Definition and Analysis of Election Processes

M. S. Raunak, B. Chen, A. Elssamadisy,
L. A. Clarke and L. J. Osterweil
University of Massachusetts, Amherst

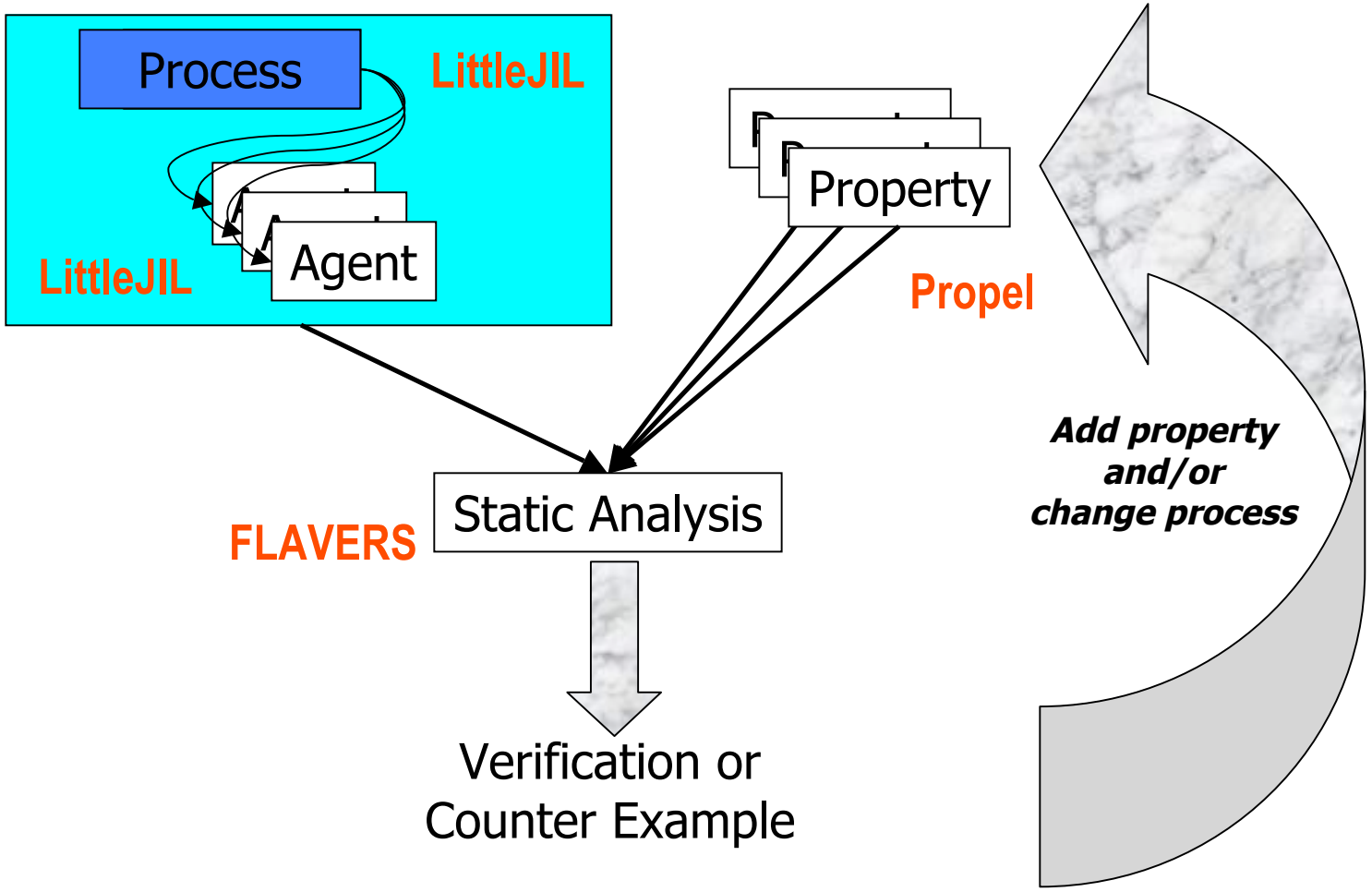
Importance of Election Processes

- Election is the basis of democracy
- Some recent elections have been quite controversial
  - Thai election, Apr 2006
  - Azerbaijan election, 2005
  - Ukraine presidential election, 2004
  - US presidential elections
    - Ohio in 2004, Florida in 2000
- Related work and our focus

