Analyzing and Modeling Tabulator Testing and Early Tabulation for Election Process

Alan Chen

Introduction

Elections are the fundamental building blocks of our democratic nation. The goal of an election is to elect representatives who reflect the will of the people. Representatives are elected by the people to further the people's best interest. Through fair elections, people have a say in what ways they want to be governed and what kind of future they want. Since elected representatives are accountable to the electorate, elections encourage some level of transparency. The intentions, beliefs, and past records can be under public scrutiny. Elections are important to our democracy. The electorate needs to have confidence in the election process. The safety of fair elections is to ensure that our democracy remains intact.

Following the 2016 presidential election, rumors about Russian meddling in U.S elections quickly circulated. The rumors resulted in a spotlight being shined on the safety of our elections. People began to investigate election safety. One of the findings was a data file containing an estimate of 14.8 million records left on an unsecured server without a password. Close to 16 gigabytes in size, this file contains dozens of fields including personal information like voter's name, address, gender and several years' worth of voting history [1]. Such information might help campaigns to position their political messaging to target specific voters. For example, if the individual's views about immigration or abortion rights were known, campaigns can use that information to flip a person's vote. Such data breaches can result in an unfair election.

Another major problem with our election system is the software used on voting machines. The public has been denied access to this software because it is proprietary information [2]. Although the voting machines are deeply tied to our democracy, the public does not have the right to know whether the algorithm has been checked or whether these systems are hackable. There is no proof that these machines have been through adequate testing. Unfortunately, there has been a history of hacked voting machines. Suspicions about the voting machine leads to a lower confidence level in the election's outcome.

In this project, we looked at the into procedure involving the tabulator by further examining two main processes, testing the tabulator and the early tabulation process. The goal of this project was to learn about the election processes and where they are vulnerable and to study approaches for evaluating election processes. Overall, we would like to provide feedback to election officials about what was learned.

Modeling and Analysis

Little-JIl Background

Little-JIl is a graphical language use to coordinate the flow of resources to complete processes [5]. In a Little-JIl process model, steps or subprocesses can specify an entire process. The step type determines which task to complete first next. Furthermore, steps can have resources also known as “artifacts” flowing in, out, or both. For example, an “online payment” process may have a sub-step call “enter credit card number” which has an artifact “credit card number” flowing in. The artifact is used to complete the steps.

Every step can have proactive and reactive control methods to respond to all possible scenarios. The proactive control methods are made up of five kinds of step:

Sequential Step – sub-steps run from left to right

Parallel Step – all sub-steps run at the same time

Choice Step – able to select which sub-steps to perform

Try Step – sub-steps run from left to right and stop when one sub-step succeeds

Leaf Step – a step that has no sub-step(s)

Sub-steps can throw exceptions that travels upwards until it reaches the root step, the highest step, or until it encounters an exception handler. Each step can have simple exception handlers for exceptions thrown by its sub-step. For reactive control methods, there are four types of continuation methods for a simple exception handler. These four types of ways to handle exceptions are:

Complete –finishes that process step

Continue –continues the process

Restart –restart the entire step process

Rethrow – retry the sub-step

In the “online payment” process example, the step “enter credit card number” may throw an exception “incorrect number”. The continuation method for the exception handler could be rethrow, which retries the credit card number.

