# JETS

## The USENIX Journal of Election Technology and Systems

Volume 2, Number 2 • April 2014

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

# JETS

## The USENIX Journal of Election Technology and Systems

**Volume 2, Number 2 • April 2014**



*JETS* articles will be presented at the Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE).
www.usenix.org/conferences/evtwote

©2014 by The USENIX Association

All Rights Reserved

This volume is published as a collective work. Rights to individual papers remain with the author or the author's employer. Permission is granted for the noncommercial reproduction of the complete work for educational or research purposes. USENIX acknowledges all trademarks herein.

ISBN 978-1-931971-12-6 ISSN 2328-2797

### *JETS* Editorial Board

**Editors-in-Chief**

Walter Mebane, *University of Michigan*
Dan S. Wallach, *Rice University*

**Editorial Board**

Vittorio Addona, *Macalester College*
Ben Adida, *Mozilla Foundation*
R. Michael Alvarez, *California Institute of Technology*
Mary Batcher, *Ernst & Young*
Josh Benaloh, *Microsoft Research*
Stephen Checkoway, *Johns Hopkins University*
Jeremy Clark, *Carelton University*
Gustavo Delfino, *Universidad Central de Venezuela*
Jeremy Epstein, *SRI International and National Science Foundation*
Kentaro Fukumoto, *Gakushuin University*
James Heather, *University of Surrey*
Michael C. Herron, *Dartmouth College*
F. Daniel Hidalgo, *Massachusetts Institute of Technology*
Candice Hoke, *Cleveland-Marshall College of Law*
Joseph Kiniry, *Danmarks Tekniske Universitet*
Philip Kortum, *Rice University*
Martha Kropf, *University of North Carolina, Charlotte*
Sharon Laskowski, *National Institute of Standards and Technology*
Joseph Lorenzo Hall, *Center for Democracy and Technology*
Tal Moran, *Interdisciplinary Center Herzliya*
Olivier Pereira, *Université catholique de Louvain*
Maria Petrova, *New Economic School, Moscow*
Ronald Rivest, *Massachusetts Institute of Technology*
Mark D. Ryan, *University of Birmingham*
Peter Ryan, *University of Luxembourg*
Hovav Shacham, *University of California, San Diego*
Alexander A. Shvartsman, *University of Connecticut*
Alberto Simpser, *University of Chicago*
Philip Stark, *University of California, Berkeley*
Bob Stein, *Rice University*
Charles Stewart, *Massachusetts Institute of Technology*
Wendy Tam Cho, *University of Illinois, Urbana-Champaign*
Vanessa Teague, *University of Melbourne*
Alexander Treschel, *European University Institute*
Melanie Volkamer, *Technische Universität Darmstadt*
David Wagner, *University of California, Berkeley*
Douglas Wikström, *KTH Royal Institute of Technology*

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

*USENIX Journal of Election Technology and Systems (JETS)*
Volume 2, Number 2 • April 2014
www.usenix.org/jets/issues/0202

# Do Voters and Poll Workers Differ in their Attitudes Toward e-voting? Evidence from the first e-election in Salta, Argentina

Julia Pomares, Center for the Implementation of Public Policies Promoting Equity and Growth
Ines Levin, University of Georgia
R. Michael Alvarez, California Institute of Technology

We provide an analysis of voter and poll worker perceptions of the advantages and disadvantages of a new e-voting system vis–à–vis traditional ballot-and-envelope voting in the 2011 election in Salta, Argentina. The results of this comparison provide new insights into how poll workers perceive the implementation of new voting technologies and show that both points of view need to be taken into account when assessing new election technology. We found that speed is perceived to be the most important advantage of e-voting; and more so for poll workers than for voters. This is not surprising since speed is an aspect of a voting technology that directly affects the workflow of poll workers. We also found that poll workers expressed more intensely negative attitudes towards e-voting than voters, especially in relation to difficulty of use and lack of training. Finally, we found that both voters and poll workers placed more importance on usability than confidence issues. This is an unexpected finding since election authorities have identified confidence in the integrity of the election process as the main trigger of the adoption of the new voting technology. Analyses like the one conducted in this paper should be an integral component of the evaluation of the implementation of new voting technologies and introduction of important procedural changes.

## 1. INTRODUCTION

In nations where voting is conducted in-person, the poll workers who staff the balloting locations are an essential component of the election administrative process. Generally their duties include setting up and opening the polling place, authentication of potential voters, distribution and receipt of voting materials, maintenance of the ballot materials and voting systems during the voting period, dealing with problems that arise on election day, conducting initial tabulations of voting materials and ballots, handling initial audits of in-person voting, closing the polling place and transporting the election materials to the authorities. Poll workers have quite impressive responsibilities; they handle ballots and voting systems, and their actions help maintain critical goals of the process—keeping the balloting process secret, accessible, secure, and fair (to name just a few). It is no wonder that research has focused recently on poll workers: they have considerable discretion in the field during an election, and their behavior is one important aspect in the development of trustworthy elections with a high degree of integrity.[1]

While poll workers are critical for the conduct of trustworthy elections, research into how they handle their tasks is still in a nascent stage. There have been studies of poll worker discretion and how they exercise it (Alvarez and Hall 2006; Atkeson et al. 2010; Kimball, Kropf, and Battles 2006), of the characteristics and qualifications of poll workers as reported by local election officials (Kimball et al. 2009), of how voters evaluate the performance of the poll workers they encounter on election day (Claussen et al. 2008; Hall and Stewart 2013), and of the impact of voters' perceptions of poll worker performance on public confidence in the fairness of the election outcome (Hall, Monson, and Patterson 2009). However there have been few self-evaluations by poll workers presented in this growing body of research, nor direct comparison between how poll workers evaluate a particular election or administrative process and how voters evaluate the same election or administrative process. Thus, one goal of our paper is to provide such a direct comparison between how voters and poll workers evaluate the same election. We compare the advantages and disadvantages of two different voting systems—a traditional paper-based voting procedure vis–à–vis a new e-voting system—reported by voters who used each

[1] Recent research has focused on the discretion of poll workers, and how voters evaluate their efforts (Alvarez and Hall 2006; Atkeson and Saunders 2007; Clausson et al. 2008; Hall, Monson and Patterson 2008; Hall and Stewart 2013).

system, with the advantages and disadvantages reported by poll workers who served at a polling place where each voting system was used.

As important as the perceptions of poll workers are, there is scant research on their performance in Latin America. Much attention has recently been devoted to the issue of the independence of election authorities and its impact on voters' and legislators' confidence in the fairness of the election process (Hartlyn et al. 2008; Rosas 2010; Barrientos del Monte 2008). However, we do not know of any study on the perceptions of poll workers in Latin America, nor any studies that compare their perceptions with those from voters.

This lacuna is even more pressing as several countries in the region are implementing new procedures and voting systems, which constitute another important aspect of a poll worker's job. As most changes in election laws or voting technologies affect in-person voting, poll workers are often called upon to implement those changes. This is particularly true in jurisdictions where new voting technologies are being deployed for the first time—poll workers in those locations need to learn how to setup, maintain, use, and close down new voting technologies, as well as how to explain to voters how they are to be used. Thus, it is important to study poll worker evaluations of voting technologies, especially in elections where they are helping to deploy those technologies for initial use. This is another key goal of our research reported below, to study poll worker evaluations of new voting technologies being used extensively for the first time in an important election in Salta, Argentina. In addition to studying the perceived advantages and disadvantages of each voting system from the point of view of poll workers, we consider other poll worker evaluations, including confidence in their ability to operate the system, assessment of the length of the vote count, and evaluations of the ease of serving as a poll worker.

In this paper we present data from a unique research project that has evaluated the implementation of electronic voting systems in Salta, Argentina (Pomares et al. 2011; Alvarez et al. 2013). Our research team was able to survey voters who used the new electronic voting system deployed in the 2011 elections in Salta, as well as voters who used the traditional voting system; some of the initial results from that research were recently presented in Alvarez et al. 2013. We also were able to provide poll workers with a very similar questionnaire, and in this paper we concentrate on comparing the results from the voter survey with similar results from poll workers. The main part of our analysis in this paper is based on identical check-all-that apply questions about the advantages and disadvantages of each voting system that were included in both the voter and poll worker questionnaires. In the next section we discuss the electoral process in Argentina, as well as the electronic voting system recently deployed in Salta. We then briefly discuss our evaluation project, and present results that compare the evaluations of poll workers with those from the voters. In our conclusion we discuss the implications of this work, as well as new directions for future study.

## 2. HOW ELECTIONS WORK IN ARGENTINA

Although national elections in Argentina are regulated by federal legislation, election administration is highly decentralized, with each of the 24 provinces enjoying the autonomy to enact its own electoral and political institutions. Despite this, provinces throughout the country use similar voting procedures. Two salient features of the paper-based voting system used throughout the country are that it is based on the French ballot-and-envelope model, whereby voters select a party-specific ballot that specifies party nominees for all the races and place it inside an envelope; and that the layout of the polling place is such that voters wait in line to enter a "black room" (*cuarto oscuro*) where they make their choices in private, and subsequently leave the room and proceed to deposit the vote in a ballot box within sight of other voters and election authorities. An important consequence of this voting procedure for electoral behavior is that the practice of listing party nominees for all the races on the same partisan ballot discourages split-ticket voting, since in order to split the vote the voter has to select and cut sections of different partisan paper ballots by hand along dotted lines, and then place the preferred pieces (specifying the choice of different parties for different races) inside the envelope.

A second important consequence of the traditional paper-based voting procedure for electoral behavior is that each political party prints, distributes and supplies its own ballots on Election Day, which are displayed on tables inside the "black room." This system used to work fairly effectively while there were two main parties (Peronist Party and the Radical Party) of relatively equal size, territorial outreach and resources. In recent years, however, extreme party fragmentation rendered this voting system archaic, ineffective and inequitable. Small parties find it difficult to guarantee their ballot is on the table. Therefore, bigger parties and incumbents enjoy an important advantage due to broader capabilities for printing, distributing and watching over their ballots. Similar to American voting procedures before the Australian ballot reform (Rusk 1970; Walker 2005; Ware 2002), this voting procedure also empowers party leaders at the local level who are in charge of recruiting party monitors and organizing ballot distribution and control at each voting precinct (Ware 2002).

Salta, a northern province located in the foothills of the Andes with an electoral roll of 850,000 voters, was the first province to introduce an e-voting system. The system was designed to resemble the country's traditional voting system and was implemented gradually. It was first tested on a small scale during the 2009 provincial elections, for the second time during the January 2011 open primary elections, and for the third time during the April 2011 general elections. The 2011 implementation of e-voting comprised 33% of the province's electoral roll and targeted voters concentrated in the capital of the province and surrounding municipalities. The justification for first rolling-out the system in urban areas was to make the process of deploying e-voting voting machines less cumbersome, and to evaluate the performance of the system when used by voters with relatively high levels of educational attainment and technological sophistication.

Whatever the voting procedure, every polling table is presided by a poll worker (*presidente de mesa*), who cannot be affiliated with any political party. She is the ultimate authority at the polling table and is helped by an auxiliary poll worker. Political parties or alliances can nominate one party monitor per polling table. Smaller parties find it more difficult to guarantee the presence of party monitors in every table, and strive to at least secure a party monitor at each precinct. An important feature of the roll-out of e-voting in Salta to bear in mind for the analysis is that poll workers assigned to e-voting precincts faced stronger incentives to attend training sessions.[2] Although all poll workers were recruited among elementary school teachers, those using the e-voting system were told that participating in the full training program would provide them with additional points for future promotion. This compensation scheme was designed by the Electoral Tribunal of the province together with the Ministry of Education.

The e-voting system, developed by an Argentine private vendor, was designed as to offer a voting experience resembling the traditional voting system. Instead of placing a party-specific ballot inside an envelope, voters insert a blank ballot inside the voting machine and subsequently select their preferred candidates for each race via the machine's touch screen. Unlike other types of DRE machines, these machines serve only as voting machines and do not keep track of votes cast. After a voter has made all choices, the machine prints the different selections on the ballot, as well as on a Radio Frequency Identification (RFID) chip embedded on the same ballot. After the ballot has been printed, the voter can verify whether options printed on

[2] The poll worker training was conducted by the Electoral Tribunal. The handbook provided to poll workers in voting e-locations is at http://www.electoralsalta.gov.ar/Sitio/Eleccion/VotoElectronico/ManualAutoridadesMesa2012.pdf.

the ballot coincide with the selections made using the machine's touch screen, as well as with the options recorded on the RFID chip. Lastly, in the same way as it is done under the traditional voting system, the voter inserts the paper ballot into a ballot box.

Although machines are located close to the polling tables where poll workers are positioned, polling places are designed as to guarantee voters' privacy. Nonetheless, the introduction of e-voting implies an important change to the layout of the polling place. The act of selecting electoral options behind doors inside the "black room" is replaced by the act of using a voting machine within sight of other voters and electoral authorities. This change might alter perceptions of vote secrecy. Previous research by the authors about voter attitudes toward e-voting and traditional voting in the same election (Alvarez et al. 2013) confirmed that there is a negative effect of e-voting on perceptions of vote secrecy.

In addition to being used by voters during polling hours, e-voting machines contain scanners that are used by poll workers to tally votes after the election is closed. In order to ensure that options recorded in the RFID chip are consistent with those marked on the ballot, poll workers scan the paper ballot and verify that choices coincide with the printed text. Party monitors gather around poll workers and also make sure that both pieces of information coincide. Since votes are counted electronically by passing them through the machine's reader, the preliminary tally is faster than under the traditional voting system where poll workers have to open each envelope, unfold the ballot (or pieces of ballots if the voter happened to split her vote), and count it by hand.

## 3. DATA AND RESULTS

Next we analyze attitudes toward voting technologies reported by voters and poll workers interviewed on Election Day, during the April 2011 general election in the Province of Salta. The voter sample comprised 1,502 individuals, mostly located in urban municipalities, with 59% of them voting in e-voting locations. E-voting and traditional voting stations were selected to produce samples of voters with similar socio-economic characteristics. Interviewers were instructed to randomly recruit voters on their way out of the polling locations. Interviewers had age quotas and half of their surveys had to be administered to men. We also administered a survey of 102 poll workers: 65 election authorities working in 23 e-voting locations, and 47 election authorities working in 15 traditional voting locations. As a part of a larger project, detailed reports from independent observers in the polling locations were also collected (Pomares et al. 2011). While voters and poll workers who used or administered the traditional voting procedure were asked primarily about their evaluation of that method, those who used or administered e-voting were asked primarily about their evaluation of e-voting.

Among several questions about confidence in the election process, ease of use and support for substituting traditional voting with electronic voting, we asked voters and poll workers about the advantages and disadvantages of the voting system that they used or administered. Because the wording of the latter questions was identical in all surveys (that is, surveys of voters and poll workers in e-voting and TV locations), we can compare results and analyze what attributes of the voting system are perceived to be relevant from voters and poll workers' perspectives depending on the type of system that they used.[3] There are two different

[3] In interpreting results it should be taken into account that voters and poll workers who used or administered the traditional voting system did not interact with the e-voting system, and had never tried an alternative voting technology in a previous election. Although it is conceivable that expectations about the pros and cons of e-voting could have colored the responses of voters and poll workers assigned to TV locations, their experiences with the TV system are the only *actual* voting experiences that could have influenced their responses. In contrast, voters and poll workers assigned to e-voting locations might have taken into account their recent experience with the use and operation of the e-voting system,

types of comparisons to be made: between attitudes toward voting systems reported by voters and poll workers; and between attitudes of either actor (voters *or* poll workers) towards e-voting and traditional voting.

Table 1. Voters and Poll Workers: Advantages of Voting Systems

| | E-voting Precincts (EV) | | Traditional Voting Precincts (TV) | |
|---|---|---|---|---|
| | Voters | Poll Workers | Voters | Poll Workers |
| Speed | 70.1 | 84.4 | 32.5 | 17.4 |
| More precise count | 7.4 | 25.0 | 1.4 | 2.2 |
| Reduced risk of fraud | 4.4 | 15.6 | 4.9 | 4.3 |
| More confidence | 7.7 | 9.4 | 10.8 | 8.7 |
| Ease of use | 28.9 | 7.8 | 37.0 | 26.1 |
| Some other advantage | 12.4 | 4.7 | 13.1 | 34.8 |
| At least one advantage | 94.3 | 93.8 | 82.4 | 78.3 |

*Note*: The table shows the distribution of responses to a "check-all-that-apply" question on advantages of voting systems, for voters (887 who used EV and 617 who used TV) and election authorities (65 who administered EV precincts and 47 who administered TV precincts).

Results corresponding to perceived advantages of voting systems from the point of view of voters and poll workers are presented in Table 1. The table shows the percentage of voters and poll workers who specified *speed*, *more precise count*, *reduced risk of fraud*, *more confidence*, and *ease of use* as advantages of each voting system. A first important finding is that when asked about the advantages of each voting system, e-voters and poll workers administering e-voting locations are considerably more likely to select *speed* as an advantageous attribute of the system than voters and poll workers in TV locations. Specifically, *speed* is identified as an advantage of e-voting by 70% of e-voters and as many as 84% of poll workers administering an e-voting table. This result comes as little surprise in the case of poll workers since *speed* is an attribute that affects poll workers' duties. Another similarity in the perceptions of voters and poll workers is that both are more likely to specify *more precise count* as an advantage of e-voting than of traditional voting. Lastly, both types of actors are considerably less likely to identify *ease of use* as an advantage of e-voting than of traditional voting; especially so in the case of poll workers. Thus, voters and poll workers largely coincide when it comes to evaluations of the relative speed, accuracy, and usability of voting systems.

However, do poll workers and voters have similar perceptions of the integrity of the electoral process? We expect some differences since each interacts with the voting system in different ways. Poll workers are responsible for the overall conduct of operations at the polling table and, as a consequence, they interact with the voting system throughout the voting process (from setting up the polling table to the preliminary tally of results) whereas the voter only interacts with the system at the voting stage of the process. Indeed, we find that the similarities between the perceptions of voters and poll workers stop when it comes to features of the systems related to trust in the election process such as reduced risk of fraud and increased confidence.

as well as previous experiences with the traditional voting system, in formulating their evaluations of the new voting system.

While e-voters are slightly less likely to identify *reduced risk of fraud* and *more confidence* as an advantage of the system compared to traditional voters, the opposite is observed for poll workers administering e-voting tables. The lower likelihood of identifying *more confidence* as an advantage among e-voters (relative to traditional voters) is somewhat worrisome, since one of the key arguments of the electoral authorities in the move toward e-voting was promoting voters' confidence in the integrity of the electoral process.

Table 2. Voters and Poll Workers: Disadvantages of Voting Systems

| | E-voting Precincts (EV) | | Traditional Voting Precincts (TV) | |
|---|---|---|---|---|
| | Voters | Poll Workers | Voters | Poll Workers |
| Manipulation of results | 1.6 | 1.6 | 9.1 | 11.9 |
| Less control by election authorities | 0.0 | 0.0 | 4.9 | 2.4 |
| High costs | 0.2 | 1.6 | 4.6 | 0.0 |
| Difficulty | 2.4 | 29.5 | 5.9 | 16.7 |
| Some other | 9.1 | 42.6 | 21.3 | 61.9 |
| At least one | 12.1 | 73.8 | 42.6 | 76.2 |

*Note*: The table shows the distribution of responses to a "check-all-that-apply" question on disadvantages of voting systems, for voters (887 who used EV and 617 who used TV) and election authorities (65 who administered EV precincts and 47 who administered TV precincts).

