Recommendations for Voting System Event Log Contents and Semantics

Paul T. Cotton, Andrea L. Mascher, Douglas W. Jones
{pcotton, amascher, jones}@cs.uiowa.edu

October 29, 2009

Abstract

Data recorded in the event logs of currently deployed electronic voting systems are insufficient to diagnose problems ranging from ballot design to user interface errors which may occur in voting systems. In the following work, we propose several types of voter interaction information which could be included in voting system event logs and highlight several policy decisions that must be considered before finalizing data format requirements.

1 Introduction

While there is significant focus the syntax of voting system event logs, decisions about what event logs should contain should be decided before format requirements are made.

In recent years, there have been several high-profile, irregular elections involving electronic voting systems. Of particular interest has been the abnormally high undervote rate in Florida's 2006 Sarasota County Congressional District 13 ("CD13") contest, which has inspired several post-election investigations [2, 5, 8]. These studies have had limited success in unambiguously determining the root causes of the problems in CD13, with a common complaint that the event logs did not record sufficient information to provide conclusive evidence to prove or disprove proposed theories.

In this paper, we discuss the event logs that would be used in a theoretical well-designed and fault-tolerant electronic voting system. We define such a log to be a timestamped, sequential record of information regarding voter or election official interaction with the voting system, and relevant associated changes in the system's state.

In the following section, we will examine two proposed approaches that were motivated in part by CD13 to illustrate possible approaches to designing the content requirements of event logs.

2 Related Work

The first question which needs to be addressed regarding voting system event logs is whether logs should be public documents used to encourage transparent elections, or if they should remain private to preserve voter secrecy. This determines what information can be recorded.

One possible approach would be to use strong logging that allows for complete reconstruction of voter intent. Cordero and Wagner proposed such a logging scheme. However, it is unable to conform to our requirement that event logs contain timestamps. This is also a federal requirement[1, 3, 4, 7].

Effectively, the Cordero-Wagner logging scheme captures a screen shot each time there is a change to the display and the touch coordinates that caused the transition to the next screen. These are grouped and stored in sequence for each voter. The series of these images can be replayed at a later time in order to re-

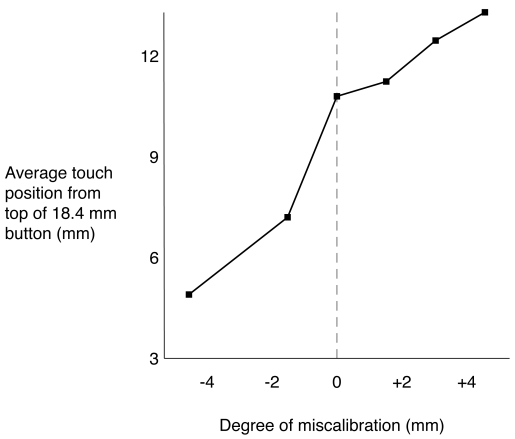
construct any given voter's interactions with the system. To preserve voter privacy, timestamps are not recorded and the sets of interaction sequences are stored in a random order. These privacy protections do not significantly increase the risk of leaking voter selections relative to that of paper ballots, provided that the logs are not publicly distributed. A major drawback to this approach is that it does not provide timestamps for other significant events such as system errors, ballot initialization, or vote casting, thus Cordero and Wagner's replayable logs are not a substitute for conventional event logs.

An alternate approach would be to use a log that is capable of identifying system issues, such as user interface errors or ballot design problems, without directly recording intent. Such a log could be publicly released without increasing the risk of vote selling or coercion.

We have implemented such a logging scheme which is intended to comply with existing standards documents [6]. This logging scheme can contain absolute timestamps and maintain the original ordering of both system events (such as system errors or ballot initialization) and user interface events (such as candidate selection or ballot page navigation). Additionally, some interaction data is recorded at the candidate selection level: it is noted in the log when a voter selects (or deselects) a candidate, but there is no indication of which candidates are involved. See section 3 for a full description. The current implementation of this scheme has some risk of leaking information about voter abstentions, but there are several possible approaches to mitigate this.

In addition to recording system events and problems, our approach to event logging allows for the identification of interface design issues. We have shown in [6] that ballot interface and design impact page navigation patterns while the event logs of voters using miscalibrated touchscreens show distinct changes in touch behaviors from the norm. Figure 2 shows how both direction and magnitude of touchscreen miscalibration can be determined from the average touch coordinates relative to displayed buttons.

Figure 1: Average Calibration Touch Position



3 User Interface Event Logging

The user interface logging scheme described below is intended to augment an existing event logging scheme for a touch screen voting system. Records of events should contain anonymized voter interactions such that no recorded information will reveal a voter's selections. However, such anonymized logs should retain enough information to be useful in post-election investigations.

3.1 System Events

System events are actions by the voting system that may impact how the voter can interact with the system. For example, a voter cannot make selections when a ballot has not yet been loaded.