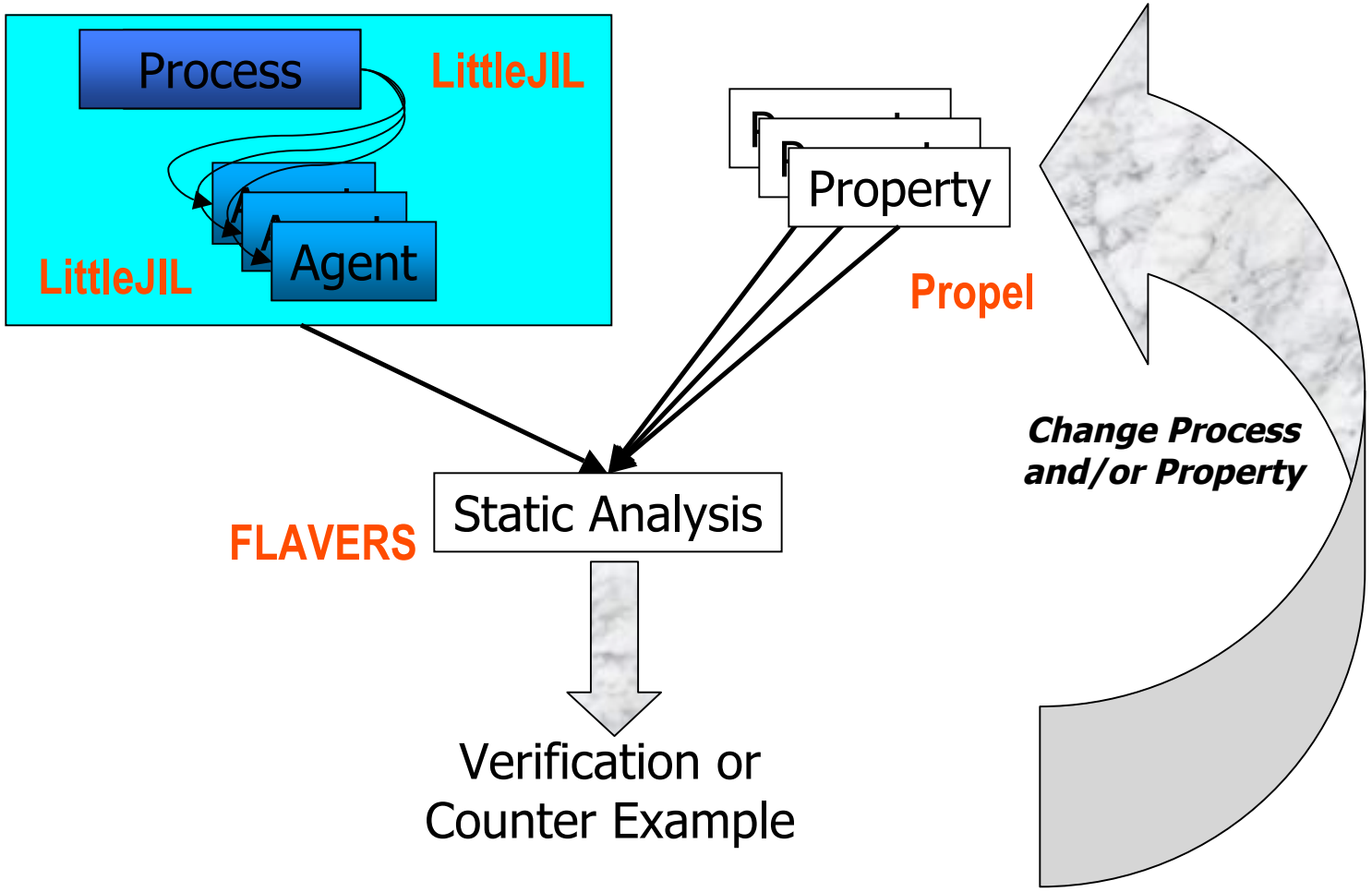
Relevance to Software Processes

- Election is an important *process* with many agents and complex details
- Process analysis techniques can be used to identify vulnerabilities like:
  - Process errors
  - Security violations due to mistakes, fraud, collusion etc.
- Demonstrates an important application of software process improvement to another domain

Our Approach and Tools



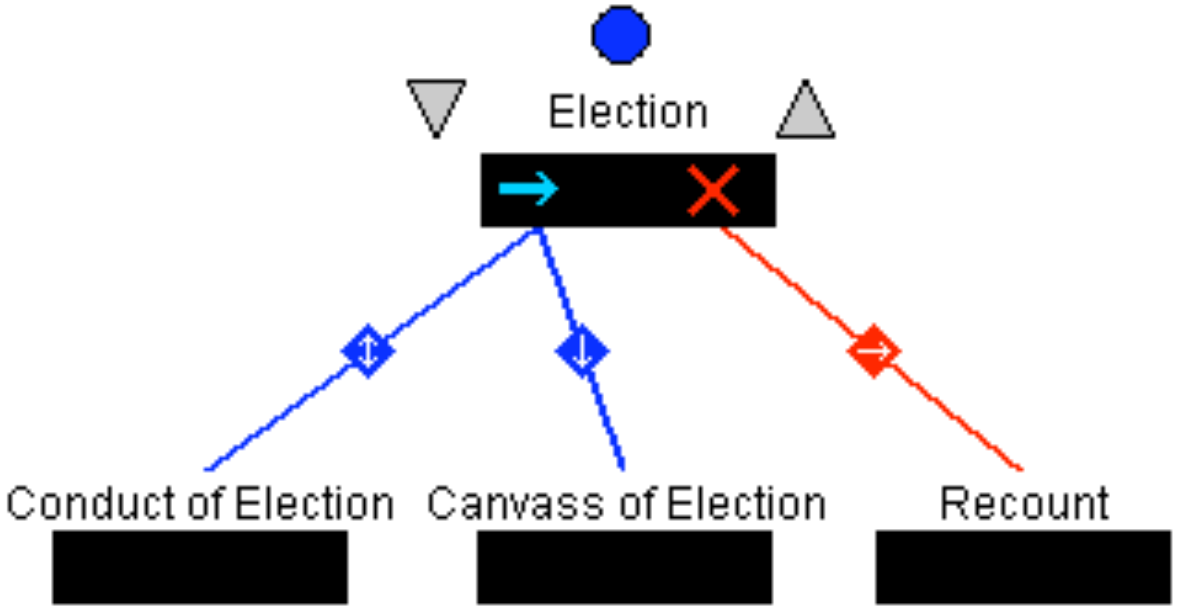
Our Approach (cont..)