Next we turn to the perceived disadvantages of each system from the point of view of voters and poll workers. In Table 2 we present the percentage of voters and poll workers who identified *manipulation of results*, *less control by election authorities*, *high costs*, and *difficulty* as disadvantages of each system. An important finding is that voters are more eager to name disadvantages of traditional voting than the electronic method. Whereas 43% of traditional voters were able to name at least one disadvantage of ballot-and-envelope system (with *manipulation of results* being the one identified the most frequently), only 12% of e-voters named a disadvantage of the new voting system. These relatively positive attitudes towards e-voting among voters were confirmed by a previous study in which we showed strong support for replacing traditional by electronic voting system (Alvarez et al. 2013). In the case of poll workers, however, about three quarter of respondents identified some disadvantage of the voting system, regardless of whether they administered an e-voting or traditional voting table. There were, nonetheless, considerable differences in the nature of the shortcomings identified by authorities administering each system. In particular, while as many as 30% of poll workers in e-voting locations selected *difficulty* as a disadvantage of the system, only 17% of poll workers in TV locations did so. In contrast, as many as 12% of poll workers in TV locations selected *manipulation of results* as a disadvantage, compared to only 2% of poll workers in e-voting locations. These findings suggest that poll workers exhibited mixed opinions about the relative benefits of e-voting; while they expressed concerns about usability, they rarely expressed trepidations about the impact of the new system on confidence and security.

Further evidence about the pros and cons of e-voting and traditional voting can be drawn from answers to follow-up questions about *other* advantages and disadvantages of the systems posed to voters and poll workers. Although relatively few respondents provided answers to these questions, voters and poll workers in traditional voting locations were more likely to identify the fact *there is more experience with this system* and being *more beneficial for the elderly* as advantages of TV. While few poll workers reported advantages of e-voting (other than those discussed earlier in this section), several voters reported other advantages, including *no paper waste*, that it is *easy to assemble the vote by race* (i.e. greater ease of split-ticket voting), *more modern use of technologies*, and that *all electoral options are easily visible* as advantages of e-voting. Also as we show in Table 2, voters and poll workers were considerably more likely to identify *other* disadvantages of traditional voting than of e-voting. In particular, voters and poll workers in TV locations often identified *slowness* and the fact that it *generates long waiting times* as a disadvantage of the traditional system. Individuals voting or administering the election in e-voting locations, in turn, often identified *lack of training and experience*, *more difficult for the elderly*, or that *the machines can break* as disadvantages of the e-voting system. The latter disadvantages point to concerns about insufficient ease of use and (lack of) familiarity rather than to concerns about security and confidence. In line with the results discussed earlier in this section, concerns about usability seem to be more prominent in the minds of voters and poll workers than concerns about security and confidence.

Table 3. Poll Workers: Other Evaluations

| | E-voting Precincts (EV) | Traditional Voting Precincts (TV) |
|---|---|---|
| Precision of instructions (1-4 scale) | 3.2 | 2.7 |
| All elements available for opening on time (%) | 81.5 | 66.0 |
| Length of vote count (minutes) | 35.4 | 125.1 |
| Confidence in ability to operate system (%) | 81.5 | 70.2 |
| Ease of election authority job (1-4 scale) | 3.1 | 2.8 |

*Note*: The table shows the distribution of responses to questions on the precision of instructions; whether all elements were available for opening the precinct on time; the number of minutes the vote count lasted; the confidence in the person's own ability to operate the voting system; and the ease of the job. Figures correspond to answers given by 65 election authorities working in EV precincts and 47 election authorities working in TV precincts.

In addition to asking about the advantages and disadvantages of each voting system, the poll worker questionnaire assessed other aspects of the electoral process, and included questions about poll workers' confidence in their ability to operate the voting systems, assessments of the precision of instructions, evaluations of whether all necessary elements were available for opening up the polls on time, estimation of the length of the vote count, and appraisals of the ease of the poll worker job. Table 3 shows how responses to these additional questions vary between poll workers assigned to e-voting and TV polling places. In general, poll workers assigned to e-voting locations describe the e-voting procedure in more positive terms than those assigned to TV locations; they assigned higher ratings to the precision of instructions and they were more likely to say that all elements were available for opening on time. There are no meaningful differences in terms of confidence in their ability to operate the system and the perceived ease of the poll worker job between e-voting and TV locations. The most prominent difference between responses provided by e-voting and TV poll workers lies in the estimated length of the vote count, which is reduced by about 90 minutes in e-voting locations relative to TV locations. The latter result is consistent with the previously discussed finding that poll workers in e-voting locations were considerably more likely to select *speed* as one of the advantages of the voting system.

Table 4. Characteristics of Poll Workers

| | E-voting Precincts (EV) | Traditional Voting Precincts (TV) |
|---|---|---|
| Age (years old) | 42.7 | 40.4 |
| Education (1-8 scale) | 6.3 | 5.8 |
| Teacher (%) | 84.6 | 78.7 |
| Male (%) | 29.2 | 10.6 |
| Technology count (1-6) | 5.1 | 5.3 |
| First time as election authority (%) | 35.4 | 66.0 |
| Received poll worker training (%) | 98.5 | 57.4 |

*Note*: The table shows the distribution of poll worker attributes. Figures correspond to answers given by 65 election authorities working in e-voting precincts and 47 election authorities working in TV precincts. The "Education" scale includes the following eight ordered categories: incomplete primary or less; complete primary; incomplete secondary; complete secondary; incomplete tertiary; complete tertiary; incomplete university; and complete university or more. The "technology count" scale is an additive index consisting of the sum of dummy indicators of use of various technologies, including: Internet at work, Internet at home, Automated Teller Machines (ATMs), cellphone, and home computer.

A caveat is in order with regards to differences between poll workers in each voting system. The level of training each received was different (see Table 4).[4] While almost all poll workers in e-voting precincts declared that they had received training before Election Day, only 57% of poll workers in traditional voting precincts answered yes. According to interviews with election authorities, a significant effort was built into training poll workers in e-voting. Also, as mentioned in the introduction, a compensation scheme was in place for promoting training among e-voting poll workers. Nonetheless, since recruitment of poll workers in traditional voting areas usually includes a large proportion of individuals who served as poll workers in some previous election, it can be expected that traditional poll workers were in less need of training compared to those in e-voting areas where the system was implemented for the first time.[5]

## 4. DISCUSSION

Our research contributes to the growing literature on election administration in a number of different ways. First, we provide important evaluative data from an implementation of a new e-voting system in Salta, Argentina. Recently, scholars, administrators and advocates have called for more systematic and thorough evaluation of electoral reforms (e.g., Alvarez, Atkeson, and Hall 2012). This e-voting implementation follows that advice; researchers were given the opportunity to systematically survey voters and poll workers, to observe the election administration process, and to have access to other data from the election. In this paper we have focused on the advantages and disadvantages of new voting system vis-à-vis the older one from both voter and poll worker' points of view, and we argue that analyses like these should be an

[4] Since the sample size is small in the case of the poll worker survey, our assessment of the impact of e-voting on poll worker evaluations is based on the complete sample of poll workers.
[5] There were also slight differences between voters assigned to e-voting and traditional voting locations. Since e-voting was introduced in polling places where the electorate was more technologically savvy and had higher socio-economic status, e-voters exhibited slightly higher levels of education, technological sophistication, and political information (See Alvarez et al. 2013).

integral component of all implementations of new voting technologies and important procedural changes.

Second, our study has produced new insights into how poll workers perceive the implementation of new voting technologies. We find that evaluations of e-voting were considerably more mixed among poll workers than among voters. Although poll workers in e-voting locations were more likely to say that the system was fast (specially at the counting stage), more precise, and less vulnerable to fraud, compared to those in traditional voting locations; they were also more likely to express concerns about lack of training and to say that the system posed usability challenges. In light of these findings—which suggest that poll workers are not as convinced as voters are that the e-voting system is easy to use—performance practice procedures (e.g. Alvarez, Atkeson, and Hall 2012) would advise administrators to follow up and determine why poll workers exhibit these concerns, and how these issues can be resolved through improvements in training, voter education, and perhaps technology upgrades.

Third, we found that both voters and poll workers are more likely to mention usability than confidence issues. Speed, in particular, is identified as an advantage of e-voting by both voters and poll workers, and especially so for the latter. This may not be surprising, since speed is an aspect of a voting technology that directly affects the workflow of poll workers. Similarly, we found that higher precision of the vote count is perceived to be an advantage of e-voting relative to TV more for poll workers than voters. These findings all point to the conclusion that both voters and poll workers were more concerned about how these new technologies contribute or hinder the familiarity and use of the system than about the potential impact on the transparency of the election. This is an unexpected finding since election authorities have identified the need to instill confidence in the integrity of the election process as the main trigger for the adoption of e-voting. Although the main focus of both—when identifying pros and cons of each voting method—is on usability, poll workers assign some relevance to the perceived lower risk of fraud while this issue is scarcely mentioned by voters. The fact that usability gets higher salience than confidence might arise from the fact that this was the first e-election and implemented in a gradual fashion and, as such, not be conceived as a definitive change. These are important findings for the future use of e-voting in Salta, in other provinces in Argentina, and of course for other nations that are implementing new voting technologies.

Also, in our work we have provided new results that solidify our understanding of the important role that poll workers play in the administration of elections. There has been considerable recent interest in studying poll workers (for a good summary of this literature see Hall and Stewart 2013). Here we follow in the footsteps of previous scholars and focus on poll workers, but we do so in a relatively novel way—using parallel surveys that both voters and poll workers completed in the election. That gives us an excellent opportunity to understand how poll workers, who are on the front-lines of democracy, perceive the implementation of this new voting system in Salta. This methodology also lets us compare those perceptions and evaluations directly with the opinions of voters. Of course, we caution that our results here may not be easily comparable to studies of poll worker perceptions in other nations, especially the United States (where most of the research to date on poll workers has been conducted). Differences in electoral law and regulations, election procedures, political context, poll worker selection mechanisms, and their training programs all will need to be considered before such comparisons could be made. But we hope that our research may point the way for a more comparative analysis of election administration (especially of poll workers), and we recommend that this approach be replicated in future studies of new voting system implementation. Differences in poll workers and voters opinions show that both views have to be taken in board when attempting to create a thorough evaluation of the implementation of an election reform.

Finally, our research here has examined mainly questions of how voters and poll workers react to the implementation of a new voting system. Similar methodologies could be used to examine more specific features of voting systems, in particular various technical or security features of those systems. For example, researchers should study not only how voters evaluate to verification mechanisms (Llewellyn et al. 2013), but also how poll workers evaluate vote verification procedures and technologies. Thus we believe that studies of voting system security, and new voting technologies more generally, should also include poll workers so that their evaluations of the usability, integrity, and reliability of new technologies can be assessed.

## REFERENCES

R. Michael Alvarez and Thad E. Hall. 2006. “Controlling Democracy: The Principal-agent Problems in Election Administration.” *Policy Studies Journal* 34, 4 (2006), 491-510.

R. Michael Alvarez, Lonna R. Atkeson, and Thad E. Hall. 2012. *Evaluating Elections*. New York: Cambridge University Press.

R. Michael Alvarez, Ines Levin, Julia Pomares, and Marcelo Leiras. 2013. “Voting Made Safe and Easy: The Impact of e-voting on Citizen Perceptions.” *Political Science Research and Methods* 1, 1 (2013), 117-137.

Lonna Atkeson and Kyle Saunders. 2007. “Election Administration and Voter Confidence: A Local Matter?” *PS: Political Science & Politics* 40, 4 (2007), 655-660.

Lonna Rae Atkeson, Lisa Bryant, Thad Hall, Kyle L Saunders, and R. Michael Alvarez. 2010. “New Barriers to Voter Participation: An Examination of New Mexico’s Voter Identification Law.” *Electoral Studies* 29, 1 (2010), 66-73

Fernando Barrientos del Monte. 2008. "Organismos electorales y confianza en las elecciones en América Latina." In *Texto preparado para el Seminario de Investigación del Área de Ciencia Política y de la Administración.* Universidad de Salamanca.

Ryan L. Claasen, David B. Magleby, J. Quin Monson, and Kelly D. Patterson 2008. "At Your Service: Voter Evaluations of Poll Worker Performance.” *American Politics Research* 36, 4 (2008), 612-634.

Thad E. Hall, Quin Monson, and Kelly Patterson. 2007. “Poll Workers and the Vitality of Democracy: An Early Assessment.” *P.S.: Political Science and Politics* 40, 4 (2007), 647-654.

Thad E. Hall, Quin Monson, and Kelly Patterson. 2009. “The Human Dimension of Elections: How Poll Workers Shape Public Confidence in Elections.” *Political Research Quarterly* 62, 3 (2009), 507-22.

Thad E. Hall and Charles Stewart III. 2013. “Attitudes Toward Poll Workers in the 2012 Election.” Manuscript, April 4, 2013.

Jonathan Hartlyn, Jennifer McCoy, and Thomas Mustillo. 2008. "Electoral Governance Matters. Explaining the Quality of Elections in Contemporary Latin America." *Comparative Political Studies* 41, 1 (2008), 73-98.

David C. Kimball, Martha Kropf, and Lindsay Battles. 2006. “Helping America Vote? Election Administration, Partisanship, and Provisional Voting in the 2004 Election.” *Election Law Journal* 5, 4 (2006), 447-61.

David C. Kimball, Brady Baybeck, Cassie Gross, and Laura Wiedlocher. 2009. Poll Workers and Election Administration: The View from Local Election Officials. Unpublished manuscript. University of Missouri-St. Louis. http://www.umsl.edu/~kimballd/dk_bb_June09.pdf (accessed March 11, 2014).

Morgan Llewellyn, Steve Schneider, Zhe Xia, Chris Culnane, James Heather, Peter Y.A. Ryan, and Shriramkrishnan Srinivasan. 2013. “Testing Voters’ Understandings of a Security Mechanism used in Verifiable Voting.” *USENIX Journal of Election Technology and Systems* (JETS) 1, 1 (2013), 53-61.

Julia Pomares, Marcelo Leiras, Carolina Tchintian, and Anastasia Peralta Ramos. 2011. Cambios en la Forma de Votar. La Experiencia de Salta con el Voto Electronico. Documentos de Politicas Publicas #94 [Changes in the Manner of Voting. Salta’s Experience with Electronic Voting. Public Policy Documents #94]. Centro de Implementacion de Politicas Publicas para la Equidad y el Crecimiento. Buenos Aires, Argentina.

Guillermo Rosas. 2010. "Trust in elections and the institutional design of electoral authorities: Evidence from Latin America." *Electoral Studies* 29, 1 (2010), 74-90.

Jerrold Rusk. 1970. "The Effect of the Australian Ballot Reform on Split Ticket Voting: 1876-1908." *The American Political Science Review* 64, 4 (1970), 1220-38.

Lee Demetrius Walker. 2005. "The Ballot as a Party-System Switch. The Role of the Australian Ballot in Party-System Change and Development in the USA " *Party Politics* 11, 2 (2005), 217-41.

Alan Ware. 2002. *The American Direct Primary*. Cambridge: Cambridge University Press.

*USENIX Journal of Election Technology and Systems (JETS)*
Volume 2, Number 2 • April 2014
www.usenix.org/jets/issues/0202

# Making Voting Accessible: Designing Digital Ballot Marking for People with Low Literacy and Mild Cognitive Disabilities[1]

KATHRYN SUMMERS, University of Baltimore
DANA CHISNELL, Center for Civic Design
DREW DAVIES, Oxide Design Co.
NOEL ALTON, University of Baltimore
MEGAN MCKEEVER, University of Baltimore

This research began with a question about addressing a broader range of accessibility issues in voting than the standards in the Voluntary Voting System Guidelines (VVSG) require. The VVSG standards cover accessibility for low vision, blindness, and cognitive disabilities. But what if anyone could mark their ballot anywhere, any time, on any device? While the likelihood of voters voting on their own devices may be remote in the current elections environment, it is likely that election jurisdictions will begin to use consumer off the shelf devices as the voter-facing part of voting systems soon. Thus, we narrowed the scope of our research to prototyping an accessible, responsive, Web standards-compliant front end for ballot marking that would be accessible to voters with low literacy (a previously ignored voter audience) or who had mild cognitive disabilities. The final ballot interface is based on principles of "plain language" and "plain interaction." The ballot interface is available under a Creative Commons license at anywhereballot.com. This paper reports on the rapid iterative testing and evaluation (RITE; Medlock et al., 2002) we conducted and the lessons we learned about designing a digital ballot interface for people with low literacy or mild cognitive disabilities.

**Keywords**
Ballot design, universal usability, accessibility, mobile design, low literacy, cognitive impairment, older adults, digital voting

## 1. INTRODUCTION

### 1.1 Why Usability Matters in Election Technology and Ballot Design

Much of the discussion of improved election technology has centered around issues of voting security and efficiency (Dill, Schneier, & Simons, 2003; Gallo, Kawakami, Dahab, Azevedo, Lima, & Araujo, 2010). However, another important priority is usability (Herrnson et al., 2006). According to the Voting Accessibility for the Elderly and Handicapped Act of 1984 and the Help America Vote Act of 2002, voting systems that result in voter error or that are easy to misunderstand can be subject to legal challenge. This is particularly true of voting systems that place an unequal burden on specific members of the population—such as low literacy readers, the elderly, and those with disabilities (United State Election Assistance Commission, 2009; Schur et al., 2002; Ward et al., 2009).

While an increasing number of jurisdictions have moved toward electronic voting systems, relatively little research has been done on ways to make the voter interface broadly usable by groups such as those listed above who may have particular difficulty filling out ballots. Additionally, while voting on mobile devices has promise to increase both efficiency and accessibility for many voters, there has been little research on the usability of electronic interfaces on such devices[2].

---

[1] Funded by the Accessible Voting Technology Initiative of the Information Technology & Innovation Foundation, with funds originally provided by the Elections Assistance Commision.

[2] Campbell et al. (2011) designed a mobile voting system for the iPhone and tested its usability compared to traditional voting platforms. They found that smartphone owners committed fewer errors on the mobile voting system than on

This research represents a step toward establishing principles of usable electronic ballot interface design—starting with the best current thinking about electronic ballot design, and then improving it. As such, it provides both specific research-based guidelines about how to make ballot interfaces more usable, and an illustration of the kind of research and design efforts that are needed to achieve the goal of ballot usability.

### 1.2 Principles of Ballot Design

General principles of ballot design are well understood. The baseline was set in work commissioned by the Election Assistance Commission (EAC) and researched by AIGA's Design For Democracy, which became Effective Designs for the Administration of Federal Elections (EAC, 2009; http://www.eac.gov/assets/1/workflow_staging/Page/70.PDF). This set of best practices for ballot design includes design specifications for many types of printed materials used on Election Day, from signs to ballots. It also proposes a basic framework for a digital user interface. The National Institute of Standards and Technology (NIST) and the EAC have worked since the Help America Vote Act (HAVA) became law in 2002 to establish standards for voting systems. The current version, *Voluntary Voting System Guidelines (VVSG) 2.5*, includes standards on usability and plain language and accessibility practices for people with low vision, who are blind, or who have cognitive disabilities (see vote.nist.gov/publications). These standards have come from NIST-sponsored studies on plain language in instructions and on use of color in user interfaces, reviews of the literature on system accessibility, expert review and opinion, and other basic and applied research (Chisnell et al., 2009; Redish et al, 2009, 2010). These standards do not have force of law, but as recommendations of the commission that is responsible for developing guidance for meeting requirements of the Help America Voting Act of 2002 (HAVA), they represent important ballot design parameters.

However, the VVSG standards assume current technology—mainly a choice between optical scan paper ballots and electronic touch screen interfaces—none of which optimally serves people with disabilities. In contrast, use of smaller mobile devices to mark ballots would allow voters with disabilities to use familiar assistive technology, thus dramatically increasing accessibility and ease of use across a range of audiences.