Diagram

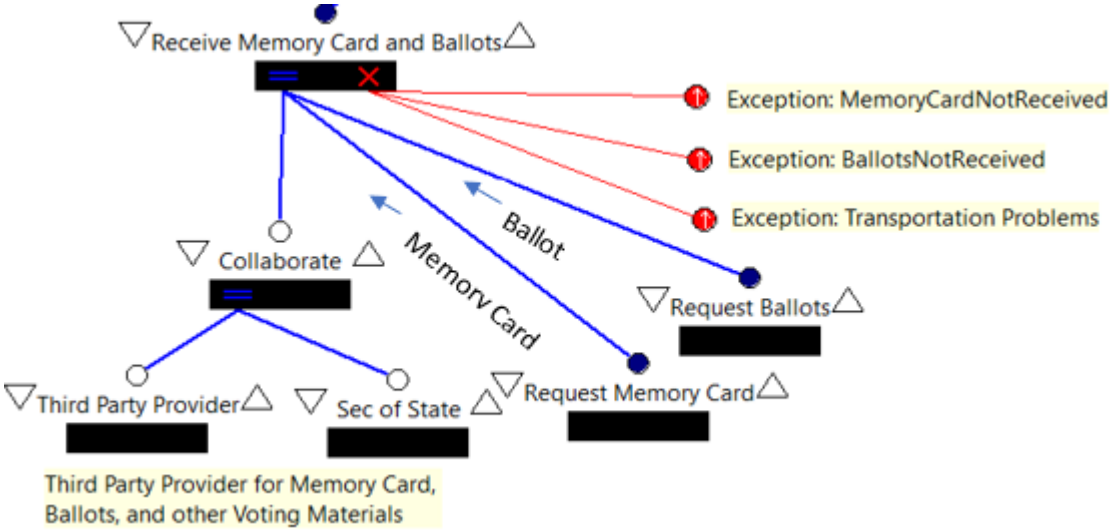

Figure 1. Little-JIl process model for Receive Memory Card and Ballot

Figure 1 is the receive memory card and ballots process. Since it is a sequential step, it first collaborates with the third-party provider and the secretary of state. Then it requests the memory card. The sub-step “Request Memory Card” may throw two exceptions, “Transportation Problems” and “MemoryCardNotReceived”. The artifact, memory card, flows from the sub-step “Request Memory Card” to the higher step “Receive Memory Card and Ballots”. The higher step is where the two exceptions thrown by the “Receive Memory Card” are being handled. The exception handlers use rethrow, which retries the sub-step where the exception originated from. Similarly, for the sub-step “Request Ballots”, it can throw “BallotsNotReceived” exception and has the artifact, ballots, as an output.

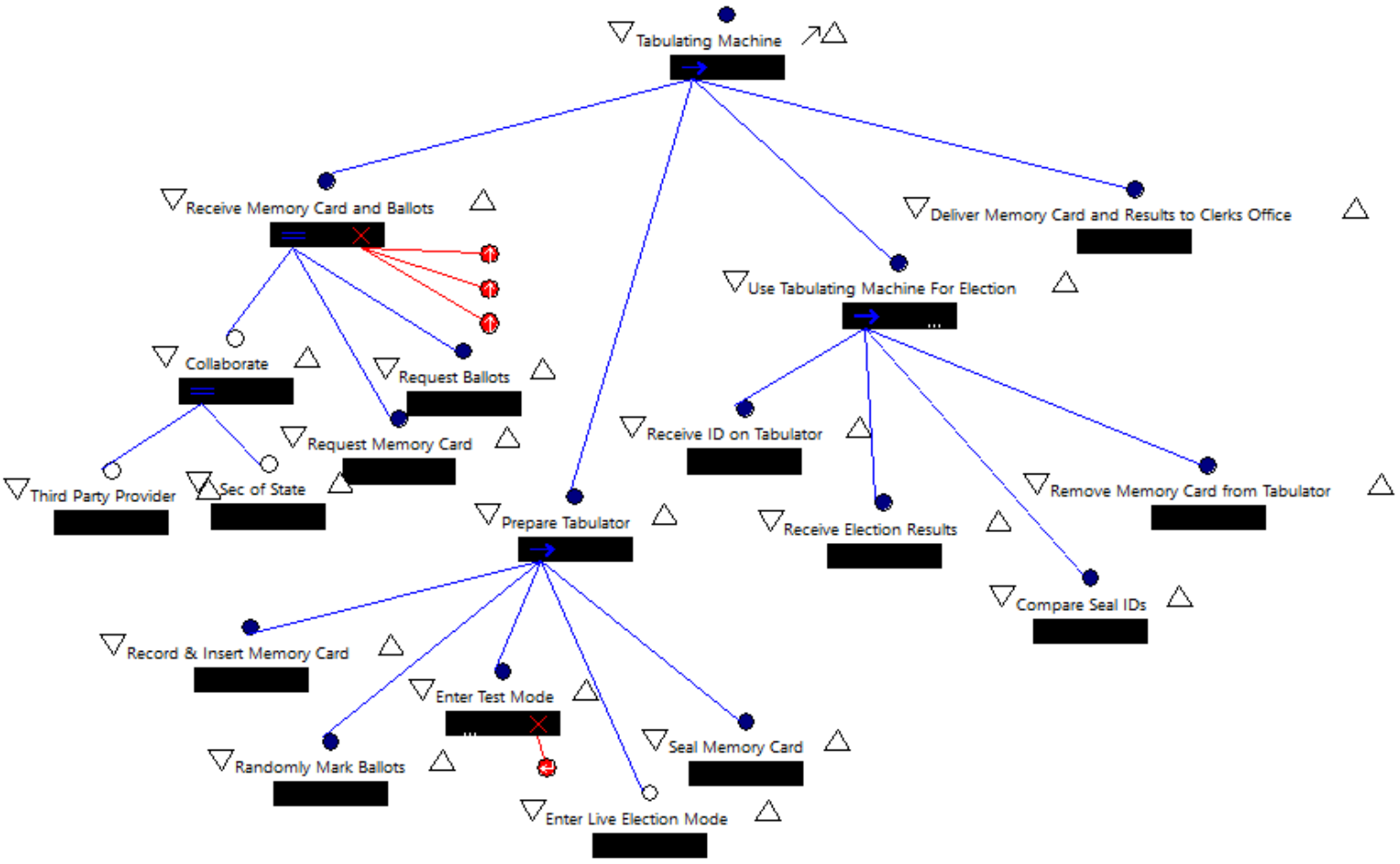

Figure 2. Little-JIl process model for Tabulating Machine

Fault Tree Analysis

Fault tree analysis (FTA) selects a step from the Little-JIl process model and utilizes the control and data flow of artifacts to determine if a specific problem could happen at that step, and if so how.

