** is NOT thrown by **`step confirm tallies match`**
- MCS #2′: Step **`confirm tallies match`** produces **`wrong tallies`**; **`Vote Count Inconsistent Exception`** is NOT thrown by **`step confirm tallies match`**
- Total 16 MCSs (same as before)
  - Only 2 of size 2 or less (compared to 10 before), no SPFs

General Thoughts

- Yolo County, CA, election process modeled
  - Should work similarly for other jurisdictions
- Using fault tree analysis seems effective
  - Automatic generation of fault trees a *big* plus!
- One model covers many hazards

Conclusion

- *Continuous Process Improvement* can be successfully applied to elections
- Defects in the model can guide improvements in the real-world process
- Modifications can be evaluated in advance through formal analysis

Future Work

- Apply other forms of analysis such as Failure Mode and Effects Analysis (FMEA)
- Apply to other jurisdictions' processes
- Derive requirements for components used in the process - specifically, e-voting components
- Work with election officials to translate results into something they can use directly, i.e. without us!

Related Work

- Direct Recording Electronic (DRE) machines
  - Research: Compuware; UConn VoTeR Center; ACCURATE; Brennan Center for Justice; RABA; EVEREST; Caltech/MIT Voting Technology Project; Proebstel et al; Yasinsac et al
  - Statewide reports: CA, MD, OH, …
- Verification of Elections
  - Mercuri & Neumann; Saltman
- Requirements for elections
  - Mitrou; Lambrinoudakis et at

Related Work (continued)

- Election Process Modeling
  - Election Assessment Hearing; Raunak et al; Simidchieva et al; Curtis et al; Antonyan et al; Hall et al
- Fault Tree Analysis
  - Helmer et al; Zhang et al; Rushdi, Ba-Rukab; Yee; Peisert et al; Nai Fovino et al

Thanks!

- Artifacts and full fault trees available at **http://laser.cs.umass.edu/elections/**
- Thanks to NSF for sponsoring work
  - Especially grant CCF-0905530; any opinions, etc. are ours, and may or may not be those of NSF
- Thanks to Yolo County, CA election officials, especially Tom Stanionis and Freddie Oakley