### 1.3 Designing for Voters with Lower Literacy Skills and Cognitive Disabilities

Research has shown that voters with disabilities are less likely to participate in elections—even those that use traditional voting technology (Schur et al., 2002; Ward et al., 2009)[3]. Moving to an digital format has the potential to increase usability, and thus participation, for such voters—but can also present additional barriers.

Our task was to design a ballot marking interface for people with low literacy and mild cognitive impairments, while incorporating what was already known about accessibility for people with low vision and who are blind. To reach this goal, we incorporated in our initial design research-established principles of what works in Web design for at-risk readers. We conducted formative evaluation sessions with individual adults with low literacy skills and age-related mild cognitive impairments. Research shows that improvements aimed at a particular special population, such as older adults and low-literacy readers, often help other groups as well (Chadwick-Dias, McNulty, & Tullis, 2003; Summers & Summers, 2005). Even expert readers (not surprisingly) tend to prefer information that looks easy to read, gets to the point quickly, and

---

traditional voting systems—a potentially promising outcome for voters using their own familiar assistive devices in digital voting.

[3] Research has not been conducted on voting participation rates of low literacy voters. However, research on other dimensions of the behavior of low literacy users, and anecdotal evidence from those who work with voter participation efforts, suggest that such voters are less likely to participate when they find voting confusing or difficult.

provides plain language explanations of unfamiliar terms and concepts (Nielsen, 1997). Expert readers also sometimes skip over long blocks of text because they feel overloaded with information. So adaptations that increase accessibility for at-risk groups help expert readers as well.

The value of plain language in designing for audiences who do not read well, who are older, who are English language learners, or who are busy or stressed, is well established (Jarrett, et al. 2012; Redish et al, 2009, 2010; Chisnell et al, 2009). From a cognitive perspective, reading involves both decoding and comprehension (Tunmer & Gough, 1986). First the reader "decodes" the text on the page by associating the code—the words—with the concepts they represent. After decoding the individual words, the reader moves on to comprehension: figuring out what the writer was trying to communicate by putting those particular words together. Adults with low literacy skills can struggle with both word recognition and understanding what the words put together actually mean. These reading difficulties can affect voter education, and unavoidably affect the process of marking a ballot and casting the vote.

Additional major challenges in designing digital interactions for lower-literacy audiences, readers with cognitive disabilities, and other at-risk audiences such as the elderly include accommodating their need for sequential processing, guiding interaction effectively, providing clear feedback cues (without text when possible), and helping users avoid, discover, and recover from errors (Summers et al., 2006).

One of the challenges faced by readers with lower literacy skills, some readers who are older, and readers under stress is that their effective field of view is especially narrow. Processing the text itself takes so much cognitive attention that they become less able to pay attention to cues about what might be coming up or remember where they came from. As they move through page content, they are not looking ahead or behind. They are not likely to notice any content above, below, or to the sides of their focus of attention (Summers et al., 2006).

Readers who are using a screen magnifier literally have a narrow field of view. They can only see what the magnifier focuses on. Much of what is around the magnified text is obscured. Readers who must rely on screen readers have an even more narrow field of view. They can only "see" what the screen reader is currently reading aloud. They have no access to information above, below, or to the side.

This narrow field of view has implications for the design and content of a digital ballot. Materials must support sequential processing. Pages must make sense independently. Headings must make sense out of context. Even adjacent paragraphs should be as independent as possible. If readers must remember one paragraph to make sense of the next paragraph, some people who do not read well are likely to come away with misinformation.

Other research has confirmed the value of designs that encourage people to focus on one thing at a time. Allowing for more linear cognitive processing increases the success rate of people experiencing any form of cognitive impairment (Detterman et al., 2000; Salthouse, 1985). Making task demands sequential rather than concurrent also increases the success rate of people under stress (Dutke & Stober, 2001). At the same time, understanding and performance also improve when users can see the structure of the information or task they are performing. The challenge is to provide an overview, or show the structure of the ballot, without losing the specificity and clarity of a "single race at a time" interaction. Even for larger screens, this can create difficulties. With smaller devices, the challenge becomes significantly greater.

## 2. RESEARCH AND DESIGN METHODOLOGY

Our initial design adapted an existing NIST ballot for use with mobile devices, incorporating principles of usability with at-risk readers. We then conducted two rounds of qualitative, formative, iterative one-on-one test sessions using the RITE method (Medlock et al., 2002). The first round consisted of 18 sessions over four days with a paper prototype; the second round consisted of 15 sessions over four days with a digital prototype, for a total of 26 participants and 33 sessions[4]. Changes to the prototypes were made between sessions as needed. Tests were conducted in the University of Baltimore's User Research Lab.

One hallmark of the RITE method is its qualitative, real-time diagnosis of design issues. To accomplish this diagnosis, the moderator interacts with the participant during the session, responding to questions and comments, and giving programmed hints when participants get frustrated and stuck. In the meantime, others from the design team listen and watch to identify the particular design elements that may be causing participants to become frustrated.

We used what we learned in each session to make changes to the ballot interface design for the next session, continuously iterating until no more problems were observed during the final sessions. Team members observing the sessions watched for voter hesitations, discomfort, mistakes, or other indications that the interface was not supporting the voter's intended actions or mental model. When necessary, the moderator asked questions to understand the voter's reactions. We then made changes based on our observations, which we tested in the next session. Analysis between sessions allowed us to come to consensus on observed issues and to generate inferences about why the issues were happening and how we could respond in future design iterations.

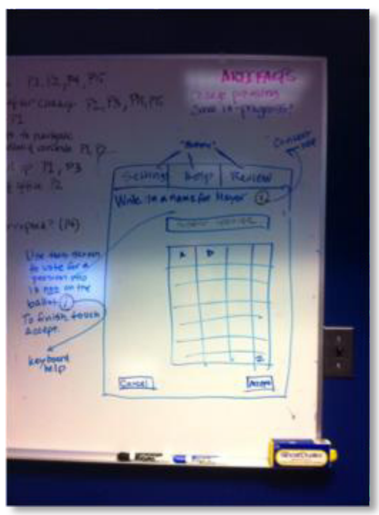

Figure 1. Observed issues were captured between sessions, and the team worked together to develop possible design responses to test in the next session.

While any observation has some impact on user behavior, direct observation has several advantages over the "think aloud" protocol that is often used in traditional usability testing, in which participants give a running commentary describing what they are doing and why (Dumas & Loring, 2008). While having participants "think aloud" can be an effective method to gain insights about an interface's design, it places an additional burden on participants: not only do they have to attempt to complete the assigned tasks, they also have to verbalize their thoughts. Having participants think aloud about what they are doing can also change what they do, thus potentially changing the very behavior that is being studied (Brinkman, 1992; Nagle & Zietlow, 2012). The

[4] Several participants from the paper prototyping sessions were invited back three months later to use the digital prototype. We needed to make sure that specific comprehension and interaction problems faced by these participants had been successfully addressed.

method becomes even more of a problem with participants who have cognitive impairments, such as those that often affect reading (Johnstone, 2006). Additionally, while users with adequate reading skills might be able to explain what parts they had trouble reading and why, those who lack such skills might not even admit they had trouble reading (Parikh, Parker, & Nurss, 1996).

The first round, using a paper prototype of an iPad interface, took place in October 2012. Using paper in the first round of testing rather than programming a digital interface provided a high level of flexibility for trying new designs and responding quickly to observations (Snyder, 2003), allowing us to create and test 16 prototype versions in 18 sessions. The second round, using a digital prototype delivered on an iPad, took place in January 2013, and involved four prototype versions. Most of the pages went through at least three iterations; some pages or sections required as many as four or five iterations before voters were able to proceed without problems.

### 2.1 Participants

Participants included 18 women and 8 men, representing a variety of literacy levels and age levels, and including some participants with mild cognitive impairment. While this may seem like a relatively small number of participants, this number of participants can still lead to reliable conclusions about user behavior related to usability (Virzi 1992; Spool & Schroeder 2001). Key participant characteristics are summarized in Tables 1 and 2.

Table I. Participant Data

| Total Number of participants | | 26 |
|---|---|---|
| Gender | Male | 8 |
| | Female | 18 |
| Age | 18-35 | 7 |
| | 36-70 | 15 |
| | 70+ | 4 |
| Ethnicity | Black | 19 |
| | Caucasian | 5 |
| | Hispanic | 1 |
| | Asian | 1 |
| Voting method in 2012 elections | Abstained | 4 |
| | Touchscreen | 8 |
| | Paper | 14 |
| REALM Score[a] | 0-44 or 6th grade or below reading level | 3 |
| | 45-60 or 7th-8th grade reading level | 6 |
| | 61-66 or 9th grade or higher | 17 |
| Tablet familiarity | Familiar with tablets | 5 |
| | Unfamiliar with tablets | 21 |

[a]The REALM (Rapid Assessment of Adult Literacy in Medicine) is a word recognition test that physicians use to determine whether their patients have low literacy skills and therefore may need additional help with instructions. It is a list of 66 medical words that the moderator asks the participant to read aloud. The number of words the participant pronounces correctly accurately reflects that person's literacy level (Davis, et al., 1991). It is a well-validated measure that is easy to administer, and is often used in non-medical user experience research to recruit participants with low literacy skills (Frascara, 2003; Carstens, 2004; Harper, et al., 2013).

Table II. Participant Self-Reported Disabilities

| | Paper Prototype Session | Digital Prototype Session |
|---|---|---|
| Difficulty Learning | 1 | 2 |
| Seizures | - | 1 |
| Physical Limitations | - | 2 |
| Serious Vision Issues | - | 1[a] |
| Difficulty Working | - | 3 |

[a]The participant with serious vision issues also reported difficulty learning and physical limitations.

We scheduled the majority of participants who were 70 years old or more or who self-reported physical or cognitive limitations, using classifications from the U.S. Census Bureau, for Round Two. We wanted to find and resolve as many potential issues as possible during the paper prototyping in Round One, but we were sensitive to the fact that moving to a digital prototype on a potentially new device could possibly introduce new interaction issues for our most "at-risk" participants. So we made sure that our most vulnerable participants interacted with the most mature version of the ballot that we could manage, and in the most realistic format. We also made sure that we included participants who were unfamiliar with tablets, as well as participants who were familiar with tablets. We wanted to make sure that we did not create barriers to voting for those who were not already familiar with tablets—particularly for elderly voters who might be uncomfortable with new technology.

### 2.2 Materials

Participants used a version of the NIST medium complexity ballot, which uses realistic but fictional names and contests, with colors for party names. The ballot had 18 pages, with 11 contests, three judge retention contests, one constitutional amendment, and two ballot measures.

We started with the visual designs and interaction designs developed by the work of Design for Democracy in *Effective Designs for the Administration of Federal Elections*, a project involving hundreds of different types of voters working with dozens of design iterations and variations. That project delivered the most widely used template for paper optical scan ballots, but it also generated a hypothetical design for an electronic ballot interface that was remarkably compatible with the iOS interface (which it had pre-dated by two years) and which provided many of the visual elements that worked well in our interface.

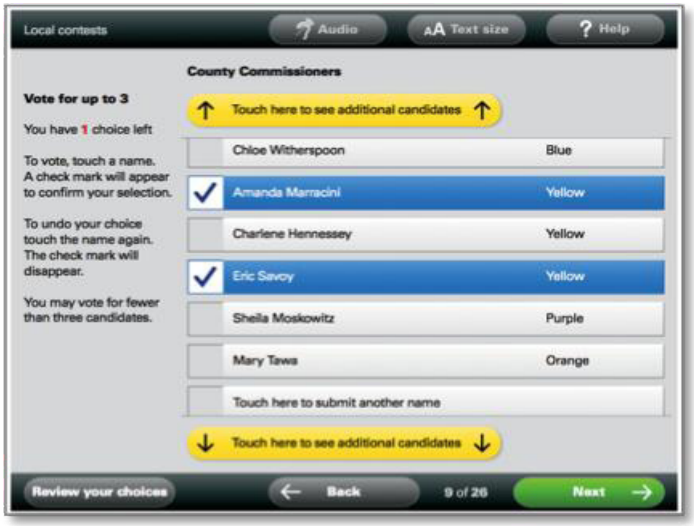


Figure 2. A contest from Design for Democracy's hypothetical electronic ballot interface

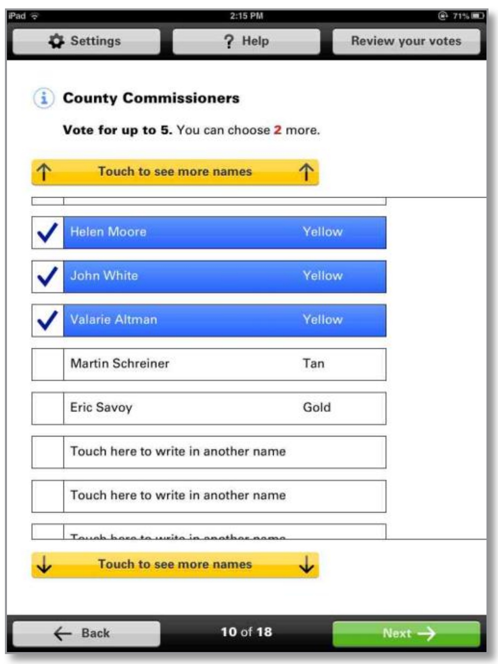


Figure 3. The same contest in the final version of our ballot. Instruction text is minimized and integrated into the path of voter action and attention, rather than being placed in a separate column.

### 2.3 Test Tasks

Participant tasks included simple selection of candidates, straight party voting, writing in candidates, reviewing votes, and changing some votes before casting the ballot. For some of the contests, participants made selections without explicit direction. Other selection tasks were party-based or candidate-based. The tasks and the ballot were drawn from NIST research on the language of instructions on ballots (Chisnell et al., 2009; Redish et al, 2009, 2010).

The tasks were originally designed for a performance test in which the researchers investigated whether plain language instructions helped voters make fewer mistakes than conventional instructions. We were conducting a formative study rather than a performance test, but the maturity of the test protocol ensured that we tested a wide range of voting behavior, and using the same tasks and slate anchored our rapid iterations. We did not measure performance because we changed aspects of the design between many of the sessions.

### 2.4 Procedure

Sessions were held one-on-one, in a physical arrangement meant to loosely mimic voting at home. Participants sat in a comfortable chair, using the "tablet" on their lap. For the paper prototype session, two video cameras recorded the session: one camera suspended over the back of the chair to show the prototype (main frame), and one camera across from the participant to capture face and body language (picture-in-picture). For the digital prototype session, the contents of the iPad screen were echoed on the recording computer using the Reflector application (main frame), and a camera was placed across from the participant to capture face and body language (picture-in-picture).

Because this was iterative, formative testing, the moderator interacted extensively with the participants, restating tasks and asking follow-up questions when needed. The rest of the research team observed through a one-way mirror, and saw a close-up video feed of participant actions. Between each session, team members captured issues, generated inferences, and agreed upon design changes, which were generally implemented between sessions. Larger design changes were sometimes implemented prior to the next day of sessions. Successful design changes were retained; other design changes were refined or replaced until the issue was resolved.

For the paper prototype, participants interacted with paper pages of the prototype held in a simple foam-core frame; a human "computer" added or removed pages in response to the participant's actions. The digital prototype was coded in HTML5 and was hosted on a private server, but behaved like an iPad application rather than being delivered in a browser.

## 3. RESULTS

The formative testing and iterative design resulted in a final ballot interface that addressed the specific problems that had been observed in earlier participant sessions. While the sample sizes were too small (and revisions too frequent) to make for meaningful quantitative comparison, nonetheless there was a noticeable increase in ease and accuracy of use, particularly among the most at-risk participants, in later iterations of the design.

While all screens went through multiple iterations, some changes had a bigger effect on the overall accessibility of the ballot for participants. The result was an interface that was noticeably easier to use, and in which all of the major observed points of difficulty from earlier versions had been successfully eliminated.

Elements of the ballot interface design that underwent significant testing and revision included the following:

| | |
|---|---|
| Instructions to voters—general | Changes to the voter instructions based on participant behavior during testing consisted primarily of the following:<br>• Fewer words<br>• Simpler language, with key phrases bolded to support skimming<br>• Changing the order of the instruction on recovering from a mistake to refocus on changing with less focus on "fault."<br>Original version: "**If you make a mistake or you want to change a vote,** first touch the blue choice you no longer want. It turns white again. Then touch the choice you do want."<br>Final version: "**If you want to change your vote or if you make a mistake,** first... [same as original]" |
| Straight party voting (for states where this is relevant) | • Instructions were revised to have minimal words and simple language.<br>• Icons leading to supplemental voter education content were first moved and made less visually prominent, and then ultimately removed. Voters kept tapping these information icons when they wanted to mark their vote, and voters with mild cognitive impairment then got confused and stuck in the supplemental content. |
| "Vote for one" contests (national, state, and local) | Instructions were reduced from nine lines to a single line, integrated into the vertical layout.<br>Final version: "**Vote for 1**." |
| Multi-candidate contests | • Instructions were reduced from 12 lines to a single line.<br>• Wording was changed to sound less directive and to avoid the word "choices" (see later discussion).<br>Earlier version: "**Vote for up to 5**. You have **2** choices left."<br>Final version: "**Vote for up to 5**. You can choose **2** more."<br>• Researchers carefully observed scrolling behaviors in order to support all methods that participants tried to use.<br>– For screenreaders, the full list of candidate names was read without requiring any scrolling.<br>– For visual display, voters could scroll by touching buttons at the top and bottom of the visible candidate names that say "Touch to see more names."<br>– Voters could also scroll using the visible scroll bar, or on the iPad by flicking with a finger. |
| Judge retention contests | We reduced the instructions to one line, and phrased the contest as a question with simple language.<br>Final version: "Keep **Esther York** as Supreme Court Chief Judge?<br>"**Choose Yes or No**" |

| | |
|---|---|
| Referenda and ballot measures | Moving to a vertical layout (to prepare for responsive design on mobile phones as well as tablets) made the biggest difference here, allowing more of the text to appear on screen. We also reduced the instructions to one line.<br>Final version for referenda: "**Choose For or Against**"<br>Final version for ballot questions: "**Choose Yes or No**" |
| Write-in candidate interactions | Instructions were reduced from 19 lines to two, focusing on the essential message that voters not write in the name of someone already on the ballot. Additional instructions were available if voters wanted them. The size of the letters in the on-screen keyboard was also increased.<br>Final version: "Use this screen to vote for a person who is **not** on the ballot.<br>"To finish, touch **Accept**. If you change your mind, touch **Cancel**." |
| Changing a vote interactions | A key feature of the interaction on the ballot forces voters to deselect their original choices before they can change their vote. With an interaction as important as voting, we wanted to make sure that changes were intentional rather than accidental. But we wanted voters to be able to recover quickly from this somewhat non-standard requirement.<br>In the final version:<br>• The "Close" button was made the same color as the action buttons in the rest of the ballot.<br>• The wording was simplified, and the most important information was bolded to make it easy to find for voters with low literacy skills.<br>• Voters could also dismiss the error message by touching the screen anywhere outside the error message. |

| | |
|---|---|
| Review votes screen | This screen caused the most problems. Voters were confused, made mistakes, and consistently got lost and confused by having to re-navigate the ballot.<br>• Adjusting spacing and leading between titles and names in the contests solved some issues.<br>• Language was refined. In particular, the word "choices" was eliminated, since this made some voters think they had choices still to make (see additional discussion below).<br>• We refined the message about undervoting to be more clear but less directive.<br>• We changed the interaction pattern. In the first version, voters who decided to change a vote from the Review screen were taken back to the ballot, starting at the contest they had wanted to change. They could either continue through the ballot a second time, one contest at a time, or use a button at the top to return to the Review screen. We changed this to an "out and back" interaction that took voters to the contest to be changed, then back to the Review screen.<br>Letting the Review screen ground their ballot changes was more successful, and voters were able to make the changes they wanted then proceed to casting. |
| Casting the vote | Our early rounds of testing showed participants becoming somewhat anxious and backing away from casting their votes, even though they did not have actual changes they wished to make to their ballots. Revising the text on the confirmation screen ameliorated this anxiety and helped participants feel more confident.<br>Original version: "**Are you sure you have finished voting?**"<br>"**Note**: Once you press the Vote button, you will not be able to make any more changes.<br>"If you want to make changes, touch the **Return to ballot** button.<br>"If you are ready to cast your ballot, touch the **Vote** button."<br>Final version: "**Are you finished?**<br>"If you want to make changes, touch the **Return to ballot** button.<br>"If you are ready to cast your ballot, touch the **Vote** button." |

| | |
|---|---|
| Supplemental content (Help screen for using the interface, supplemental information about contests and candidates) | Our early versions included three types of supplemental content: help information for using the interface, basic civics information about offices and contests, and simple voter education materials about candidates.<br>However, in the testing we found that the voter education materials caused voters to lose track of their voting activity, sometimes to the point of needing help to restart the voting process. Voters also persistently tried to use the information icons next to candidate names to mark their votes. We tried to eliminate these problems through moving the icons, changing their visual treatment, and simplifying the voter information to make it easier to process.<br>In the final version:<br>• Help information for the interface can be accessed through a button at the top, and through an information icon next to the voting instructions on each page.<br>• Information about offices and contests can be accessed through an information icon next to the contest name. The information icons went through multiple iterations to make them less visually intrusive but still findable.<br>• Eventually we reluctantly eliminated the voter education materials. The process of marking and casting the ballot is essential for democracy, and the benefit for "at risk" voters of eliminating disruptions that might prevent voting outweighed the benefits of supporting additional research about candidates. |

Most of these design revisions reflected changes towards plainer, simpler language, and plainer, simpler interaction: a style we have summed up as plain language and plain interaction[5]. More information about these findings and the design principles they suggest is provided below. Additionally, the finalized ballot interface is available for use under a Creative Commons license at anywhereballot.com.