The data flow of artifacts can be either in, out, or in/out. The flow determines how artifacts move from one step to another. For example, in Figure 1, the “Receive Memory Card and Ballots” step receives the artifact, memory card, from its sub-steps. Furthermore, based on the data flow, artifacts may be localized. This means that the artifact is only needed to complete a subprocess and nowhere else. Therefore, there is no need for the artifact to flow out of the subprocess, hence the word local.

The control flow is determined by the exception handlers and step declarations. In an exception handler, there is a continuation action which determines how to deal with exceptions. The step declaration determines how the sub-steps are completed.

Using the Little-JIl process model, the FTA maps out series of cause and effect rigorously. By mapping it out, it looks at every possible problem tracing back to the origin.

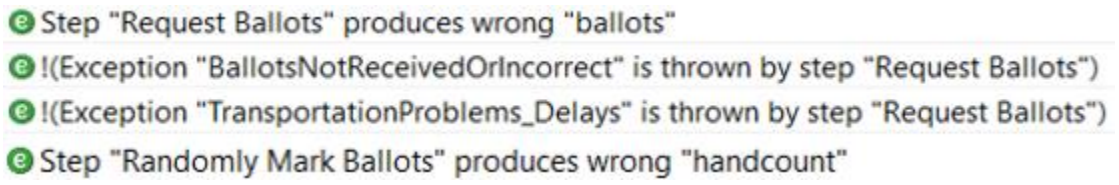

Figure 3. FTA table results from figure 2

Figure 3 is created from a hypothetical scenario that the artifact, hand count results (not shown in the figure 2), to the step “Enter Test Mode” is wrong. From figure 3, the possibilities that can result in wrong hand count results include incorrect ballots, transportation problems, and incorrect counting of randomly marked ballots.

Lesson learned from fault tree analysis

The fault tree analysis results suggested possible problems. Some of the suggested problems confirmed the observed problems. However, to effectively apply FTA, we would need to develop a more detailed Little-JIl process model since the FTA is derived from the process model. For example, from figure 3, the artifact, hand count results, could be wrong from a tabulator or memory card problem, but it was not shown in the FTA table. Nevertheless, FTA proposed some originating points that could cause problems.

Conclusion

What I learned

Election integrity is a serious problem. Some of these problems such as data breaches and proprietary software to the 2016 presidential election or even earlier. Yet, these problems still exist today. Some states still use electronic voting machine with no paper trial making the auditing process impossible. Moreover, for this project, we noticed some problems with tabulators testing and the early tabulation process.

The testing of the tabulator follows a very specific guideline to ensure that it works correctly. However, there are uncertainties, such as the adequate number of ballots to test. Due to proprietary software, we may never know what happens in the tabulator beyond the testing range or in live election mode.

Next, for the early tabulation process, the keys are the same for all AccuVote-OS ballot repository. Additionally, the early tabulation process is unsupervised making it easy to exchange, switch ballots around, or to remove ballots from the repository. Consequently, there could be incorrect number of votes for a candidate.

Finally, I learned about process modeling and analysis using Little-JIl. We create a Little-JIl process model and then used FTA to analyze the model. The resulting analysis is fairly aligned with my observed problems such as problems with hand count results from randomly marked ballots can be caused by human error.

Overall, I learned about election problems that still exist today, which could cause the integrity of votes to be at stake. I had the opportunity to observed problems that have occurred and even participated in parts of the tabulation process. Beyond that, we created a model that depicts the election process and use that to trace possible weak points within our election system.

References

[1] Whittaker, Z., & Whittaker, Z. (2018, August 23). Millions of Texas voter records exposed online – TechCrunch. Retrieved from https://techcrunch.com/2018/08/23/millions-of-texas-voter-records-exposed-online/

[2] Zetter, K. (2018, September 26). The Crisis of Election Security. Retrieved from https://www.nytimes.com/2018/09/26/magazine/election-security-crisis-midterms.html

[3] Premier/Diebold (Dominion) AccuVote OS. (n.d.). Retrieved from https://www.verifiedvoting.org/resources/voting-equipment/premier-diebold/accuvote-os/

[4] Email and Internet Voting: The Overlooked Threat to Election Security. (n.d.). Retrieved from https://www.commoncause.org/page/email-and-internet-voting-the-overlooked-threat-to-election-security/

[5] Little-JIL. (n.d.). Retrieved from http://laser.cs.umass.edu/tools/littlejil.shtml

[6] Voting Information. (n.d.). Retrieved from https://www.amherstma.gov/81/Voter-Information

[7] PD43 » Massachusetts Election Statistics. (n.d.). Retrieved from http://electionstats.state.ma.us/