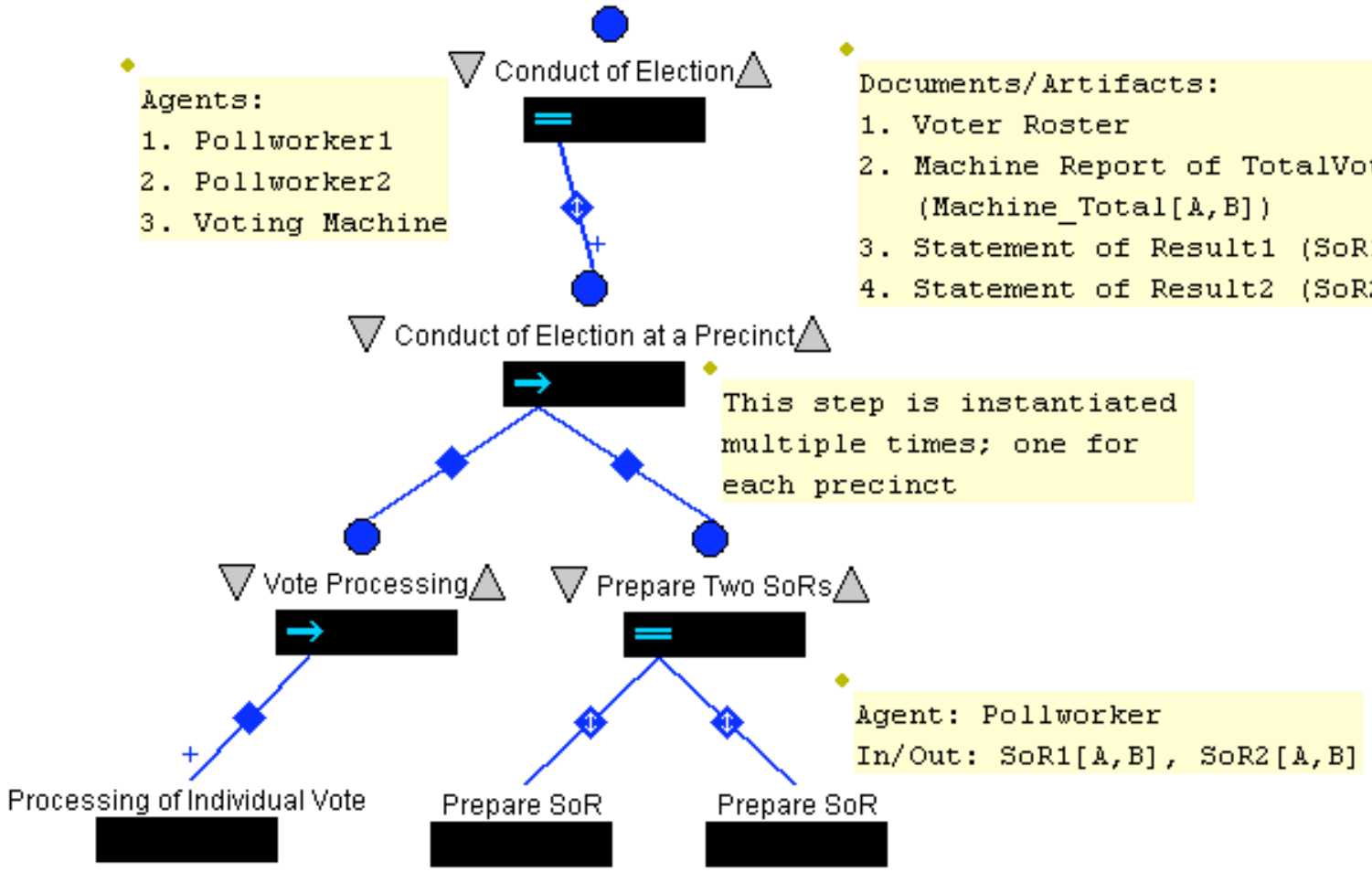
The Election Process We Studied

- A simplified election process
  - One DRE machine every precinct
  - One position up for election
  - Two candidates (A and B)
- Creation of 'Statement of Results'
  - Two copies of SoR by two poll workers
- State level aggregation
  - Validation of precinct level reporting
  - Creation of statewide summary