## 4. LESSONS LEARNED FOR DESIGNING ACCESSIBLE BALLOT INTERFACES

Based on our observations and iterative testing of multiple interface revisions, we have articulated the following broad principles for design of digital ballot interfaces—with reference to established findings from the previous research literature related to usability for low literacy readers, the elderly, and those with disabilities.

[5] *Plain language* has been defined as "language that avoids obscurity, inflated vocabulary and convoluted sentence construction" (Eagleson), and as "language that reflects the interests and needs of the reader and consumer rather than the legal, bureaucratic, or technological interests of the writer or of the organization that the writer represents" (Steinberg, 1991).

We define *plain interaction* as interaction designed to help the user clearly understand the information presented to them, the action or actions expected of them, and the results of their actions. Guidelines that help achieve plain interaction are discussed in this paper, including supporting immediate action, creating a linear flow, and supporting a narrow field of view and staying in the user's current locus of attention. We are indebted to Shaun Kanefor the term "plain interaction," which he used in a phone conversation with our team to sum up the interaction design findings we were sharing from our study.

### 4.1 Plain Language: Test Language to Ensure Usability

Every ballot needs instructions. But a digital ballot also needs instructions for using its interface, as well as button labels and messages that paper ballots don't need. Redish et al. (2009) tested conventional instructions for ballots on a touchscreen tablet voting system, and found that voters performed better with plain-language instructions, as opposed to instructions that included technical voting terminology. At the end of their report, they proposed a new set of ballot instructions based on their research. This work by Redish et al. (2009) provided valuable guidance, but our testing with participants who had low literacy skills, or mild cognitive impairment, or who were seniors unfamiliar with touchscreen technology, demonstrated that instructions can sometimes be too complete. Designers may try to accommodate vulnerable voters by adding more instructions. But if the system and the ballot design are tightly integrated, and the interface design of the ballot is simple and plain, fewer instructions are needed. Keeping instructions minimal allows voters to focus their attention and effort on the tasks of voting, rather than on the interface or the interaction.

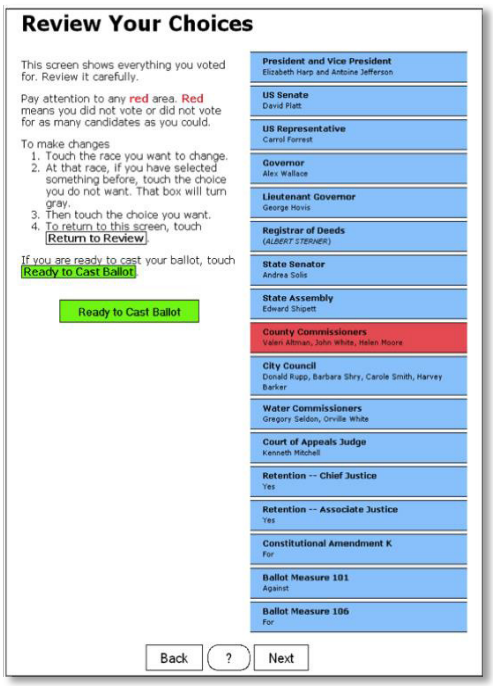


Figure 4. Redish, et al. (2009) recommended thorough and complete instructions in plain language.

We started with simple instructions, but found that we needed to simplify even further. Based on prior research, we knew that participants with lower literacy needed simple, familiar words, and that they would not be familiar with election jargon (Doak et al., 1996; Redish et al., 2009). So our initial instructions were written to be very simple, with short sentences. Vocabulary use was consistent, so that we did not use multiple terms to refer to the same concept. The most helpful instructions were those that used positive, specific, prescriptive wording, and that were designed for immediate action.

Despite our best efforts, however, some election jargon made its way into the instructions, causing participants to hesitate and stumble over the words. Thus, "Touch to see more candidates" and "Touch here to write in a candidate" needed to become "Touch to see more names" and "Touch here to write in another name."

As other researchers have found, people with low literacy skills tend to take words very literally and to act on every word (Doak, Doak, & Root, 1996; Summers et al, 2006). If the instructions said to touch the screen, they touched the screen, even if they were temporarily on a help screen rather than the ballot page. If the instructions said, "You have two choices left," participants thought they were obligated to make two more selections, thus potentially changing their voting behavior.

Sometimes small language tweaks have a very big impact. In the first version of the ballot interface, the word "choices" appeared throughout. Participants had a hard time with the word "choices" due to the diphthong "oi," often misreading this word as "choose" or "chose" (Burrows & Lourie, 1963; Gleason & Ratner, 2009). As a result, multi-candidate voting posed a problem for participants: not only was there too much wording in the instructions, but participants also had a hard time understanding how many candidates *could* be voted for and how many they had already voted for. Subsequent iterations simplified the language, leading to improved participant success.

On the Review screen, this difficulty was compounded by the fact that for most of the ballot, the word "choices" meant "options." As a result, when the review screen instructions (and the navigation button to go to the Review screen) said, "Review your choices," participants thought they had to make *more* choices on the Review screen. The concept of reviewing decisions that were already made was lost, and the page became a source of confusion and delay.

Our solution was to eliminate the word "choice" throughout the ballot.

"You have 2 choices left" became "You can choose 2 more." This revision eliminated the difficult diphthong in "choice" and emphasized opportunity rather than obligation, so participants felt free to undervote if they wished. The navigation button to go to the Review screen became "Review your votes," and the instruction on the Review screen was revised to read, "This screen shows everything you voted for. Review it carefully. If you are ready to cast your ballot, touch Cast your vote." With this version, participants were more successful with reviewing their ballot and moving forward to casting their vote.

Similarly, the instructions for contests allowing multiple votes and the undervote message on the Review screen for these contests made some participants feel obligated to avoid undervoting. For the Review screen, we replaced "You could have voted for 5 candidates, but you only voted for 2" with "You voted for 3 people. You can vote for 2 more." The revised version moves from the familiar to the new—what is known to what is unknown—and sounds more optional.

### 4.2 Plain Language: Make It Look Easy to Read

To keep instructions and actions tightly integrated, we eliminated all but the most necessary instructions. We also made the typesize large and inviting.

Because reading is such a demanding cognitive activity for those with low literacy, we needed to minimize the amount of reading participants had to do. We did not want to divert cognitive resources from the process of voting to the task of using the ballot interface. When text needed to be longer than a sentence, we bolded key phrases to guide those who might be intimidated by full paragraphs to the most essential information, as shown below:

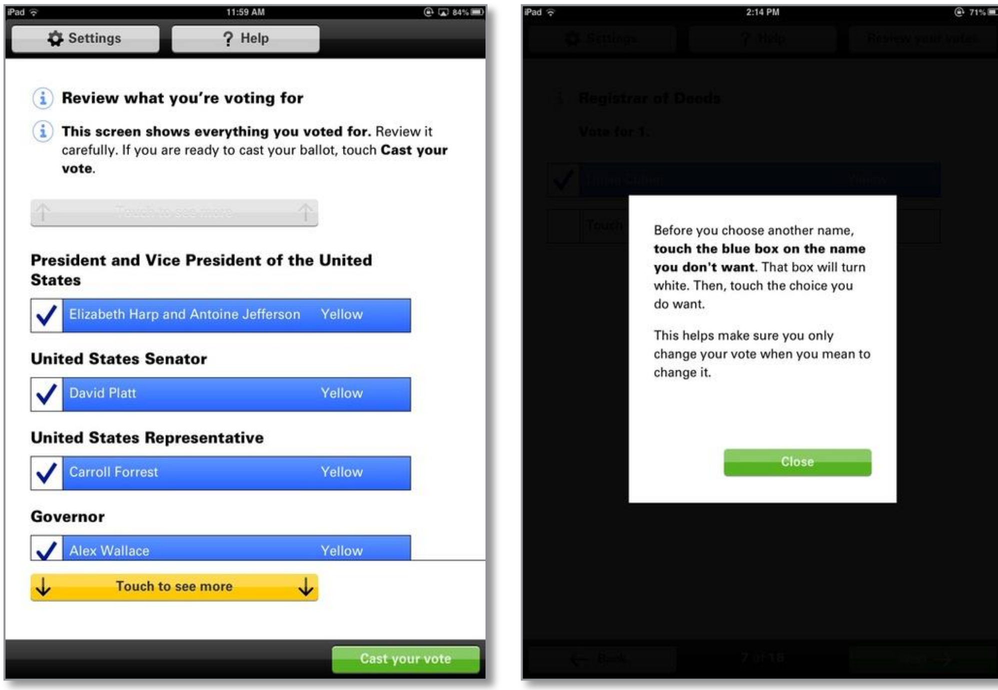


Figure 5. Bolding key phrases made long instructions easier to process

We also identified ways to reduce participant anxiety during ballot marking. Anxiety can affect comprehension and willingness to move forward, as well as comfort. The original text on the "Cast Your Vote" screen inspired fear of making an error, and participants backed away from casting their vote and returned to the ballot—sometimes more than once. Revising this message reduced anxiety and allowed voters to cast their vote successfully.

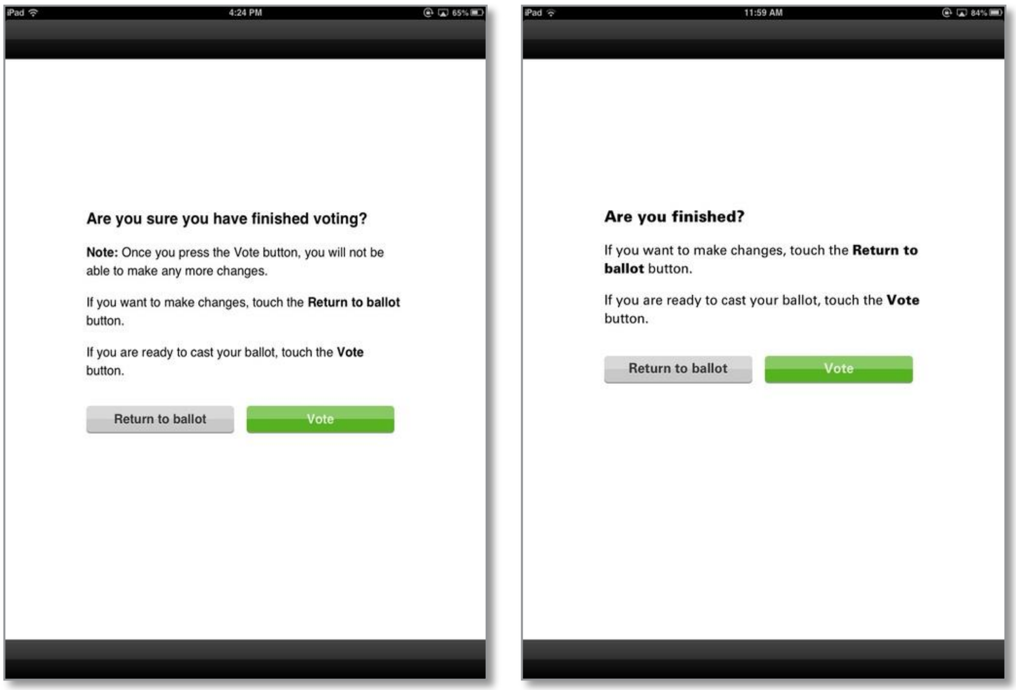


Figure 6. The early version of the confirmation screen on the left triggered anxiety for some voters. They responded as if it were an error message, and worried that they had somehow made a mistake. The final version on the right reassured voters that they were proceeding as planned, but allowed them to return to the ballot if they still wanted to make changes.

### 4.3 Plain Interaction: Support Users' Preferred Actions

We also watched to see how users expected to interact with the interface, so we could make these natural user behaviors successful if possible. For voters who are familiar and comfortable with a gestural, touch interface, we supported dragging the scrollbar and flicking a finger near the content area. But not everyone knows how to use a gestural or touch interface, so we included buttons for scrolling and specific interaction for selecting. Likewise, having reasonably large button targets that didn't rely on subtle physical motion helped voters with dexterity or coordination problems.

The final interface supports multiple behaviors, so that users can scroll by using a button, using a scrollbar, or flicking a finger. Users can select a name by touching anywhere on the box containing the name; they can close a message box using the CLOSE button or by touching anywhere outside the box.

We also had to reduce distractions by eliminating some kinds of interactions. We initially thought voters might benefit from layering in supplemental voter education content about the candidates or measures. But the extra icons on the interface created a strong call to action for voters with mild cognitive impairment: these voters repeatedly tried to select names by tapping the information icons. We tried various design changes to reduce this call to action, but ultimately we had to remove this supplemental content[6].

[6] Providing supplemental information about candidates could possibly be valuable for voters. However, this kind of supplemental content is not currently supported by any voting interface, and could potentially raise issues of biasing the act of voting. Thus, it seemed unwise to jeopardize the activity of ballot marking by adding supplemental content that was itself fraught with potential and mostly unstudied complications.

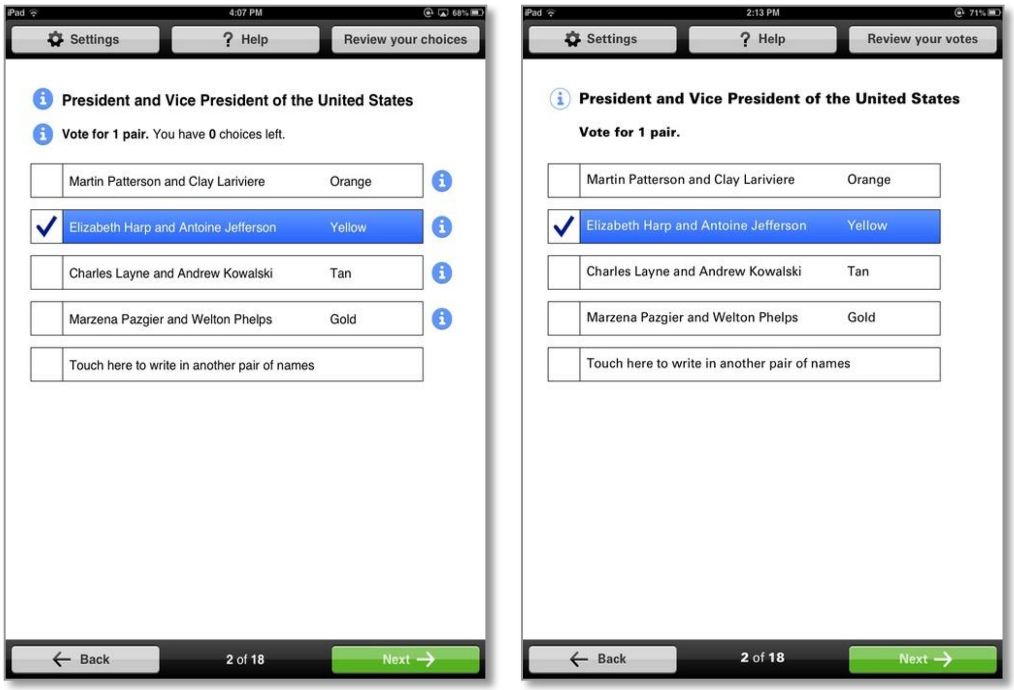


Figure 7. Ballot before and after removing extra icons

Other icons leading to supplemental content remain in the interface, because some voters with low education are missing basic civics knowledge (Redish, et al., 2009; Center for Information & Research, 2010). For example, we saw voters confuse U.S. Senator with State Senator, or not know what a water commissioner does. We moved these icons slightly farther away from the action area of marking ballot choices and reduced their visual impact, so they were available as support when needed but did not interfere with ballot marking.

### 4.4 Plain Interaction: Support Immediate Action

Our low-literacy participants tended to act on every word. This meant that instructions needed to be integrated with the interface so that instructions were designed to be read and followed right away, in the same location (see also Summers et al., 2008).

Our initial design had a landscape orientation, with instructions on the left and the action area for marking choices on the right. Many electronic ballots follow this format. Our final design has a portrait orientation, with instructions minimized and integrated into the action area. This design allowed for a tighter coupling between reading and action, while simultaneously preparing for future development of the ballot for smaller screens.


Figure 8. The first iteration of the ballot interface, based on the work from Design for Democracy.

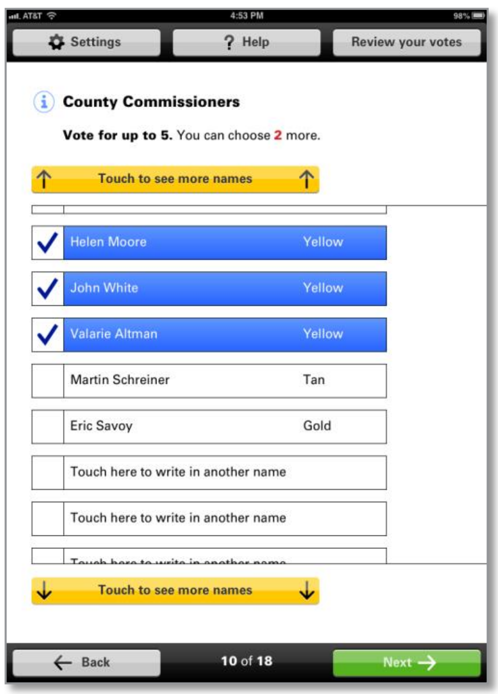


Figure 9. The final version of the same contest.

During testing, we also discovered that including images on instruction screens had a negative effect in the context of a tablet: participants tried to use these images to make changes to the ballot. They assumed these images would be interactive, and were frustrated when they were not.