- INITIALIZE: Load a new ballot. The ballot style is also recorded when multiple ballot styles are supported.
- UPDATE: Report any change to the screen (eg. highlight, unhighlight, change page)

- CAST: Finalize voter session

Note: it is not necessary to record the type of UPDATE event. Since each UPDATE follows either a selection, navigation or initialization, the type can be inferred.

3.2 Input Events

To anonymize the record of voter input events, we record the minimum information needed to diagnose interface problems. There are two types of locations on a touch screen that a voter could touch: a button or the background. A touch on the background does not change the state of the ballot or screen, but an excessive number of background touches may indicate a system issue. It is often the case that a background touch is a miss on a nearby button, so to preserve voter privacy, we only need to know when a background touch occurs, not where.

When a button is touched, the button type (such as candidate selection or navigation) and button action (such as selection, deselection, or the navigation destination) should be recorded. The location where the button was touched is recorded as an (x,y) pair of the pixel distance relative to the button itself, not to the screen as a whole. This prevents leaking a voter's selection, since touching the exact center of a 200x100 button for either "Candidate A" or "Candidate B" would be recorded as (100,50). These relative coordinates are not associated with the identity of button touched so it is impossible to determine which candidate the voter selected. We record the navigation destination screen type (Candidate Selection, Write In, Summary, Cast Ballot) to give diagnostic information about the approximate location in the ballot where issues occur, but we do not record the exact identity of the button or page.

Navigate Button Actions:

- Next/Previous screen
- Write In/Return to Ballot page
- Review/Return to Summary
- Cast Ballot

Candidate Button Actions:

- SELECT: Highlight a button.
- DESELECT: Unhighlight a highlighted button.
- I-SELECT (Invalid Select): The button cannot be highlighted because the maximum number of buttons has already been selected.

4 Discussion

Both event log schemes discussed in section 2 are designed to detect and identify user interface issues (miscalibration, poor ballot design, etc.). However, Cordero and Wagner have designed their event logs such that they could potentially be used in election recounts, while the event logs proposed by Mascher, et al. prioritize ballot privacy as they assist post-election investigations. Even though the final goals for both event logging schemes are similar, differences in their premises create large differences in how these schemes are implemented.

The first step of standardization should be to agree on the what actions an event log should help facilitate, for example: post-election audits, investigations, or a recount based on voter intent. In the remainder of this section, we will outline a proposal for the steps to standardize an event log for post-election investigations.

Second, a decision should be made regarding the acceptability of event logs that cannot be made public without risks to privacy. If it is decided that a private log is acceptable, then a two level approach should be considered. That is, the logging system should contain one log composed of a replayable log for each voter which does not contain timestamps and does not preserve the order of voters who used the system, while a second more minimal log contains a timestamped

recording of critical system events such as when new voter sessions begin and when ballots are cast. With careful consideration of past investigations, such as those for CD13, a standard set of potential problems that event logs need to detect should be agreed upon. Then there should be a research effort to determine the minimum amount of data required to identify those errors that the event log is designed to diagnose. It is important to consider the risk of reducing ballot secrecy when deciding what data event logs should contain. Since increasing the amount of information contained in a log increases the risk of violating voter privacy, any logging standard should collect the minimum amount of information needed to identify expected problems.

After these steps are completed, implementation details such as a standard syntax and semantics for event logs should be developed. The common format for event logs will be most strongly influenced by the decision to use public or private logs, as this decision could require changes to federal regulations and guidelines. The format should allow for future extensions to be incorporated into event logs in the event that it is determined that more information is needed. The basic design of any new formatting standard must not preclude a possible logging design until such an approach has been ruled out as an option.

References

[1] Cordero, A., and Wagner, D. Replayable voting machine audit logs. In *Proceedings of the 2008 USENIX/ACCURATE Electronic Voting Technology Workshop* (2008).

[2] Dill, D. L., and Wallach, D. S. Stones unturned: Gaps in the investigation of Sarasota's disputed congressional election. 2007.

[3] Federal Election Commission. Performance and test standards for punchard, marksense and direct recording electronic voting systems. Tech. rep., 1990.

[4] Federal Election Commission. Voting systems performance and test standards. Tech. rep., Federal Election Commission, 2002.

[5] Frisina, L., Herron, M. C., Honaker, J., and Lewis, J. B. Ballot formats, touchscreens, and undervotes: A study of the 2006 midterm elections in Florida. *Election Law Journal 7*, 1 (March 2008), 25–47.

[6] Mascher, A. L., Cotton, P. T., and Jones, D. W. Improving voting system event logs. In *Proceedings of the First International Workshop on Requirements Engineering for E-voting Systems* (2009).

[7] U.S. Election Assistance Commission. Voluntary voting system guidelines. Tech. rep., 2005.

[8] Yasinsac, A., Wagner, D., Bishop, M., Baker, T., de Medeiros, B., Tyson, G., Shamos, M., and Burmester, M. Software review and security analysis of the ES&S iVotronic 8.0.1.2 voting machine firmware, final report. Tech. rep., Security and Assurance in Information Technology Laboratory, Florida State University, February 2007.