The Election Process in LittleJIL

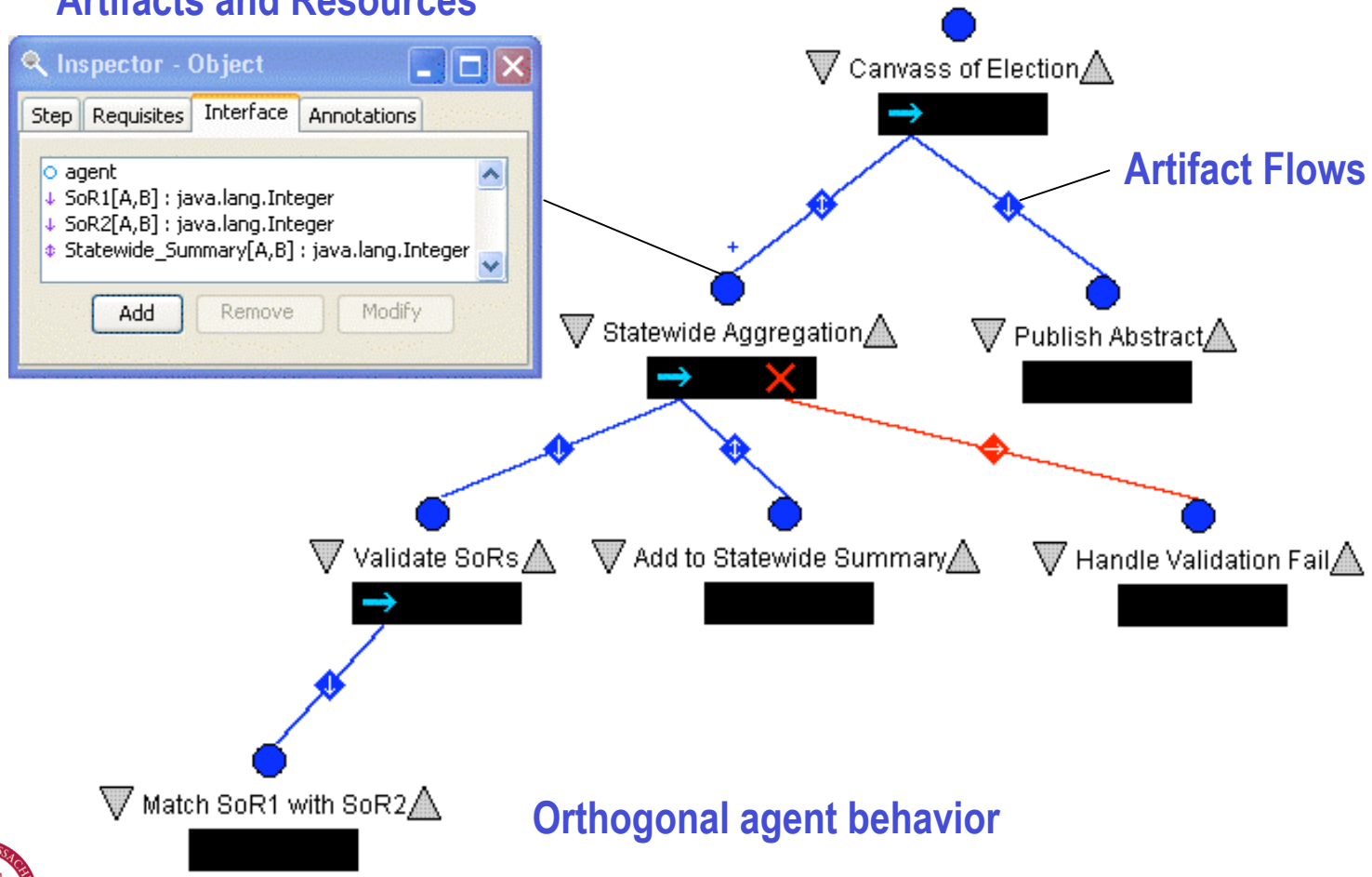


Conduct of Election Process



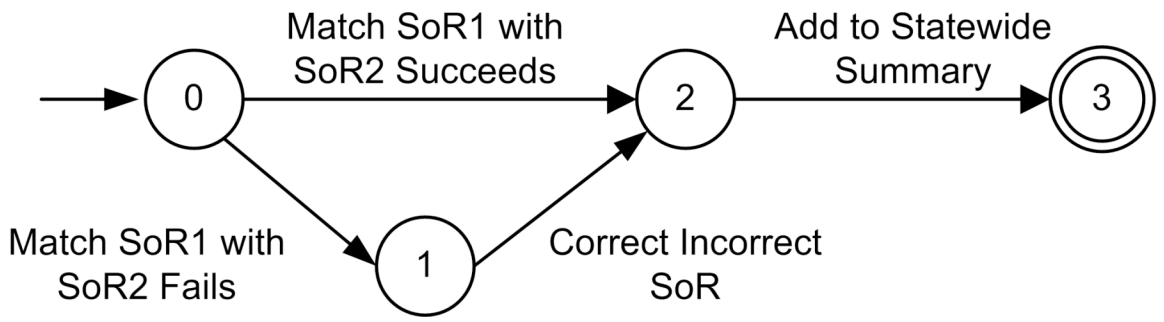
Canvass of Election Process

Artifacts and Resources

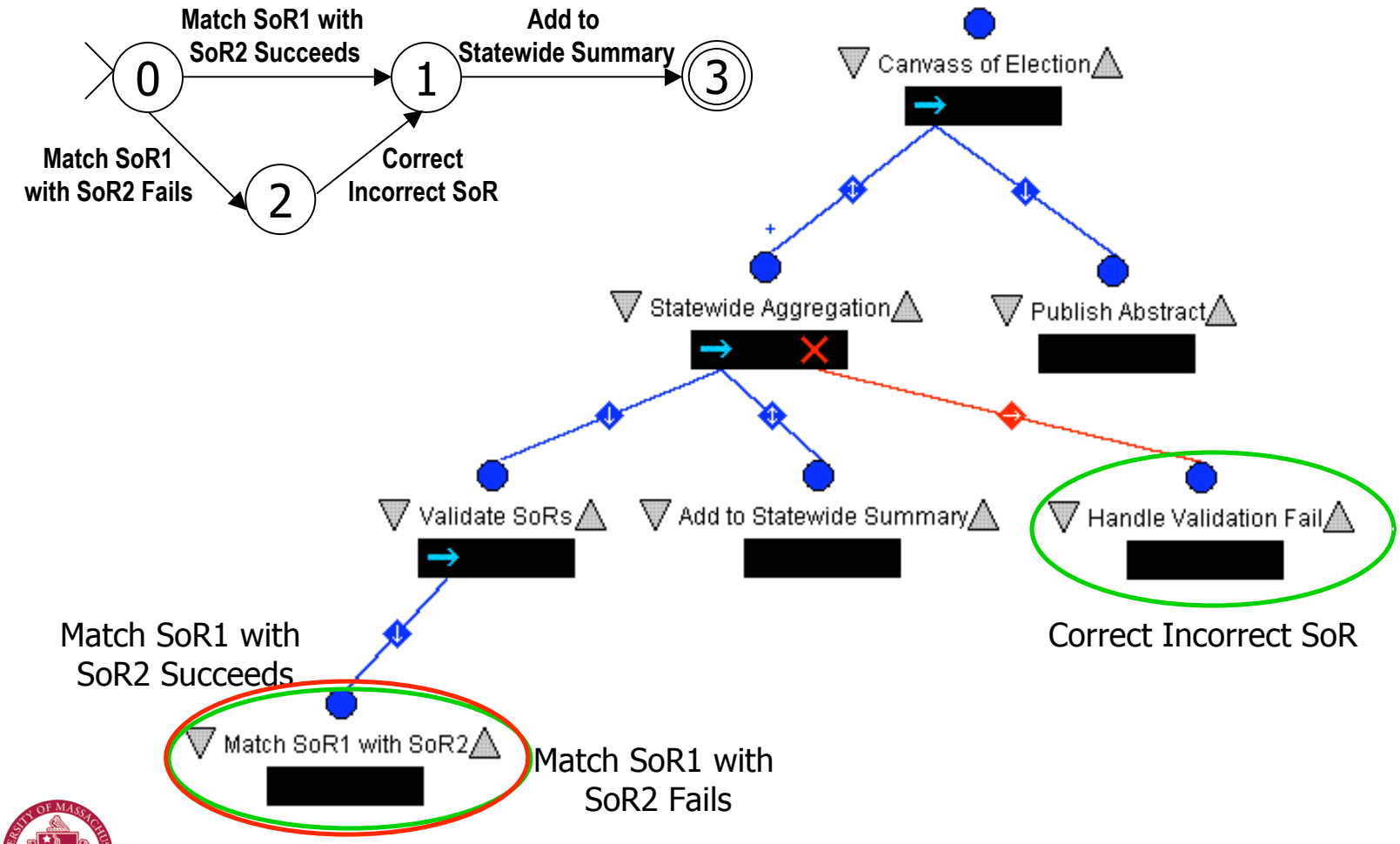


Property Specification: An Example

- *If two SoRs mismatch, the incorrect SoR gets detected and corrected before getting added to the Statewide Summary.*
- Transition labels in the property FSA corresponds to events in the process