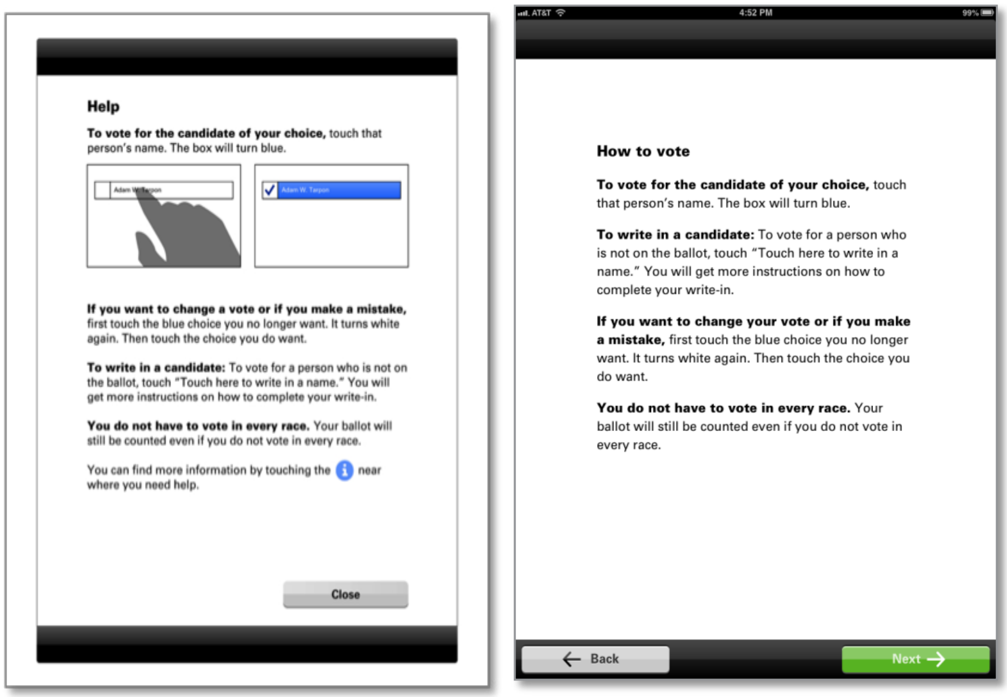


Figure 10. Initial version and final version of the Help screen. Voters expected to be able to interact with visual elements and were frustrated if they could not. Instruction and help pages were more successful when images were removed, text was minimized, and key messages were visually emphasized.

### 4.5 Plain Interaction: Create a Linear Flow

Participants did not always have a clear understanding of the hierarchy of government, which sometimes made navigating the ballot more difficult. This problem was most apparent when they tried to move around in the ballot after leaving the "Review Your Votes" screen. This problem was resolved by making each visit to a ballot screen initiated from the Review screen into an "out and back" interaction: the navigation buttons on the ballot screens were replaced with a single button that returned users to the Review screen[7]. The structure provided by the Review screen itself allowed participants to find their way among races to revise their selections (Summers, et al., 2014).

[7] The superior usability of "out and back" interactions for users with low literacy was confirmed by the Summers et al. (2005) study on navigational behaviors of users with low literacy skills. It has also been confirmed in recent testing with 34 participants (low literacy, seniors, blind, low vision, low dexterity, and hearing-impaired) performed for the Maryland State Board of Elections (Summers, et al., 2014).

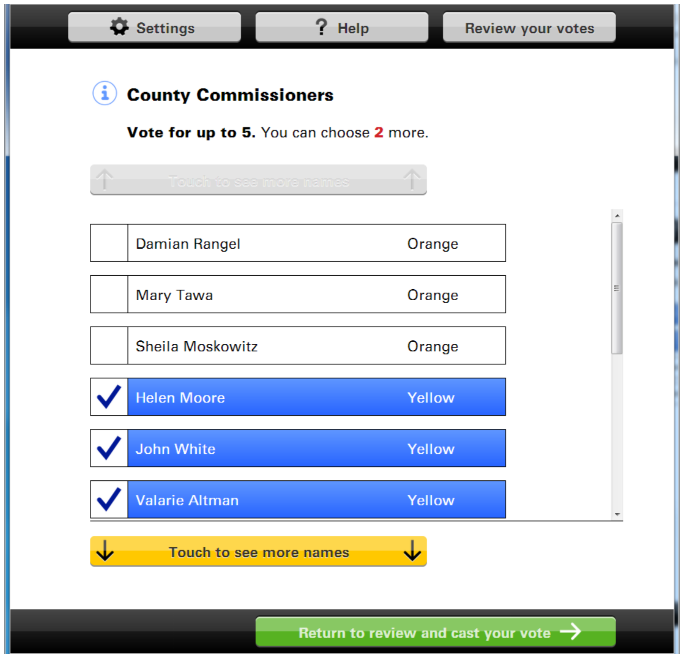


Figure 11: An example of an "out and back" visit to change votes for county commissioners.

## 5. DISCUSSION

### 5.1 Moving Toward the Goal of Using Voters' Own Assistive Technology

One of the goals of our research was to develop a ballot marking interface that would allow voting on voters' own devices. The creation of an easily usable interface format using current Web standards represents an important step forward. Informal quality assurance testing of the developed interface using common assistive devices is in process. Before this interface is used in actual elections, more exhaustive user testing should be conducted on how well it works with a wide variety of commonly used assistive devices.

### 5.2 Creating Mobile Access to Citizenship Activities

While voting using mobile devices raises substantial concerns, particularly with respect to voting security, it poses substantial potential benefits as well. Current patterns and trends in mobile device adoption and use indicate that mobile devices represent an increasingly important resource for many citizens (Smith, 2010; Zickuhr & Smith, 2012; Zickuhr & Madden, 2012). Particularly noteworthy is the use of mobile devices by African Americans and Hispanics, who tend to use such devices more extensively and for a broader range of activities than white Americans (Horrigan, 2009; Zickuhr & Smith, 2012). Younger adults are also highly likely to use mobile devices (Lenhart et al., 2010). In short, creation of a ballot interface that supports mobile devices has the potential to substantially expand voting, particularly among groups that are historically less likely to vote.

## 6. CONCLUSION

We have a vision of every eligible adult being able to vote independently and successfully. Although we are very proud of our final ballot marking interface, which takes long strides towards a universally usable ballot marking interface, there is further work to do. We'd like to do additional testing with participants with short-term memory loss and aphasia and with participants who use screen readers and other assistive devices. (Informal "friends and family" testing with a few blind users trying the interface with iOS accessibility features and the JAWS screen reader show that our ballot design has real promise but needs more testing.) We still need to establish settings for font size, contrast, language, etc., so that voters can personalize the interface. We need to pilot the ballot in a real election with real choices, which we hope to do soon. And in the context of a real election, we need to deepen our understanding of whether the interface is successful in encouraging voters to review the summary of their ballot choices thoroughly before casting their vote.

There are also many system architecture issues to deal with in voting that were not addressed in this project—chief among them security issues. When security elements are added, they must be designed and tested with universal usability in mind, so that the security elements also embody principles of plain language and plain interaction. For example, the voting system must support pausing and resuming the ballot marking process to support the needs of those with traumatic brain injuries for rest breaks.

Moreover, by making the activity of marking one's ballot possible, usable, and accessible on any mobile device, we make voting much more available not only to people with disabilities, but also for people who want to vote but who may face a variety of convenience barriers or comfort barriers. By developing a portable, technology-independent ballot front end that will work for people with low literacy and cognitive issues, we anticipate a direction that many election jurisdictions are already pursuing.

## ACKNOWLEDGEMENTS

We thank the Information Technology & Innovation Foundation for the grant that made this research possible, the University of Baltimore for providing lab space for our testing and logistical support for our grant, and Whitney Quesenberry for inspiring the project and shepherding it to fruition.

## REFERENCES

J. A. Brinkman. 1992. Methodological Problems When Determining Verbal Protocol Accuracy Empirically. *Travail Humain, 57*(2), 97-110.

A. T. Burrows, & Z. Lourie. 1963. When "Two Vowels Go Walking." In *The Reading Teacher*, 17(2), 79-82.

B. A. Campbell, C. Tossell, M. D. Byrne, & P. Kortum. 2011. Voting on a Smartphone: Evaluating the Usability of an Optimized Voting System for Handheld Mobile Devices. *In Proceedings of the Human Factors and Ergonomics Society 55th Annual Meeting* (pp. 1100-1104). Las Vegas, NV: Human Factors and Ergonomics Society.

A. Carstens. 2004. Tailoring Print Materials to Match Literacy Levels: A Challenge for Document Designers and Practitioners in Adult Literacy. *In Language Matters: Studies in Languages of Africa*, *35*(2), 459-484.

The Center for Information & Research on Civic Learning and Engagement, Tufts University. 2010. High School Civic Education Linked to Voting Participation and Political Knowledge, No Effect on Partisanship or Candidate Selection. Retrieved Jan. 29, 2014 from http://www.civicyouth.org/high-school-civic-education-linked-to-voting-participation-and-political-knowledge-no-effect-on-partisanship-or-candidate-selection/. .

A. Chadwick-Dias, M. McNulty, & T. Tullis. 2003. Web Usability and Age: How Design Changes Can Improve Performance. In *Proceedings of the 2003 Conference on Universal Usability*, 30-37. doi:10.1145/957205.957212

D. Chisnell, S. Becker, S. Laskowski, & S. Lowry. 2009. Style Guide for Voting System Documentation: Why User-Centered Documentation Matters to Voting Security. In *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections* (pp. 2-2). USENIX Association

D. Dill, B. Schneier, & B. Simons. 2003. Voting and Technology: Who Gets to Count Your Vote?. *Communications Of The ACM, 46*(8), 29-31.

T. Davis, M. Crouch, S. Long, R. Jackson, P. Bates, R. George. 1991. Rapid Assessment of Literacy Levels of Adult Primary Care Patients. *Family Medicine, 23*(6), 433-435.

D. K. Detterman, L. T. Gabriel, & J. M. Ruthsatz. 2000. Intelligence and Mental Retardation. In R. J. Sternberg (Ed.), *Handbook of Intelligence*. Cambridge: Cambridge University Press, 141-158.

C. Doak, L. Doak, & J. Root. 1996. *Teaching Patients with Low Literacy Skills* (2nd ed.). Philadelphia: J. B. Lippincott

Company.
J. Dumas, & B. Loring. 2008. *Moderating Usability Tests.* United States: Morgan Kaufmann.
S. Dutke, & J. Stöber. 2001. Test Anxiety, Working Memory, and Cognitive Performance: Supportive Effects of Sequential Demands. *Cognition and Emotion*, 15, 381-389.
J. Frascara. 2003. *Design and Social Sciences: Making Connections*. London: Taylor and Francis.
K. Fuglerud, & T. Rossvoll. 2012. An Evaluation of Web-Based Voting Usability and Accessibility. *Universal Access in The Information Society, 11*(4), 359-373.
R. Gallo, H. Kawakami, R. Dahab, R. Azevedo, S. Lima, & G. Araujo. (2010). T-Dre: A Hardware Trusted Computing Base for Direct Recording Electronic Vote Machines. In P*roceedings of the 26th Annual Computer Security Applications Conference*, Austin, Texas. doi:10.1145/1920261.1920291
J. B. Gleason, & N.B. Ratner. 2009. The Development of Language (7th Ed). Boston: Pearson.
L. Harper, M. McMacken, L. Appelt, & K. Summers. 2013. The Importance of Choice Design for Low Literate User Experience. In *Communications in Computer and Information Science, 374*, 430-434.
Help America Vote Act (2004). Retrieved from http://www.eac.gov/assets/1/workflow_staging/Page/41.PDF
P.S. Herrnson, R.G. Niemi, M.J. Hammer, B.B. Bederson, F.G. Conrad, & M. Traugott. 2006. The Importance of Usability Testing of Voting Systems. In *Proceedings of the 2006 USENIX/ACCURATE Electronic Voting Technology Workshop*, Vancouver, B.C., Canada.
J. Horrigan. 2009. *Wireless Internet Use*. Washington, DC: Pew Internet & American Life Project. Retrieved from http://www.pewinternet.org/Reports/2009/12-Wireless-Internet-Use.aspx
C. Jarrett, J. Redish, & K. Summers. 2012. Designing for People Who Do Not Read Easily. In L. Meloncon (Ed.), *Rhetorical Accessibility: At the Intersection of Technical Communication and Disability Studies*. New York: Baywood.
C.J.M. Johnstone. 2006. Using the Think Aloud Method (Cognitive Labs) to Evaluate Test Design for Students with Disabilities and English Language Learners. University of Minnesota, National Center on Educational Outcomes. Minneapolis, MN: University of Minnesota.
A. Lenhart, R. Ling, S. Campbell, & K. Purcell. 2010. *Teens and Mobile Phones*. Washington, DC: Pew Research Center's Internet & American Life Project. Retrieved from http://pewinternet.org/Reports/2010/Teens-and-Mobile-Phones/Summary-of-findings/Findings.aspx
M. C. Medlock, D. Wixon, M. Terrano, R. Romero, & B. Fulton. 2002. Using the RITE Method to Improve Products: A Definition and a Case Study. *In Proceedings of the Usability Professionals Association*.
M. Nagle, & E. Zietlow. 2012. An Exploratory Comparison of Two Website Usability Testing Methods: Concurrent Think-Aloud and Eye-Tracking with Retrospective Think-Aloud. (Master's Thesis). University of Baltimore, Baltimore, MD.
J. Nielsen. 1997. Be Succinct! (Writing for the Web). Retrieved from http://www.nngroup.com/articles/be-succinct-writing-for-the-web/
N.S. Parikh, R.M. Parker, & J.R. Nurss. 1996. Shame and Health Literacy: The Unspoken Connection. *Patient Education and Counseling, 27*(1), 33-39.
J.G. Redish, D.E. Chisnell, E. Newby, S.J. Laskowski, & S. Lowry. 2009. Report of Findings: Use of Language in Ballot Instructions. Retrieved from http://www.nist.gov/itl/vote/upload/NISTIR-7556.pdf
J.G. Redish, D.E. Chisnell, S.J. Laskowski, & S. Lowry. 2010. Plain Language Makes a Difference When People Vote. *Journal of Usability Studies*, *5*(3), 81-103.
T. Salthouse. 1985. *A Theory of Cognitive Aging*. Amsterdam: North Holland.
B. Shneiderman. 2000. Universal Usability. Communications of the ACM, 43(5), 84-91.
L. Schur, T. Shields, D. Kruse, & K. Schriner. 2002. Enabling Democracy: Disability and Voter Turnout. *Political Research Quarterly, 55*(1), 167-190.
A. Smith. 2010. *Mobile Access 2010*. Washington, DC: Pew Internet & American Life Project. Retrieved from http://www.pewinternet.org/Reports/2010/Mobile-Access-2010.aspx
C. Snyder. 2003. Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces. New York: Morgan Kaufmann.
J. Spool, & W. Schroeder. 2001. Testing Web Sites: Five Users is Nowhere Near Enough. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems* (pp. 285-286). ACM.
K. Summers, & M. Summers. 2005. Reading and Navigational Strategies of Web Users with Lower Literacy Skills. In *Proceedings of the American Society for Information Science and Technology*, *42*(1). doi:10.1002/meet.1450420179
K. Summers, M. Summers, & C. Helm. 2008. *PAP Application Usability and Lessons Learned*. Center for Business Intelligence Patient Assistance Programs, Baltimore, MD.
K. Summers, J. Langford, J. Wu, C. Abela, & R. Souza. (2006). Designing Web-Based Forms for Users with Lower Literacy Skills. In *Proceedings of the American Society for Information Science and Technology, 43*(1), 1-12. doi:10.1002/meet.14504301174
K. Summers, C. Rinn, E. Rhodes, N. Alton, & J. Lee. 2014, Jan. 13. Maryland Online Ballot Marking Tool Usability Testing Report. Submitted to the Maryland State Board of Elections. Retrieved January 29, 2014 from http://www.elections.maryland.gov/press_room/documents/OnlineBallot_UsabilityTestResults.pdf.
M.F. Theofanos, & J. G. Redish. 2003. Guidelines for Accessible and Usable Web Sites: Observing Users Who Work with

Screen Readers. *Interactions*, *10*(6), 38-51.

W.E. Tunmer, & P.B. Gough. 1986. Decoding, Reading, and Reading Disability. *Remedial and Special Education,* 7, 6-10.

United States Election Assistance Commission. 2007. Effective Designs for the Administration of Federal Elections. Retrieved from http://www.eac.gov/election_management_resources/designing_polling_place_materials.aspx

United States Election Assistance Commission. 2009. Voluntary Voting Systems Guidelines (v 1.1). Retrieved from http://www.eac.gov/testing_and_certification/voluntary_voting_system_guidelines.aspx

A. Ward, P. Baker, N. Moon. 2009. Ensuring the Enfranchisement of People with Disabilities. *Journal of Disability Policy Studies, 20*(2), 79-92.

R.A. Virzi. 1992. Refining the Test Phase of Usability Evaluation: How Many Subjects is Enough?. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 34*(4), 457-468.

Voting Accessibility for the Elderly and Handicapped Act of 1984. Retrieved from http://codi.tamucc.edu/archives/legislation/voting.access.htm

K. Zickuhr, & M. Madden. 2012. *Older Adults and Internet Use*. Washington, DC: Pew Research Center's Internet & American Life Project. Retrieved from http://www.pewinternet.org/~/media//Files/Reports/2012/PIP_Older_adults_and_internet_use.pdf

K. Zickuhr, & A. Smith. 2012. *Digital Differences*. Washington, DC: Pew Internet & American Life Project. Retrieved from http://pewinternet.org/Reports/2012/Digital-differences/Main-Report.aspx

*USENIX Journal of Election Technology and Systems (JETS)*
Volume 2, Number 2 • April 2014
www.usenix.org/jets/issues/0202

# A Systematic Approach to Analyzing Voting Terminal Event Logs*

Laurent D. Michel  Alexander A. Shvartsman  Nikolaj Volgushev

Center for Voting Technology Research and Computer Science & Engineering Department
University of Connecticut, Unit 4155, 371 Fairfield Way, Storrs, CT 06268, USA.

**Abstract**

This paper presents a systematic approach to automating the analysis of event logs recorded by the electronic voting tabulators in the course of an election. An attribute context-free grammar is used to specify the language of the event logs, and to distinguish compliant event logs (those that adhere to the defined proper conduct of an election) and non-compliant logs (those that deviate from the expected sequence of events). The attributes provide additional means for semantic analysis of the event logs by enforcing constraints on the timing of events and repetitions of events. The system is implemented with the help of commodity tools for lexical analysis and parsing of the logs. The system was rigorously tested against several thousand event logs collected in real elections in the State of Connecticut. The approach based on an attribute grammar proved to be superior to a previous approach that used state machine specifications. The new system is substantially easier to refine and maintain due to the very intuitive top-down specification. An unexpected benefit is the discovery of revealing and previously unknown deficiencies and defects in the event log recording systems of a widely used optical scan tabulator.

## 1 Introduction

Auditability of electronic voting equipment used in elections emerged in recent years as an important requirement in ensuring the integrity of the electoral process. Election audits may encompass a wide set of activities ranging from the physical or electronic examination of the voting equipment to independent retabulation of election results in the jurisdictions that use paper ballots. While no single auditing activity may be able to answer all questions regarding the proper conduct of an election, it is desirable to have a portfolio of tools and methods for auditing various aspects of an election. Among the important questions that should be posed in any audit is whether a particular voting terminal was used in accordance with established procedures and whether its behavior deviated from the expected. To answer these questions it is instrumental to analyze the event logs generated by most electronic voting terminals.

An earlier work [1] pursued a similar goal for optical scan voting terminals, resulting in a tool that was used in several elections in the State of Connecticut. An interesting

*Research funded in part by the Office of the Secretary of the State of Connecticut and a grant from the U.S. Election Assistance Commission.

by-product of that development was the identification of several defects and deficiencies in the event logging system embedded in the voting terminal. The tool was based on a state transition diagram representation of the way that the voting terminal generates and records events. The state diagram was enriched by rules that captured the expected timed sequences of events that would be consistent with the proper conduct of an election. The system was successfully used to analyze event logs collected from voting terminals used in real elections. This led to several findings surrounding the electoral processes in which actual behaviors did not fully adhere to the prescribed rules and procedures, and incidentally this helped improve the training of election workers. (We note that no deviations from the expected processes could be attributed to malicious actions or severe errors in the voting terminal implementation.)

Yet, the tool [1] has its Achilles' heel as it proved difficult to extend and maintain. Over time, the state transition diagram became very complex due to the sequence of extensions needed to model various aspects of the electoral process, and to identify commonly occurring deviations from the process. This complexity made it difficult to revise the implementation as the necessarily low-level modeling of events and state transitions obscured the high-level model of the election process, resulting in a system where localized changes caused unexpected and broad side-effects. Finally, a clear and meaningful reporting of the analysis results was difficult precisely because of the obfuscation of the overall electoral process. The approach outlined in this paper is offered as a solution to these difficulties and is focused on flexibility, extensibility, and maintainability.

The paper focuses on a specific voting terminal: the Premier's Accu-Vote Optical Scan (AV-OS) terminal that is used in a large number of counties nationwide. Specifically, we present a part of the post-election audit process that deals with the analysis of the event logs. Nonetheless, it is worth pointing out that the results directly generalize to any other voting terminals that produce event logs as well as variations around the electoral process. The event log of the voting terminal is stored in a removable memory card [7]. While the voting terminal hardware may be identical across many precincts deploying AV-OS systems, the contents of the memory cards differ depending on the specific elections in individual precincts. The event log on a memory card contains a history of the actions performed by the voting terminal(s) using the card since it was programmed for the specific precinct.

Event logs are analyzed to determine whether the prescribed election procedure was followed by the poll workers and whether the tabulator operated as expected. (The election procedures for the State of Connecticut are found in [14, 12].) Since it is impossible to verify the actual proceedings, we analyze their traces as recorded in the event logs. A precondition for an accurate log analysis is that the event log is a true, complete, and unambiguous reflection of the events that have occurred during the election process. In the case of the AV-OS event logging system, this is unfortunately not achievable as the event logs may be incomplete or ambiguous. For instance, when several totals reports are printed consecutively, only the first one is logged. This logging deficiency, and others, are discussed in [1]. Additionally, there are several defects in the event logging system that further reduce the accuracy of the logs. Accordingly, we distinguish the probable causes of a given event log deviation. The system detects two different types of deviations: those caused by system malfunction and those resulting from operator errors, e.g., a failure of the poll workers to follow the prescribed procedure.

Note that the deficiency of the AV-OS logging module makes it possible for an operator to conceal invalid behaviors as system malfunctions thereby reducing the usefulness of the logs. Detecting such involved manipulation goes beyond the scope of this paper.

**Our Contributions.** The contributions presented in this paper are as follows.

(1) We specify an event log language in terms of an annotated context-free grammar. The language models the event sequences that are compliant with the proper electoral procedures as well as deviant sequences.

(2) We present a multi-layered compliance analysis that enhances the reporting of deviations. By defining different levels of non-compliance we can classify event sequences with more precision. Rather than simply reporting that a deviation has been observed, we report the class and severity of the deviation.

(3) We use a familiar and accessible modeling medium—context-free grammars—to describe and implement the forensics tool. Context-free grammars are ubiquitous, transparent, easily modifiable and completely extensible, making it a convenient tool for advanced forensics in the hands of professionals. We follow a standard implementation design based on `Bison` [5], an established parser generator tool.

(4) We implement a detailed notification system recognizing a wider variety of deviations, and capable of issuing concise, appropriate, ranked notifications, thus reducing the required amount of manual analysis. These notifications are understandable by anyone with a knowledge of the election process.

(5) The proposed analysis tool features multiple layers of usage. While refining the underlying grammar requires a technical background (context-free grammars and `Bison`), using the diagnostic component does not. In particular, one can imagine a Political Science researcher using our tool to data-mine event logs of past elections. We also note that the meaning of the grammar can be understood by an election official with just a brief introduction by a specialist.

(6) An inventory of previously unknown deficiencies and defects in the AV-OS logging system surfaced as a result of the systematic formalization of the compliant and non-compliant event traces. In particular we observe *a*) several deficiencies in the logging of specific events, *b*) a defect that results in a failure to clear the election counters, and *c*) a defect that creates an ambiguity as whether the first cast ballot was counted.

(7) We performed an analysis on a large archive of log files (over 1,000 files). We used the system to analyze the same collection of log files as in [1]; our new analysis compares favorably with respect to [1].

We emphasize the importance of including the event log analysis as a part of a post-election audit of technology [2]. A careful examination of timestamped events reveals information about the procedures followed during an election process, including the information suggesting improper conduct of an election or malfunction of voting terminals.

Lastly, we observe that while the presented tool is geared specifically towards AV-OS, our methodology emphasizes modifiability. We chose a context-free grammar representation, as opposed to regular expressions or grammars, or a finite state machine model, because its superior expressive power enables us to maintain the desired level of modifiability at a sufficiently high level of abstraction.

It is perhaps valuable to point out that, at least in the case of the AV-OS, the language of traces could be and indeed was modeled with a (counting) finite-state machine. Yet, as the set of deviant behaviors grows, the finite-state machine specification quickly becomes large and cumbersome which significantly hampers the ability to maintain it. Trying to model the language by means of a regular grammar is also problematic: the (left or right) linearity of regular grammars do not allow one to model the language in a hierarchical top-down fashion, and to model deviant behaviors requires the introduction of numerous "transitional" non-terminals that obfuscate the structure of logs. On the contrary, a context-free grammar specification provides a natural top-down definition that is understandable to non-specialists (upon a brief introduction), and is robust in the sense that it elegantly adapts to new deviant traces and remains highly maintainable.

The election processes are fundamentally similar across states; we anticipate that the grammar can be readily modified and/or extended to capture the procedural specifics of a given state election and a given specific voting equipment, provided that the equipment incorporates a temporal event logging system.

**Disclaimer.** In this work, the AV-OS terminal was treated as a "black box." All the results are obtained through testing using the functions provided by the machine, and through observations of how the terminals' use and behavior is recorded in its event logs.

**Related Work.** Numerous research papers have been published on the topic of log analysis, both in the area of election audits, as well as in other digital systems areas [1, 3, 4, 18, 15, 8, 16]. In [16] Wallach et al. propose the generalized log analysis tool Querifier used to identify tampering in secure logs. A high-level framework for log systems is proposed in [9], emphasizing the importance of the completeness and exactness of the log files that determine the effectiveness of any potential log analysis system. This concern is shared across several papers dealing with election log analysis [1, 3, 18]. Baxter et al. [3] developed an automated analysis of audit log files produced by ES&S iVotronic to detect vote miscounts, machines with hardware problems, and polling locations with long lines. Audit logs have also been used to analyse elections for operator misconduct or process deviation in [1] with the focus on AV-OS. Further, Wagner gave an extensive study on the audit logs produced by six voting terminals approved for use in elections in California in [18]. The study examined the support provided by the various terminals for collecting, managing, and analysing the audit log files. The study established several deficiencies across all six systems, in particular in providing third party access to the files. The works [1, 3, 18] underlined that vendors ought to provide a rigorous documentation of the audit log features, including the structure of the logs.

**Document Organization.** Section 2 describes the election and audit process. Section 3 discusses the language of event logs. Section 4 contains the models and definitions. Section 5 presents the modeling of the election traces. Section 6 details the approach, its implementation and discusses our findings. Section 7 discusses the logging system deficiencies. Section 8 presents the results of using our system. We conclude with a discussion in Section 9.

## 2 The Election Process and Audits

Here we describes the election process and election audits. The model given here is specific to the State of Connecticut, but it is easy to see that it is generalizable to other jurisdictions.

**Before Election Day:** Preparations begin at least 30 days prior to the election day. The memory cards of AV-OS terminals are programmed for each precinct. The voting terminals also undergo maintenance and testing to detect any malfunctions and to help prevent failures during the election. The memory cards are programmed by a service company contracted by the State. The programming normally starts three weeks before the election and completes in under 10 days in most cases. Four programmed cards then are securely transported to each polling location. When the cards arrive, usually one to two weeks before the election, officials conduct pre-election tests on all the voting terminals with all the cards. Then the officials randomly select two cards, one to be used in the election and and the other in the back-up terminal. The two selected cards are sealed in their respective voting terminals and are set for election; no further actions must be performed until the election day.

**On Election Day:** The election day imposes strict time constraints on when and what actions are performed. The summary of the activities is as follows.

*Before The Polls Open:* On the morning of the election day, from 5:00 to 6:00 AM before the polls open, the election officials verify the seals on each AV-OS terminal, turn it on, and confirm that the machines are properly initialized. This includes making sure all candidate counters are set to zero, by printing a Zero Totals Report.

*While The Polls Are Open:* Each eligible voter is entitled to a single ballot that s/he receives once they are verified against the voter registration database. Once the voter fills the ballot s/he feeds the ballot to the optical scanner of AV-OS.

*After The Polls Close:* After the polls close at 8:00 PM the officials print the totals report directly from the AV-OS terminal (the event log can also be printed at this point). The results are delivered to the central tabulation process where the totals are computed and reported to the Secretary of the State Office (SOTS) for certification. In jurisdictions that use automated central tabulation, the results can be electronically transferred to the central server either by uploaded directly from AV-OS terminals, or by transferring the cards to the central location for uploading to the server (this is not used in Connecticut).

**Audits:** Three independent audits are performed for each election (in Connecticut).

The *Pre-Election Technical Audit* involves the examination of one randomly chosen memory card from the four cards supplied to each district. Depending on the election, the audit typically covers at least 25% of the districts.

The *Hand Count Audit* is a post-election audit that consists of complete manual counting of a subset of races in 10% of precincts randomly selected after each election.

The *Post-Election Technical Audit* involves the examination of the memory cards used in the election, typically covering up to 30% of the districts.

The technical audits include examining the cards for proper programming and absence of extraneous or unexpected executable code [7]. As the audit results become available, SOTS Office follows up with the districts in cases that raise questions about malfunctions or potential deviations from the proper election process. The results of the follow up are also used to refine the audit process. Our current work deals with a detailed analysis of the event logs collected from the voting terminals.

# 3 The Language of Event Logs and the Previous System

The AV-OS voting terminal incorporates an event logging feature that records selected events and associated timestamps in an internal log. The log can be printed or extracted for analysis. The AV-OS documentation focuses on the meaning of individual events that are logged, but it does not provide a definition of the structure of the event sequences that are to be considered correct with respect to a properly executed electoral process (this is in part sensible, given that jurisdictions may have different expectations of what constitutes a proper process). For these reasons there is a need to reconcile the event logging features of the voting terminal with an externally-specified definition of a proper electoral process. Thus any model, including the finite state machine model in [1] and the model in the current work, that aims to describe the language of valid log sequences is *a*) an approximation, and *b*) designed empirically. The construction of such models requires that event log sequences are manually examined and interpreted. Furthermore, the language of the model needs to be routinely extended and revised to include all observed event log sequences (whether they are representative of expected proper sequences or not). Even with a model that is nearly complete, it is still expected that certain event log sequences may be flagged as invalid without necessarily deviating from the prescribed election procedures. If such a log sequence is encountered, the language of the model needs to be refined to accommodate the newly discovered behavior. Consequently, refining of the model is crucial.

The previous analysis tool [1] classifies log files as normal, i.e., showing a sequence of events expected within a normal course of an election, or as irregular, i.e., containing unexpected events, unusual sequences of events, and unanticipated timing of events. The system is modeled as a finite state machine that simulates the states that the AV-OS terminal may enter during an election process.

Over the course of several audits, two major drawbacks have become apparent. Firstly, the finite state model proved to be difficult to extend and modify. Secondly, invalid log sequences that feature more involved deviations from the norm challenge low-level state machine implementations which cannot produce appropriate and concise notifications (we provide a concrete example in Section 7). Instead of recognizing the main cause of the irregularity, the tool often reports only on its secondary manifestations. This increases the need for substantial additional manual analysis.

To provide a better system that issues precise and meaningful notifications, a mapping between a sizable known set of irregularities and notifications needs to be established. Extending the state machine underlying [1] would lead to a substantial increase in its size. The inherently low-level abstraction of the state machine obscures the higher-level, logical view of the expected event sequencing. Finally, the previous tool does not provide an assessment of the severity of deviations. A high number of notifications (e.g., the several hundred event logs examined after the 2012 election generated over 1,000 notifications), diminishes the benefit of automation when the system is unable to identify event sequences that truly deserve to be manually examined. To be effective, the systems must be able to assign to each notification a severity ranking and meaningfully identify the source of the deviations.

# 4 Models and Definitions

The section details the models underlying the analysis tool and the terminology for the event log analysis. It defines the notions of an election trace and election trace compliance captured through an attributed grammar.

**Event Log Terminology.** A *log file* is the data recorded on a memory card during the election process and it contains a sequence of entries of the types *event*, *time*, and *date*. Event entries are logged when certain events occur. A restart of the machine, for example, leads to a corresponding event entry. Time and date entries are associated with event entries and specify the time at which the event is logged. A *trace* refers to the sequence of entries contained in an event log.

**Attributed Grammar for Traces.** The trace language is defined in terms of a context-free grammar (CFG) [6] given as a quadruple $G = (V, \Sigma, P, Start)$. $V$ is a finite set called the *non-terminals* or *variables*; (Capitalized below). $\Sigma$ is a finite alphabet, disjoint from $V$, called the *terminals*; (*italicized strings* below). $P$ is a finite set of *productions* or *rules*, with each production written as $B \rightarrow \alpha$, where $B$ is a variable, and $\alpha$ is a string from $(V \cup \Sigma)^*$; Lastly, $Start \in V$ is the start symbol.

A non-terminal $B$ *derives* a string $w \in \Sigma^*$, written $B \overset{*}{\Rightarrow} w$ if there is a sequence of productions in $P$ by which $B$ can be transformed into $w$, i.e., $B \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow w$ where each $\alpha_i$ is a string of grammatical symbol and $\Rightarrow$ denotes the one-step expansion in the grammar. The language defined by $G$ is the set of all strings (in our case traces) that can be derived in $G$. An attributed grammar [11] $M$ is a CFG $G$ augmented with a finite set $A$ of *typed attributes* that represent semantic values associated to the corresponding grammatical symbol. Semantic rules are used to define the semantic attribute of a non-terminal as a function of the semantic values of the symbols appearing in its defining productions [17]. A string in the language yields a parse tree, where non-terminals are internal nodes, terminals are the leaf-nodes, and productions determine the parse tree topology.

The attributed grammar presented in this paper models the language of the election traces represented by event logs. The non-terminals of the grammar models the stages of the election. The terminals of the grammar (the alphabet) are the event entries. Finally, the time and date entries are used to initialize attributes associated with the terminals at the leaf-nodes, with the rest of the attributes derived by traversing parse trees.

**Election Traces and Compliance.** Let the language $L_{all}$ be the set of all traces that are known to be produced by the event logging system. This language is defined on the basis of our prior work [1], and by consulting the AV-OS documentation and by hands-on experimentation with the AV-OS voting terminal. The attributed grammar must be able to generate all traces in $L_{all}$.

We partition $L_{all}$ into two subsets, $L_{sc} \cup L_{snc}$. $L_{sc}$ contains all syntactically compliant traces. A trace is *syntactically compliant*, if its event entries occur in the expected order, and at an appropriate stage in the election process. The determination of compliance is made on the basis of the process reviewed in Section 2, and specifically by adhering to the official requirements [14, 12] that impose timing and sequencing constraints. $L_{snc}$ contains all *syntactically non-compliant* traces. These traces are known syntactic deviations (based on [1] and experimentation) that need to trigger notifications when recognized by the parser.

Apart from syntactic compliance, we also consider semantic compliance. A trace is *semantically compliant*, if its event entries occur a permissible number of times, and within the correct time frame. To establish the *semantic compliance* of a trace, semantic analysis is performed by imposing additional constraints whose truth values are expressed by semantic attributes and their defining equations over attributes in $A$. We present the constraints and attributes in Section 6.1. Semantic analysis is performed on all traces in $L_{all}$.

We define a trace as *compliant*, iff it is syntactically and semantically compliant. We let the language $L_c$ be the language of all compliant election traces. A trace $s \in L_c$ iff $s \in L_{sc}$ and $s$ is semantically compliant. We define all traces not in $L_c$ to be *deviations*.

Finally, it is possible to encounter traces that are not in $L_{all}$ when (1) an unrecognized trace may be caused by the voting terminal malfunction (or an error in its software), (2) memory card corruption may cause errors in the stored trace, and (3) a previously unknown trace pattern is encountered (e.g., because voting terminal documentation is incomplete or because this pattern was never seen before), in which case the definition of $L_{all}$ needs to be extended to account for such trace patterns. We define $L_u$ to be the set of all traces that are not in $L_{all}$, or in other words, $L_u$ is the complement of $L_{all}$ with respect to the (unknown) set of traces that can ever be encountered.

**Analysis Flow.** Given a grammar $G$ and its attribute system $A$ that generates $L_{all}$, the analysis of a trace $s$ is as follows. If $Start \overset{*}{\nRightarrow} s$ (trace $s$ cannot be derived), then $s \in L_u$, and it is neither a compliant trace, nor a known invalid deviation. No further automated analysis

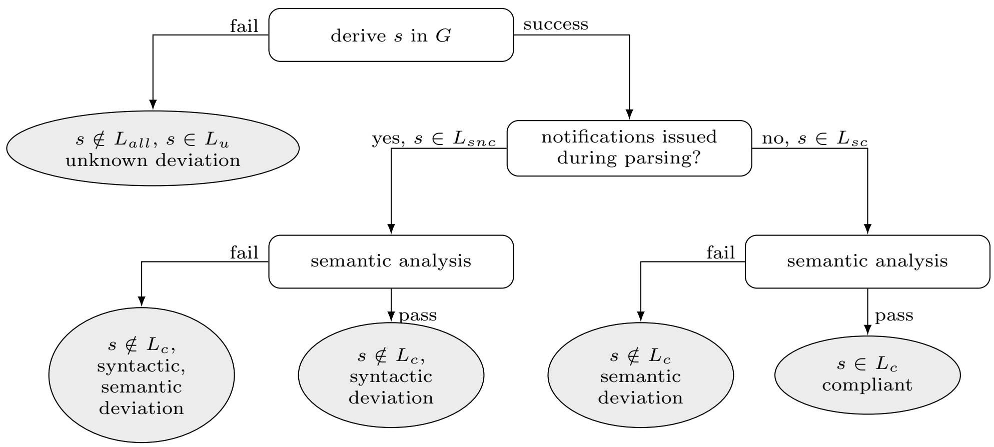


Figure 1: Analysis Flow

is performed. Such a trace needs to be analyzed manually (in the sequel we observe that this happens very infrequently). If $Start \overset{*}{\Rightarrow} s$, two cases are possible: $(i)$ during the derivation one or more notifications are issued, indicating that $s \in L_{snc}$, or $(ii)$ no notifications are issued, indicating that $s \in L_{sc}$ and $s$ denotes a validly ordered sequence of events. Finally, the trace $s$ undergoes semantic analysis. If any deviations in the timing, or the number of occurrences of events are observed, then $s \notin L_c$, and appropriate notifications are issued. If $s$ does not induce any notifications, it is a compliant election trace, i.e., $s \in L_c$.