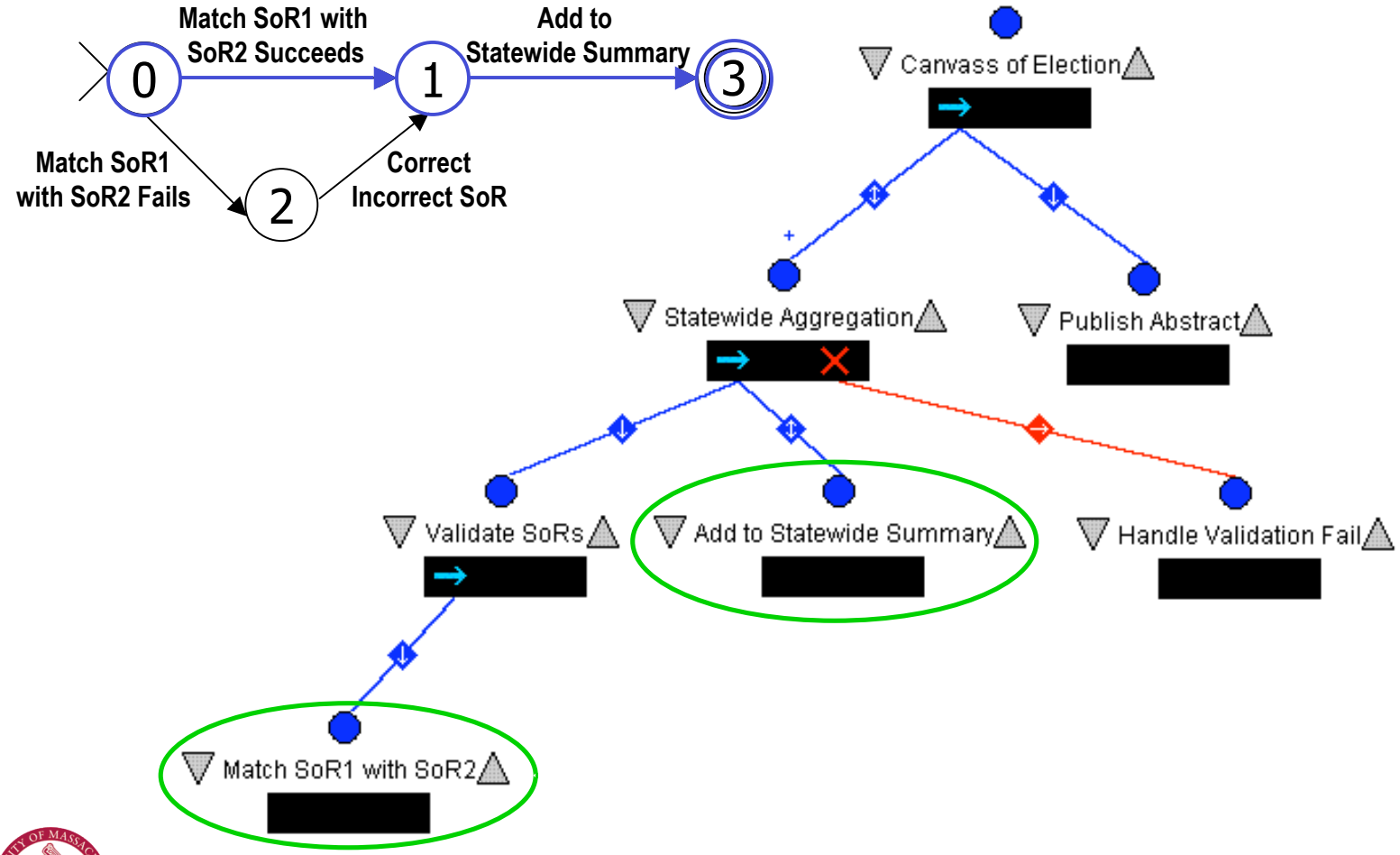
Process steps and property labels



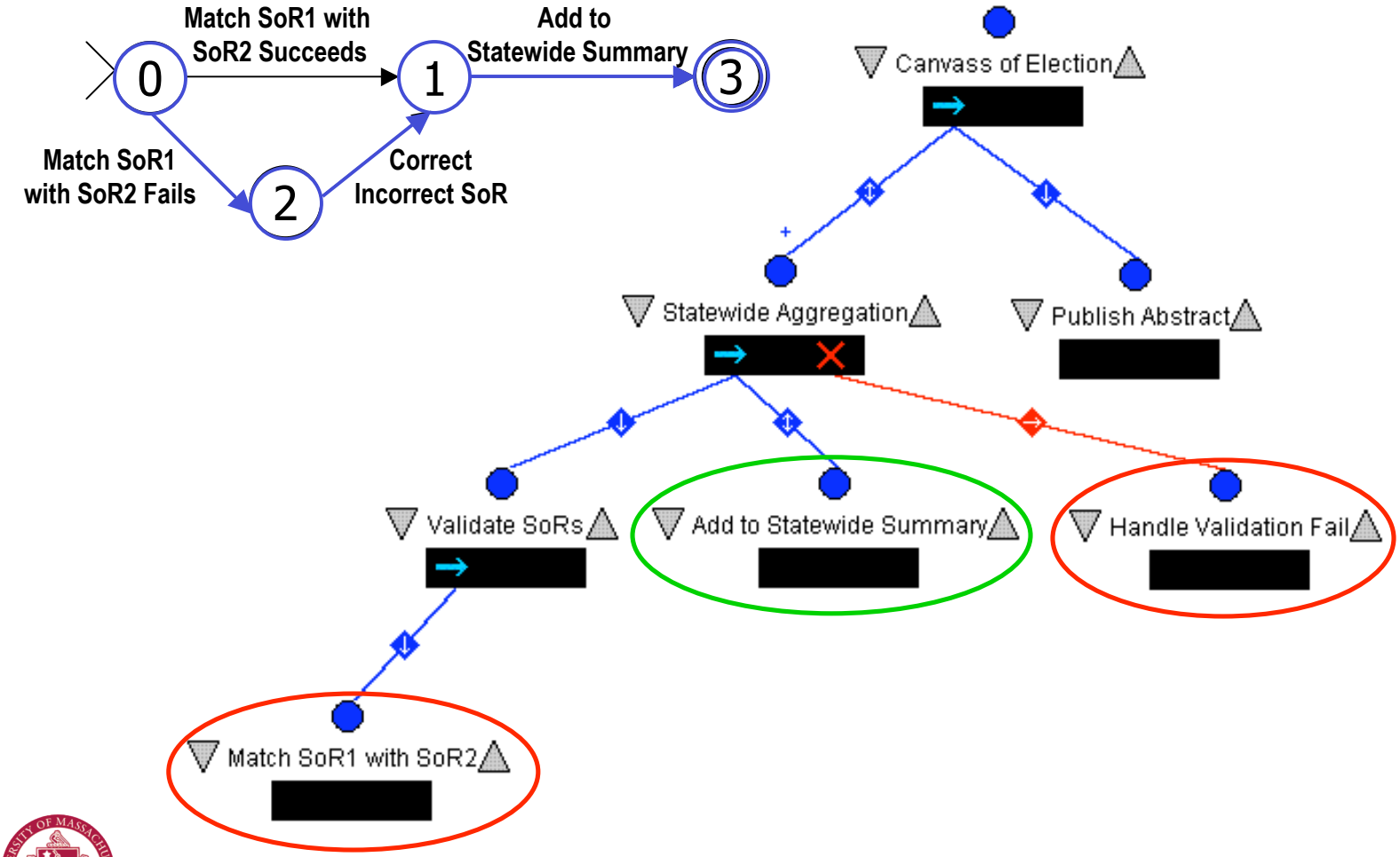
Analysis of Frauds

- Model two different poll workers that perform 'Prepare SoR' step
  - One honest and one dishonest
- FLAVERS analyzer evaluates all possible traces through the process definition
  - Verifies if property holds in all traces
  - If not, produces a counterexample
- Need for automated analysis

Step through an analysis example



Step through an analysis example



Observations

- The described property verifies resistance to some fraudulent behaviors
  - Catches one honest and one dishonest
- This property will not detect two colluding poll workers for this process
- Need additional properties and/or a modified process
  - The process should now verify the existing and the additional property
- Incremental process improvement

Example of an Additional Property

- *An SoR will never get added to the 'Statewide Summary' if it is different from the Machine_Total*
- Catches frauds with two colluding poll workers
  - May require changes in the process
- The new process is verified against both the properties
  - Increased resistance to frauds