# 5 Defining the Grammar

The grammar for the language of event traces was developed through an iterative process. Based on our analysis, we include all traces that the voting terminal is able to produce, while delineating between syntactically compliant and non-compliant traces. Regression testing is performed against a known collection of traces. When a trace is found that cannot be generated by the grammar, it is examined manually. If the trace adheres to the requirements [14, 12], the grammar is extended to allow it. If it does not, analysis is performed on whether it is a valid trace per tabulator documentation. If it is, the grammar is again appropriately extended. Otherwise analysis is performed to understand the circumstances that lead to this kind of trace. If the behavior cannot be reproduced, we suspect a tabulator malfunction, else we conclude that there is a defect in the tabulator logging system. Decisions on whether to extend the grammar is made on a case-by-case basis.

## 5.1 Events Recorded in the Log: the Terminals of the Grammar

Table 1 conveys the event names as they are recorded by AV-OS in the log file, with an explanation. Recall that the log file contains *event*, *date* and *time* entries. All event entries in the log are associated with the time entry indicating when the event occurs. Some event entries, i.e., INITIALIZED and SESSION START are also followed by a date entry. Each event entry in Table 1 corresponds to a terminal symbol in the alphabet of the grammar given in Figure 2. In the grammar specification we represent each terminal symbol as the lower-case italicized string corresponding to the event name.

The temporal information associated with the events, i.e., time and date, is not represented in the (non-attributed) grammar. This information is filtered out during the lexical analysis prior to parsing a trace, and is preserved and associated with each event in the trace as its time attribute. We discuss the processing of these attributes in Section 6.1.

| **Event Name** | **Event Description** |
|---|---|
| AUDIT REPORT | Appears when an Audit Report is printed. |
| BAL COUNT END | After the ender card is inserted in an election, this action appears. |
| BAL COUNT START | Appears when the first ballot is cast in an election. |
| BAL TEST START | Records the beginning of a test election. |
| CLEAR COUNTERS | Appears when the counters are set to zero. |
| COM ERROR | A communication error between the machine and the GEMS system. |
| COUNT RESTARTED | The machine is reset during an election, after at least one ballot is cast. |
| DOWNLOAD END | Records the end of data load to the card when using GEMS. |
| DOWNLOAD START | Records the start of data load to the card when using GEMS. |
| DUPLICATE CARD | Records that a card duplication occurs (in the master card and the copy). |
| ENDER CARD | Records when an ender card is inserted, signifying the end of an election. |
| INITIALIZED | The 1st action in the Event Log; this action records date. |
| MEM CARD RESET | The card is returned to 'not set' status, if it was set for election. |
| OVERRIDE | Records an override by a poll worker. Used for overvoted ballots in CT. |
| POWER FAIL | Appears if the machine is unplugged or a power failure occurs. |
| PREP FOR ELECT | Recorded when the card is set for election. |
| SESSION START | Date action. Appears every time you reset the machine. |
| TOTALS REPORT | Appears when a Totals Report is printed. |
| UNVOTED BAL TST | Appears when an unvoted ballot test is performed. |
| UPLOAD END | When an upload is completed, this action is recorded. |
| UPLOAD ERROR | Appears when an upload error is detected. |
| UPLOAD STARTED | Marks the beginning of an upload. |
| VOTED BAL TEST | Appears when an voted ballot test is performed. |
| ZERO TOT REPORT | Appears when a Zero Totals Report is printed. |

Table 1: Action Types

## 5.2 Election Trace Grammar: Non-terminals and Productions

The context-free grammar in Figure 2 defines the syntactic structure of all election traces that can be produced by the AV-OS terminal during the election process, from the time a card is programmed, until the election is closed and the results are printed. The grammar is designed to generate the language $L_{all}$, that is, all election traces that, given the current understanding, can be produced by interacting with the AV-OS terminal. Recall that $L_{all} = L_{sc} \cup L_{snc}$, where $L_{sc}$ are the syntactically compliant traces and $L_{snc}$ are the syntactically non-compliant traces. The productions that are involved in generating any trace in $L_{snc}$ account for known, non-compliant behavior and issue notifications whenever they are triggered during the parsing process.

The productions in the grammar are given in extended Backus-Naur Form (readers familiar with grammars should be able to read it). Reiterating briefly, the non-terminals are given as Capitalized strings and the terminals are the *italicized* strings. Each production has the form "NonTerminal = right-hand-side", where the right-hand-side is a sequence of terminals and non-terminals. The notation "A $\rightarrow$ rhs1 | rhs2" is a shorthand for two production, "A $\rightarrow$ rhs1" and "A $\rightarrow$ rhs2". Zero or one repetition of an expression is denoted

```
1:  Start →  Init, TestPrepElect?
2:        | AbortedInit;

3:  Init →  Initialized, Downloadstart+, Clearcounters, Downloadend
4:       | Initialized, Downloadstart+, Clearcounters, Prepforelect;

5:  AbortedInit →  Initialized, Downloadstart*;

6:  TestPrepElect →  (Test | Test, Prep | Test, Prep, Elect), CardReset?;

7:  CardReset → Memreset, TestPrepElect;

8:  Test →  (Votedbaltest | Unvotedbaltest | Counttestbal)*;

9:  Counttestbal →  Clearcounters, Zerototsreport?, Baltestsstart?, Override*,
                    Endercard, Baltestsstart?, Printtotals*;

10: Prep →   Prepforelect, Clearcounters
11:       | Prepforelect;

12: Elect →  BeforeValBalCast
13:       | BeforeValBalCast, (Balcountstart, Countrestart)*
14:       | BeforeValBalCast, (Balcountstart, Countrestart)*, Endercard,
            Balcountstart, Balcountend, Printtotals*
15:       | BeforeValBalCast, (Balcountstart | Override), AcceptingMoreBallots,
            Endercard, Balcountend, Printtotals*;

16: BeforeValBalCast → Zerototsreport, (Zerototsreport | Balcountstart Zerototsreport)*
17: AcceptingMoreBallots →  (Countrestart | Countrestart Balcountstart | Override)*;
18: Global → audit report | com error | duplicate card | power fail | mem overflow | session start;

19: Balcountend →  Global*, bal count end;
20: Balcountstart →  Global*, bal count start
                   | Global*, bal count start, bal count start;
21: Baltestsstart → Global*, bal test start;
22: Clearcounters →  Global*, clear counters;
23: Countrestart →  Global*, count restarted;
24: Downloadend →  Global*, download end;
25: Downloadstart →  Global*, download start;
26: Endercard →  Global*, ender card;
27: Initialized →  Global*, initialized;
28: Memreset →  Global*, mem card reset;
29: Override →  Global*, override;
30: Prepforelect →  Global*, prep for elect;
31: Printtotals →  Global*, totals report;
32: Unvotedbaltest →  Global*, unvoted bal test;
33: Votedbaltest →  Global*, voted bal test;
34: Zerototsreport →  Global*, zero tot report;
```

Figure 2: Context-free grammar for AV-OS event traces

by the post-fix "?", zero or more repetitions is denoted by the post-fix "*", and one or more repetitions is denoted by the post-fix "+".

To implement the notification of deviant traces, we rely on `Bison` [5] actions that execute an arbitrary piece of C++ code when a grammatical production characterizing a deviation is reduced. Each paragraph below describes a non-terminal, the productions where it appears as the left-hand-side, and the relationship to the election traces. Rule numbers refer to the line numbers in Figure 2.

**Start.** This is the start symbol of the grammar. An election trace can either have a completed Initialization stage followed by an optional sequence of Test Elections, Preparing for an Election, and Elections (rule 1), or an aborted Initialization stage, in which case no further events can be recorded in the log file (rule 2).

**Init.** During the Initialization stage a memory card is programmed. Rule 3 represents the expected sequence of events: an INITIALIZED event (dated), one or more DOWNLOAD START events, a CLEAR COUNTERS indicating that all counters have been zeroed, and finally a DOWNLOAD END event, indicating that the card has been successfully programmed with the data for the selected election. Additionally, it has been observed in at least one trace that a PREP FOR ELECT was substituted for the DOWNLOAD END event. This invalid behavior is accounted for by rule 4. A corresponding notification is issued. (This newly discovered logging deficiencies is discussed later.)

**AbortedInit.** Traces have been encountered where an INITIALIZED event is followed by zero or more DOWNLOAD START events. This sequence occurs when a card is not successfully programmed. E.g., if a card's memory is cleared through the supervisor function at the precinct and reprogrammed (without connecting to an election management system), the download process is aborted and no DOWNLOAD END event is recorded. A notification is issued if rule 5 is triggered since this sequence indicates an incorrect procedure.

**TestPrepElect.** Production rule 6 encompasses all traces that can be generated after the Initialization stage, that is, after the card has been successfully programmed. The expected course of events here is a sequence of test-election related events accounted for by the non-terminal Test, followed by the card being prepared for the election mode, accounted for by the non-terminal Prep, and lastly events related to the actual election process, accounted for by the non-terminal Elect. Note that the production rule includes the following disjunction: (Test | Test, Prep | Test, Prep, Elect). This disjunction accounts for the cards that have not been used in the actual election process. These cards only show events related to test elections and possibly preparing for the election, but no election events. In this case a notification is issued, but only for the post-election audit because in this case it is expected that the trace shows an election. Lastly, it is possible to reset a memory card to the pre-election stage through the supervisor functions at any point after the card has been programmed. This is accounted for by the optional CardReset non-terminal. However, resetting the card is not a compliant behavior and thus it is associated with a notification.

**CardReset.** In case a card is reset during the election stage a MEM CARD RESET event is recorded and the card returns to the pre-election stage. At this point more test elections can be run, the card can be prepared for election and subsequently elections can be run.

**Test.** This non-terminal derives election traces produced during the pre-election stage. At this stage, voted and unvoted ballots can be tested as well as test election performed. Events logged during a test election are accounted for by the Counttestbal non-terminal. All of these events are optional: it is admissible to skip this stage entirely and simply proceed to preparing the card to the election.

**Counttestbal.** This corresponds to the events produced if a test election is run. The procedure and traces recorded in the event log are similar to those of an actual election. A test election starts with the counters being cleared, in which case a CLEAR COUNTERS event is recorded. The user then has the option to print a zero totals report. If the user chooses to do so, a ZERO TOT REPORT appears in the log. Further a BAL TEST START is recorded once the first valid ballot is cast. Overrides can be performed, with each Override recorded in the log. The test election can be ended by inserting an enders card, which is recorded, or by turning the machine off, in which case only a SESSION START event is recorded. After an ender card is cast the user has the option to print a totals report. If an ender card is inserted before any other ballots are cast the system records an ENDER CARD event followed by a BAL TEST START event. (This is a known defect in the AV-OS logging system [1].)

**Prep.** Before an election can be run the card needs to be set into election mode. In the pre-election stage the user has the option to prepare for election. If that option is chosen, a PREP FOR ELECT event is logged. All the counters are supposed to be zeroed and a CLEAR COUNTERS event recorded. A notification is issued if a CLEAR COUNTERS event is missing, which is accounted for by rule 11.

**Elect.** This non-terminal accounts for the events logged during the actual election stage. Rule 15 describes the expected sequence of events. After a card is prepared for election, the machine needs to be restarted. Upon restarting a zero totals report is printed. In this stage, before any valid ballot is cast, a restart of the system always yields another ZERO TOT REPORT event. If the first ballot cast is not a valid ballot, e.g., a ballot not recognized by the machine, a BAL COUNT START event is logged. However, since the counters aren't updated, upon restarting the machine another zero totals report is printed and logged. This behavior is accounted for by the BeforeValBalCast non-terminal. After the first valid ballot is cast, or an override is issued, the system continues to accept ballots until the ender card is inserted. This stage is accounted for by the AcceptingMoreBallots non-terminal. If the machine is restarted in this stage, a COUNT RESTARTED event is recorded. Inserting the ender card ends the election. An ENDER CARD event followed by a BAL COUNT END event is recorded. A totals report is printed and the user is asked if the election should be closed. Shutting off the machine without selecting an option at this point leads to a known error [1]: the TOTALS REPORT event is not logged even though a totals report has in fact been printed. Restarting the machine at this stage results in another totals report printed and logged. Rules 12 and 13 account for cards on which an election has been started but not successfully completed. Notification are issued when these rules are triggered. Rule 14 accounts for the scenario in which the ender card is inserted before any other valid ballots are cast.

**BeforeValBalCast.** After a card has been prepared for election but no valid ballot has been cast yet, restarting the machine results in a zero totals report. If any invalid ballot is cast at this stage a BAL COUNT START event is logged.

**AcceptingMoreBallots.** After the first valid ballot has been cast but the ender card has not yet been inserted, the machine accepts ballots. A restart in this stage results in a COUNT RESTARTED event. If a new ballot is cast, a BAL COUNT START event is logged.

**Global.** This non-terminal can yield any global event terminal, i.e., an event that can occur at any stage in the election process. Note that production rules 19 through 34 simply account for the fact that a global event can occur at any stage of the election process by prefixing any non-global event with a sequence of zero to many global events.

# 6 Event Log Analysis

Now we present the semantic analysis and the notification system.

## 6.1 Attributes and Semantic Analysis

The analysis is performed on the abstract syntax tree (AST) that is constructed during the parsing of a trace. The leaf nodes of the tree correspond to the terminals of the grammar in Figure 2, which in turn correspond to the event entries in election traces. The internal nodes of the tree correspond to stages in the election process, such as the test election stage, prepare for election stage, etc. Each node in the AST has a set of attributes. The semantic analysis consists of verifying that specified events lie within a permissible time frame, and occurred a

| Attribute | Event | Scope |
|---|---|---|
| #S | session start | All |
| #MR | mem card reset | All |
| #PF | power fail | All |
| #CE | com error | All |
| #DS | download start | Init |
| #BT | bal test start | Test |
| #PT | totals report | Test, Elect |
| #ZR | zero tot report | Elect |
| #CR | count restarted | Elect |

Figure 3: Multiplicity Attributes

| Attribute | Event | Scope |
|---|---|---|
| TI | initialized | Init |
| TBT | bal test start | Test |
| TPR | prep for elect | Prep |
| TZR | zero tot report | Elect |
| TBE | bal count end | Elect |

Figure 4: Time Stamp Attributes

permissible number of times. These constraints are expressed through the attributes of the tree nodes. Attributes are synthesized throughout the tree, i.e., derived by a parent node from its children in a bottom-up fashion. There are three types of attributes: *time-stamp*, *multiplicity*, and *constraint* attributes. A time-stamp attribute of some node $D$ in the AST corresponds to the time-stamp of a specified leaf node in the subtree of $D$. A multiplicity attribute of node $D$ in the AST corresponds to the number of times some specified node $E$ occurs in the subtree of $D$. A constraint attribute of node $D$ in the AST corresponds to a multiplicity or time stamp constraint defined over the attributes of the subtree of $D$. A constraint attribute can be defined over the attributes of a specific node in the tree, or over the attributes of several nodes in the tree.

Three components contribute to the semantic analysis: *nodes*, *attributes*, and *constraints*.

**Nodes.** There are seven different types of nodes in the AST. The leaf nodes of the AST are defined as *event nodes*, and as mentioned previously, correspond to the terminals of the grammar. The AST expresses the grouping of events into election stages through node hierarchy. The internal nodes $Init$, $Test$, $Prep$, and $Elect$ correspond to the four different stages of the election process, namely programming the card, the pre-election test mode, preparing for an election, and the actual election stage. Their descendant nodes correspond to the events that are recorded during a given stage in the election process. The $TestPrepElect$ node expresses the fact that the Test, Prep, and Elect nodes are a unit. Every time a memory card is reset, the card returns to the pre-election mode. From there the pre-election, prepare for election, and election stages can be repeated. Each such repetition corresponds to a $TestPrepElect$ node. Finally, the $All$ node is the root node of the tree.

**Attributes.** Figure 3 defines multiplicity attributes, and Figure 4 defines the time-stamp attributes. The Attribute column gives the names of the attributes, the Event column shows the event with which the attribute is associated, and the Scope column shows over which subtree the attribute is defined, e.g., the scope of the attribute. If, for example, the Scope column has the value Test, the attribute is defined over each subtree with root node Test.

Attributes are defined inductively over the nodes of the tree. A multiplicity attribute is a single integer. Given a multiplicity attribute $\#M$ that tracks the multiplicity of event $e$ in the subtree with root node $R$, attribute $\#M$ is defined as follows.

$$R.\#M = \begin{cases} \sum_{c \in children(R)} c.\#M & \text{if } R \text{ is an internal node,} \\ 1 & \text{if } R \text{ is a leaf, and corresponds to } e, \\ 0 & \text{otherwise} \end{cases}$$

A time-stamp attribute is a set of time-stamps. Given a time-stamp attribute $TI$ that tracks the time stamps of all occurrences of event $e$ in the subtree with root node $R$, attribute $TI$ has the following definition.

$$R.TI = \begin{cases} \bigcup_{c \in children(R)} c.TI & \text{if } R \text{ is an internal node,} \\ \{e.timestamp\} & \text{if } R \text{ is a leaf, and corresponds to } e, \\ \emptyset & \text{otherwise} \end{cases}$$

**Constraints.** A constraint attribute is a boolean predicate defined over the time stamp and multiplicity attributes of the nodes. Currently we use two types of constraint attributes: multiplicity constraint attributes and time-stamp constraint attributes. A multiplicity constraint attribute $PM$ of node $N$ defined over multiplicity attribute $\#X$ of node $N$ has the following general form:

$$N.PM = (r_1 < N.\#X < r_2)$$

where $r_1$ is the lower bound on the number of occurrences, and $r_2$ is the upper bound.

A time-stamp constraint attribute $PT$ of node $N$ is defined over its time-stamp attribute $TX$ as follows:

$$N.PT = (\forall t \in N.TX,\ ts_1 < t < ts_2)$$

where $ts_1$ is the earliest date and time on which the tracked event can occur, and $ts_2$ is the latest date and time. Should a constraint attribute evaluate to false, a notification is issued stating that a semantic constraint has been violated.

We note that the time-stamp constraints are crucial to the analysis due to the time-sensitive nature of the election. We build on the foundation for the temporal analysis of election traces given in [1]; following that work we next provide a detailed example of the temporal restrictions on the election process.

The election day order imposes time limitations on when and what actions can be performed. According to the election process we expect to see the following: from 5:00 to 6:00 AM election officials should turn on the AV-OS terminal to be used in the election and produce a zero totals report. Thus we expect to see a session start (SESSION START) event about an hour before the polls open, followed by a zero totals report (ZERO TOT REPORT) event. We expect to see the ballot count start (BAL COUNT START) event after the time the polls open. This can be followed by a sequence of override (OVERRIDE) events. Finally, by the time the polls close, we expect to see the ender card (ENDER CARD) event, followed by a ballot count end (BAL COUNT END) and a totals report (TOTALS REPORT) events.

**Example AST.** Figure 5 shows an example trace, and the resultant AST is given in Figure 6. In the interest of space we abbreviate the names of the leaf nodes; the abbreviations are given next to the event names in Figure 5. The tree is augmented with four attributes: $\#S$, $TI$, $PM$, and $PT$. The attributes are written in italic.

For the sake of clarity we have only included four attributes in the tree. Attribute $\#S$ is a multiplicity attribute which tracks the number of occurrences of the SESSION START event throughout the entire election process. Attribute $TI$ is a time-stamp attribute that tracks the time-stamps of all INITIALIZED events that are logged during the Initialization stage. $PM$ and $PT$ are constraint attributes. $PM$ is defined for the All node of the tree, and restricts the SESSION START event to occur between 0 and 3 times: $All.PM = 0 \leq All.\#S \leq 3$. $PT$ is defined for the Init node of the tree, and restricts the time for all INITIALIZED events in the Initialization stage to occur between 0:00 on 10-20-12 and 11:59 on 10-30-12: $Init.PT$ = ($\forall t \in Init.TI$, 0:00 10-20-12 $\leq t \leq$ 11:59 10-30-12). The attributes are synthesized throughout the tree as described in the previous section.

***Remark.*** The semantic analysis presented above relies on semantic attributes and constraints to recognize non-compliant traces. Programming languages do rely on type systems

| **Event** | **Abbreviation** | **Time** | **Date** |
|---|---|---|---|
| initialized | init | 12:00 | 10-20-12 |
| download start | dlstart | 12:00 | 10-20-12 |
| clear counters | clrc | 12:00 | 10-20-12 |
| download end | dlend | 12:00 | 10-20-12 |
| session start | sstart | 11:00 | 10-24-12 |
| bal test start | btest | 11:05 | 10-24-12 |
| ender card | ender | 11:07 | 10-24-12 |
| totals report | trep | 11:10 | 10-24-12 |
| prep for elect | prep | 11:20 | 10-24-12 |
| clear counters | clrc | 11:20 | 10-24-12 |
| session start | sstart | 05:30 | 11-06-12 |
| zero tots report | zrep | 05:31 | 11-06-12 |
| bal count start | bcstart | 08:01 | 11-06-12 |
| ender card | ender | 20:14 | 11-06-12 |
| bal count end | bcend | 20:14 | 11-06-12 |
| totals report | trep | 20:14 | 11-06-12 |

Figure 5: Example Election Trace

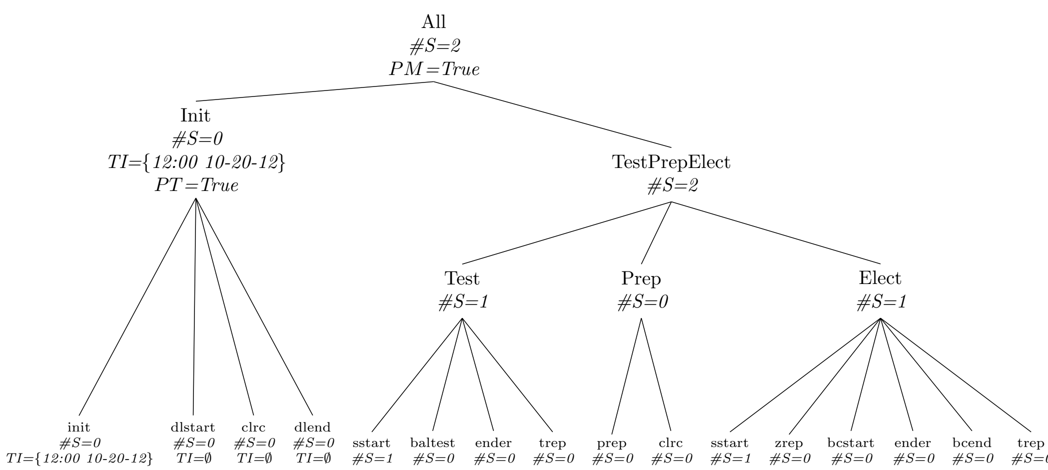


Figure 6: Abstract Syntax Tree (AST)

to annotate and analyze parse trees, hence, it it tempting to assess their potential for our analysis. Given the key roles of multiplicities and timestamps in such an analysis, one must consider so-called *dependent type systems* [13] to capture, within types, the critical inputs used by the analysis. While this is certainly a feasible avenue, it is not clear that it would significantly simplify the specification currently captured through constraints.

## 6.2 Notification System

The outputs of the parsing process and the semantic analysis manifest themselves through the notification system. It includes four types of notifications: *syntax errors* that are issued when an election trace cannot be parsed, *grammar notifications*, issued when deviant productions are triggered during parsing, *multiplicity notifications*, issued when a specified event occurs too few, or too many times, and *time notifications*, issued in case a specified event occurs outside of its permissible time range. Both repeat and time notifications are generated during the semantic analysis, after the input file has been successfully parsed.

Syntax errors are rare and signify that the offending log file shows major deviations from the expected format (suggesting an unknown software or hardware malfunction in the voting terminal, or a failure of the memory card that stores the log). In a recent analysis of over 500 memory cards, we have encountered only one syntax error. Once a syntax error is encountered and the manual examination suggests that it is not due to equipment failure, the grammar can be amended with additional productions so that similar future errors are recognized and flagged accordingly without raising another syntax error. Analysis of syntax errors is crucial in identifying equipment failures and/or exposing new deficiencies in the logging system. (Instances of this are discussed in Section 7.)

Grammar notifications are issued to account for known, erroneous behavior where multiplicity or timing semantic constraints are violated. E.g., if the last event recorded in the log is a BAL COUNT START event, with all the subsequent, expected events missing, a notification is issued stating that the election has been aborted midway. All notifications carry custom messages to concisely explain a specific, known deviation from the expected behavior.

# 7 New Identified Event Logging Deficiencies

The implementation of the event logging in the AV-OS terminal is known to contain several defects and deficiencies [1]. Our systematic approach helped reveal additional issues with the AV-OS logging. Although doing so was not among the goals of our development, we note that it is the rigor of our approach and the comprehensiveness of our analysis that enabled the identification of new defects and deficiencies.

**Manifestations of Logging Deficiencies.** Here we elaborate on the deficiencies in the AV-OS logging system exposed by our analysis, provide examples, and discuss the ramifications. Consider the three deviating traces shown in Figure 7. For brevity, we only show the relevant section of a given trace. The three event traces are real—these have been recorded during the official voting terminal use.

Event log A in Table 7 shows an election run, followed immediately by a PREP FOR ELECT event. It is possible, after an election, to reset a card to the pre-election stage. Resetting a card is done through the supervisor functions and is recorded in the event log as a MEM CARD RESET event entry. In Log A, however, no MEM CARD RESET is logged between the concluded election and the PREP FOR ELECT event. We have not been able to reproduce such a log sequence using plausible scenarios, leading us to suspect that this is either an intermittent error or unintended behavior due to a race condition. The time-stamp of the first election indicates it was run before the official election date, during the test stage. We have encountered several event logs suggesting that in testing the voting terminal an election was run instead of a test election (while it can be beneficial to run an election instead of test election during pre-election test, the current official rules require that test election is run). This incorrect usage appears to be harmless. However, log A suggests a machine malfunction. Not only is the MEM CARD RESET event absent from the log, we also see that the PREP FOR ELECT event is not followed by a CLEAR COUNTERS event, thus the next election starts with a COUNT RESTARTED event. This indicates that the counters from the previous election were not zeroed, and thus the totals of this subsequent election are incorrect. Fortunately, in this case, the subsequent election is merely another election run as a test election, so no election data was compromised. However, the same machine malfunction could affect actual election counts.

| Event Log A | Event Log B | Event Log C |
|---|---|---|
| 10-31-12 | 11-03-12 | 11-04-08 |
| 15:38 ZERO TOT REPORT | 11:05 PREP FOR ELECT | 06:41 ZERO TOT REPORT |
| 09:29 BAL COUNT START | 11:05 CLEAR COUNTERS | 07:00 BAL COUNT START |
| 09:31 ENDER CARD | 14:11 SESSION START | 07:05 SESSION START |
| 09:31 BAL COUNT END | 11-06-12 | 11-04-08 |
| 10:32 TOTALS REPORT | 14:26 ZERO TOT REPORT | 07:06 ZERO TOT REPORT |
| 10:33 SESSION START | 15:18 BAL COUNT START | 07:07 BAL COUNT START |
| 11-01-12 | 15:34 BAL COUNT START | 20:58 ENDER CARD |
| 10:33 PREP FOR ELECT | | 20:58 BAL COUNT END |
| 10:33 SESSION START | | 21:02 TOTALS REPORT |
| 11-01-12 | | |
| 10:33 COUNT RESTARTED | | |
| 10:34 ENDER CARD | | |
| 10:34 BAL COUNT START | | |
| 10:34 BAL COUNT END | | |
| 10:52 TOTALS REPORT | | |

Figure 7: Event logs causing syntax notification (A) and grammar notifications (B and C)

| Event Log | Type | Severity | Message |
|---|---|---|---|
| Log A | Syntax | 10 | Syntax Error at line 33. |
| Log B | Grammar | 2 | Two consecutive BAL COUNT START events. |
| Log B | Grammar | 2 | Election aborted before Balcountend. |
| Log B | Date | 1 | PREP FOR ELECT occurred at 11:05 11-03-12. |
| Log B | Date | 2 | ZERO TOT REPORT occurred at 14:26 11-06-12. |
| Log C | Grammar | 1 | ZERO TOTALS REPORT after BAL COUNT START. |

Figure 8: Notifications issued for event logs A, B, and C

This situation was identified because our system produced a syntax error notification, as shown in Figure 8. We note that the analysis tool in [1] failed to report this log file as non-compliant.

The analysis of event log B (Figure 7) results in two grammar and two time-stamp notifications (Figure 8). The grammar notifications indicate that two consecutive BAL COUNT START events occur and that the election has not been properly concluded. The BAL COUNT START event should only occur when the first ballot is cast in an election, or if a ballot is cast after the machine has been restarted. The latter should be preceded by a SESSION START and a COUNT RESTARTED. Log B, however does not display this sequence of events. The consecutive BAL COUNT START events indicate another deficiency. Log B initially resulted in a syntax error, however, the addition of an error-handling production (production 19, Figure 2) accounts for this behavior and issues an appropriate grammar notification. Time stamp notifications indicate that two events have occurred outside of their allotted time-frame.

Log C exposes another logging system deficiency. A BAL COUNT START at 7:00 is logged, followed by a SESSION START and a ZERO TOTALS REPORT. The BAL COUNT START event indicates that a ballot has been cast, while the ZERO TOTALS REPORT indicates that the counters are at zero. This means that casting the first ballot did not increment the counters. This event trace occurs when the first ballot cast in an election is not recognized by the machine. The ballot is cast and the BAL COUNT START event is logged, however, since the ballot cannot be read, the counters are not incremented.

**Discussion of Mitigations.** In light of the above problems we strongly recommend that a rigorous analysis of the AV-OS hardware and software is performed by the vendor to determine the cause of the machine malfunction captured in log A, and the logging deficiency in log B. Further we recommend that the training of poll officials be improved.

There is also an enhancement to the AV-OS logging system that would greatly reduce the ambiguity in the logs. Currently it is impossible to discern whether a memory card was used in one terminal only, or in several different terminals throughout the election process. In fact, the established procedures implicitly require that every memory card that was used in an election are used in at least two different terminals. First, each card is programmed on one terminal (using the election management system), then it used with a different terminal at the precinct during the actual election. Each precinct has two voting terminals, and any card can be switched at any time from one terminal to the other. This causes ambiguity in the log analysis assessment and presents the potential for masking incorrect tabulation or improper use of the voting terminals. One way to approach this is to augment the SESSION START action type. The SESSION START event is logged whenever a voting terminal with a memory card already inserted is turned on, or when a memory card is inserted into a terminal. Currently it is impossible to distinguish between these two scenarios because the terminal is never identified inside the event log. Supplementing the SESSION START event with a paired event that records a unique voting terminal identifier would resolve the ambiguity.

# 8 Running the Event Trace Analysis System

The analysis tool is implemented in C++, using `Bison` [5], a GNU parser generator, and using `Flex` [10] for lexical analysis. The grammar is an unambiguous, zero-conflict, deterministic LR(1) grammar. In the course of the development, refinement, and regression testing of our system we used it to analyze several thousand event logs collected in recent years from actual elections. The performance of the system is quite good, certainly making it feasible to rapidly analyze large numbers of event logs. In particular, running the analysis tool on a conventional laptop, the system is able to process over 150 log files per second on average.

**Case study.** We review the results produced by the analyzer on a data set of 421 event logs from the 2008 Presidential Election (these represent a total of 30% of all districts: 10% randomly selected districts and 20% of districts that chose to participate in the audit); 279 logs were collected from AV-OS voting terminals used in the election, and 142 logs were collected from the back-up terminals. The date and repeat notifications produced by our tool are consistent with those reported in [1]. We now summarize selected observations.

In event logs from 19 terminals (four of these were used in the election) we observed a MEM CARD RESET event. Resetting a memory card places it in the pre-election state. However, the election protocol states that memory cards should not be reset. Fortunately, all 19 cards show an election run during the test election stage. Thus this is not problematic, since preparing for election zero the counters anyway. Nevertheless, the rules disallowing resetting are there for a purpose. If a card were to be reset in the middle of an election, after ballots have been cast, the counters would be zeroed and votes lost. For example, one event log containing MEM CARD RESET events was reset on and used on the election day. The

MEM CARD RESET event is logged less than twenty minutes before the first ballot is cast in the election. Resetting a card on the election day should not occur under any circumstance.

In two event logs we found incomplete election sequences. Both logs end on a BAL COUNT START event logged on the election day. All further expected event entries are missing. This indicates that either the election was not completed properly, or that the memory card was removed before the election was concluded. However, the cards have zero counters, thus no ballots were cast.

Additionally, we discovered events that were not supposed to occur in post-election event logs. The AUDIT REPORT events are recorded in two logs. Two logs contained UPLOAD START events. Neither the printing of an audit report nor the upload of any data should occur during the election process. This indicates the election procedure was not followed.

**Logging system deficiencies.** We have also identified new deficiencies in the event logging system. As we already discussed earlier (and in [1]), these deficiencies result in ambiguity and could in principle be used to mask invalid operator behavior.

Our analysis detected three event logs showing a BAL COUNT START event, followed by a ZERO TOTALS EVENT. This exposes a deficiency in the logging system and suggests that the first ballot cast in the election was not read correctly by the machine. It is impossible to establish whether the ballot cast was indeed invalid, or whether the machine simply failed to read it.

Another logging deficiency related to ballot casting was observed in one log where an election is run during the test election stage. The (out-of-order) sequence of events for the election is as follows: ENDER CARD, BAL COUNT START, BAL COUNT END. We have been able to reproduce this log trace by inserting the ender card before any other ballots are cast. Such an event sequence occurs if the first ballot cast in an election is the ender card.

While the results of our analysis confirm that in several instances election procedures were not followed, we have found no indication of security problems or malicious intent. However in some cases the analysis suggests software/hardware malfunction.

# 9 Conclusions and Future Work

Election audits are a critical procedural component of the electoral process to guarantee the proper conduct of an election. Our work demonstrates yet again how audits can be valuable in the forensic analysis of data collected from voting terminals used during the election. Indeed, the audit process reveals several classes of problems ranging from voting terminal malfunctions and defects to deviations in the recommended behaviors for system operators. Our contributions encompass a new formalization of voting machine event logs to systematize a multi-layered compliance analysis that delivers detailed notifications characterizing election traces. The event log analysis uses attributed context-free grammars, making the system highly extensible and maintainable, and readily available for refinements that reflect requirements for a correct conduct of an election. Additionally, our methodology led to the identification of previously unknown deficiencies and defects in the AV-OS logging system, further emphasizing the value of comprehensive audits.

We are currently preparing recommendations on implementing event logging systems for voting terminals that would enable even more comprehensive audit analyses. In our future work we will continue refining our approach and we intend to adapt the language definition for use in other jurisdictions using similar equipment. Other research directions prompted by our work include the exploration of machine learning as a means for automating the generation of grammars, and the development of an automated approach for aligning grammars with the requirements and expectations of other jurisdictions. Finally, we intend to make our system available for educational and research use.

**Acknowledgments.** The authors thank the referees for detailed and insightful comments that resulted in substantial improvements to our presentation.

# References

[1] Antonyan, T., Davtyan, S., Kentros, S., Kiayias, A., Michel, L., Nicolaou, N., Russell, A., and Shvartsman, A. Automating voting terminal event log analysis. In *Proceedings of the 2009 conference on Electronic voting technology/workshop on trustworthy elections* (Berkeley, CA, USA, 2009), EVT/WOTE'09, USENIX Association, pp. 15–15.

[2] Antonyan, T., Davtyan, S., Kentros, S., Kiayias, A., Michel, L., Nicolaou, N. C., Russell, A., and Shvartsman, A. A. State-wide elections, optical scan voting systems, and the pursuit of integrity. *IEEE Transactions on Information Forensics and Security 4*, 4 (2009), 597–610.

[3] Baxter, P., Edmundson, A., Ortiz, K., Quevedo, A. M., Rodríguez, S., Sturton, C., and Wagner, D. Automated analysis of election audit logs. In *Proceedings of the 2012 International Conference on Electronic Voting Technology/Workshop on Trustworthy Elections* (Berkeley, CA, USA, 2012), EVT/WOTE'12, USENIX Association, pp. 9–9.

[4] Bing, M., and Erickson, C. Extending unix system logging with sharp. In *Proceedings of the 14th USENIX Conference on System Administration* (Berkeley, CA, USA, 2000), LISA '00, USENIX Association, pp. 101–108.

[5] Bison gnu parser generator manual. Free Software Foundation http://http://www.gnu.org/software/bison/manual/bison.html, 2013.

[6] Chomsky, N. Three models for the description of language. *Information Theory, IRE Transactions on 2*, 3 (1956), 113–124.

[7] Davtyan, S., Kentros, S., Kiayias, A., Michel, L., Nicolaou, N. C., Russell, A., See, A., Shashidhar, N., and Shvartsman, A. A. Pre-election testing and post-election audit of optical scan voting terminal memory cards. In *Proceedings of the 2008 USENIX/ACCURATE Electronic Voting Workshop (EVT 08), July 28-29, 2008, San Jose, CA, USA* (2008).

[8] Detecting security incidents using windows workstation event logs. SANS Information, Network, Computer Security Training, Research, Resources https://www.sans.org/reading-room/whitepapers/logging/detecting-security-incidents-windows-workstation-event-logs-34262, 2013.

[9] Etalle, S., Massacci, F., and Yautsiukhin, A. The meaning of logs. In *Proceedings of the 4th International Conference on Trust, Privacy and Security in Digital Business* (Berlin, Heidelberg, 2007), TrustBus'07, Springer-Verlag, pp. 145–154.

[10] Flex (the fast lexical analyzer). Sourceforge http://flex.sourceforge.net, 2008.

[11] Knuth, D. E. Semantics of context-free languages. *Theory of Computing Systems 2*, 2 (June 1968), 127–145.

[12] Marksense voting tabulator. Connecticut. Secretary of the State. http://www.ct.gov/sots/lib/sots/legislativeservices/regulations/12_opscanusereg.pdf, 2008.

[13] Martin-Löf, P. *Intuitionistic Type Theory*. Bibliopolis, Naples, 1984.

[14] Connecticut. Secretary of the State. Print.

[15] Rouillard, J. P. Refereed papers: Real-time log file analysis using the simple event correlator (sec). In *Proceedings of the 18th USENIX Conference on System Administration* (Berkeley, CA, USA, 2004), LISA '04, USENIX Association, pp. 133–150.

[16] Sandler, D., Derr, K., Crosby, S., and Wallach, D. S. Finding the evidence in tamper-evident logs. In *Proceedings of the 2008 Third International Workshop on Systematic Approaches to Digital Forensic Engineering* (Washington, DC, USA, 2008), SADFE '08, IEEE Computer Society, pp. 69–75.

[17] Slonneger, K., and Kurtz, B. *Formal Syntax and Semantics of Programming Languages: A Laboratory Based Approach*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[18] Voting system audit log study. California Secretary of State, Debra Bowen http://www.sos.ca.gov/voting-systems/oversight/directives/audit-log-report.pdf, 2010.