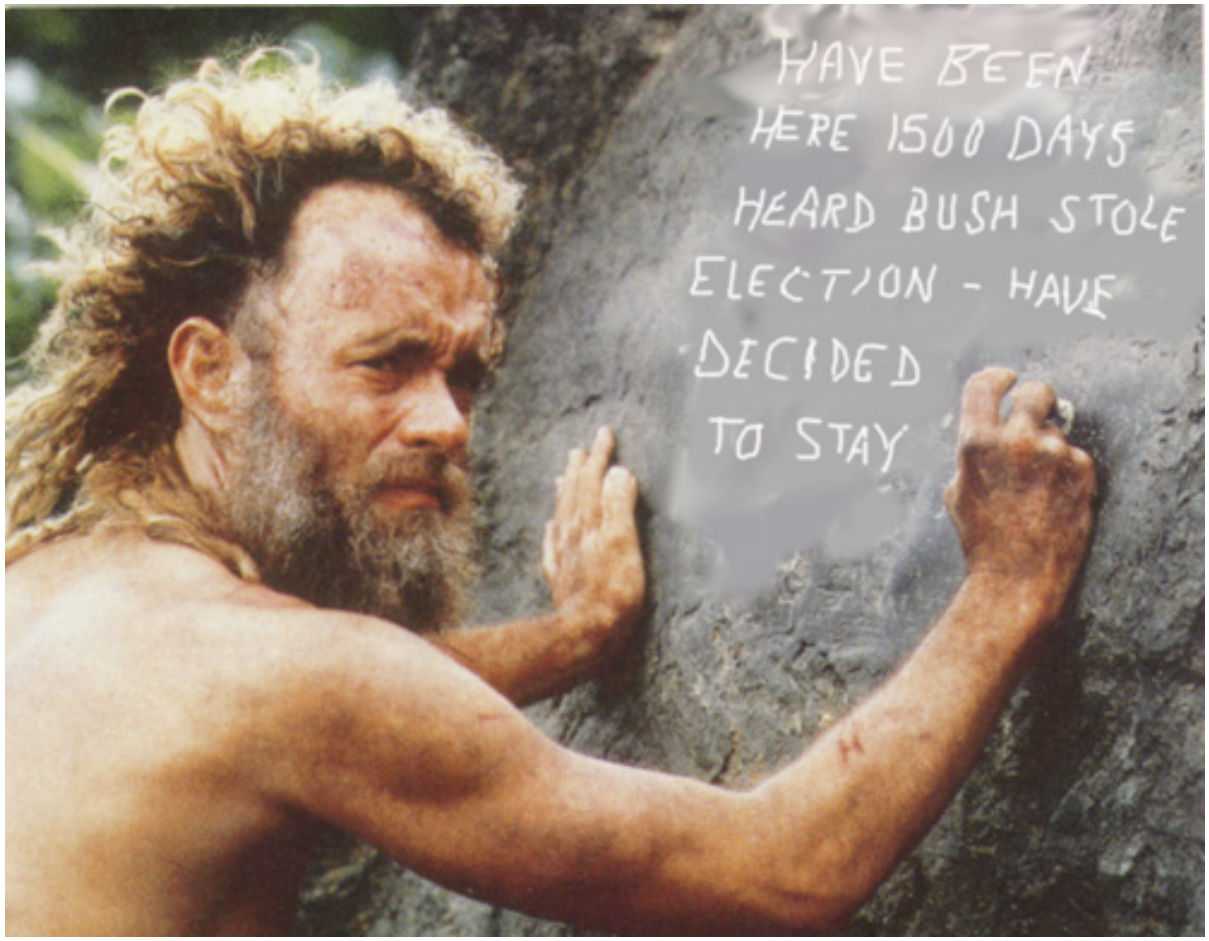
Conclusion

- There is more to election than just the voting machine
- No process will defend against all possible kinds of fraud
  - Needs to be guided by cost effectiveness
- Need a systematic study of process improvement
  - Our approach shows a promising direction
  - Demonstrates an important application of software process improvement to another domain

Future Work

- Model real world election processes
  - Likely to be large and complicated
  - Expected to have a lot of parallelism and exceptional flows
- Develop an ontology of election processes and fraudulent behaviors
- Identify most commonly occurring security vulnerabilities (fraud patterns)
  - Properties representing prevalent fraudulent behaviors
  - Pattern of resistant processes

Thank you!


HAVE BEEN
HERE 1500 DAYS
HEARD BUSH STOLE
ELECTION - HAVE
DECIDED
TO STAY

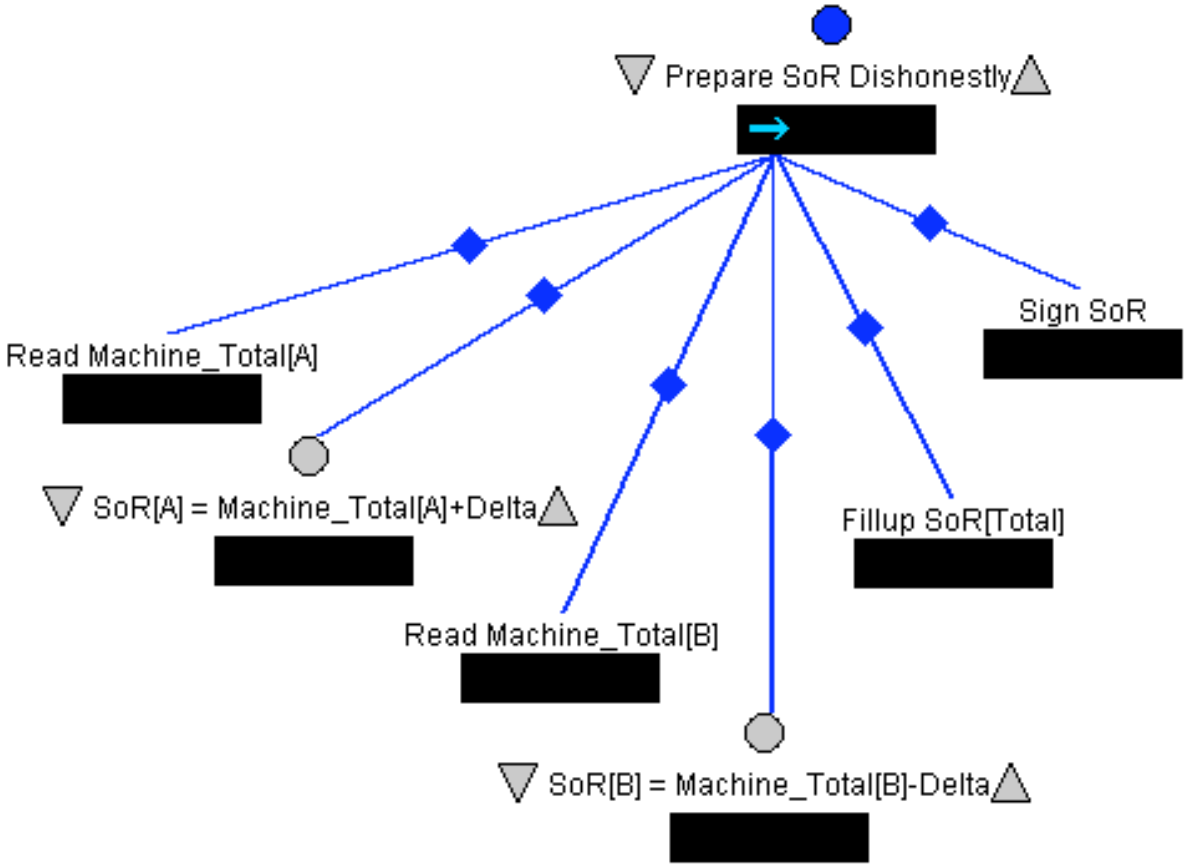
Extra Slide 0: Our Approach and Tools

- Develop a rigorous discipline of election process improvement:
  - Define an election process with appropriate level of details using LittleJIL
    - Add different (possibly malicious) agents
  - Define security policies using PROPEL
    - Properties that we want to be satisfied by an election process
  - Identify vulnerabilities using FLAVERS
    - Verify the properties or identify where in the process the properties fail
  - Improve the process or strengthen the property
  - Iterate

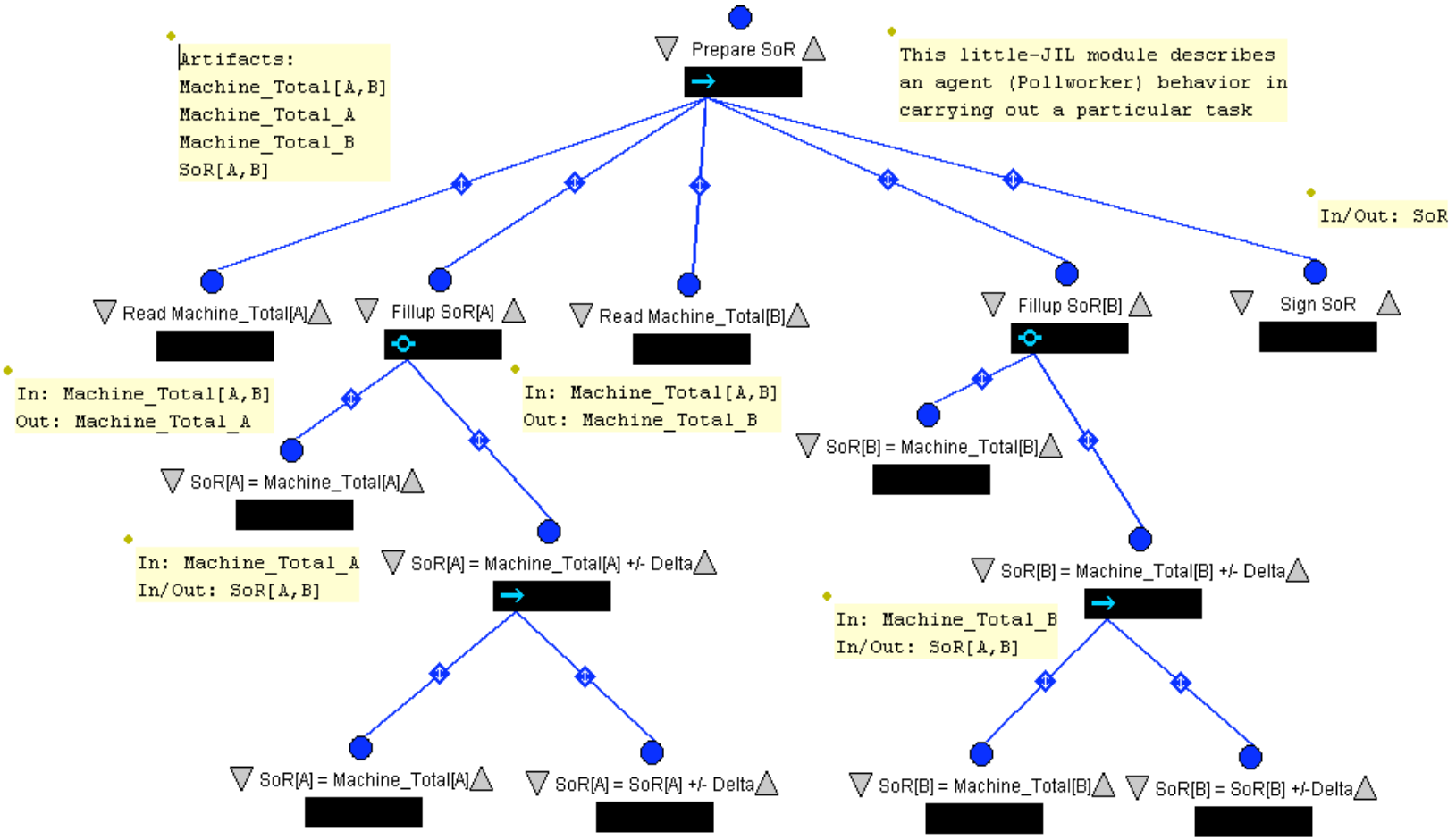
Extra Slide 0.1: Modeling Artifacts and Agents

- Artifact and resource information is attached to the step interfaces
- Artifact flow is bound to the edges
- Separation of coordination and computation
  - Agent behavior is orthogonal
  - For this study, agents have been modeled using Little-JIL

Extra Slides 1:
A Dishonest Agent (Cont.)



Extra Slide 2:
Agent Behavior (in Little-JIL)



Extra Slides 3: Tools and Techniques Used

- Use Little-JIL process language for modeling elections
  - Enrich the model with resource declarations and artifact definition and flow
  - Define agent behavior
- Model security properties using PROPEL
  - Properties as Finite State Automaton
- Verify the properties using FLAVERS analysis tool
- Iteratively change process and/or properties to improve election process

Extra Slide 4: Little-JIL Overview

- Visual coordination language
  - Rigorous semantics
- Hierarchical decomposition of tasks (steps)
- Rich exception handling with scoping
- Separation of Coordination and computation
  - Capability of defining agent behavior
- Declaration and flow definition of artifacts
- Orthogonal treatment of resource definitions
  - agents and other resources

Extra Slide 5: Little-JIL Step Structure