Who Counts Your Vote, and How?

Dirk Pattinson

The Australian National University

June 2015

joint work with Carsten Schürmann, Copenhagen

Electronic Voting Is Happening

- Australia (counting and remote voters)
- Belgium (vote casting)
- Brazil (casting and counting)
- Canada (municipial level)
- Estonia (general elections)
- France (overseas residents)
- India (state and national)
- Italy (Ceremona 2006)
- Phillipines (national elections)
- Romania (overseas forces)
- Switzerland (expatriates)
- US (2010 Arizona Primaries)

But not without Problems

Finnland 2008. Scytl voting machines failed to record 232 votes resulting in recount

Germany 2005. Hamburg pilot scrapped because of lacking public confidence

Ireland 2002. Unsuccessful pilot in 3 constituencies in general election. Scrapped due to public dissatisfaction.

Netherlands 2007. Voting machines banned in 2007 over security concerns.

Phillippines 2010. 76,000 of 82,000 voting machines were faulty and had to be replaced within two days.

Scotland 2007. 150,000 spoilt votes.

Aspects to consider

For voters

- participation of visually impaired and disabled persons
- voters in remote locations or overseas

For election authorities

- possibly more resistant to manipulation
- more complex, but fairer voting schemes

What does Trust Really Mean?

"Those who cast the vote decide nothing. Those who count the vote decide everything."

What does Trust Really Mean?

"Those who cast the vote decide nothing. Those who count the vote decide everything."

Josef Stalin (1923)

What is Trust?

Paper Based Elections

- public scrutiny
- election observers

Electronic Elections

- . . . ?

Example: Single Transferable Vote

1. calculate the quota to be elected and count first preferences
2. candidates with enough first preferences to meet the quota are elected, surplus votes are transferred to next preference
3. if there are still seats to fill, eliminate candidate with fewest first preferences and transfer her votes

Used e.g. in Malta, Australia, Ireland (parliamentary elections), Northern Ireland (national assembly), Scotland (local governments), New Zealand (local governments), US (some city elections), Pakistan (senate), India (upper house), Iceland (constitutional assembly).

Australian National University

Electing the Most Influential Leader

Ranking	# votes
Barack > Li > Angela	4
Li > Barack > Angela	3
Angela > Li > Barack	2

First Preference Count. Barack wins

(But the majority thinks that Li is more influential than Barack.)

STV Count.

- Use Droop quota $= 1 + (\#\text{votes}/\#\text{seats} + 1) = 5$
- first tally: Barack [4], Li[3], Angela[2].
- Angela is eliminated, her votes are transferred
- second tally: Barack[4], Li[5] and Li is elected.

Australian National University

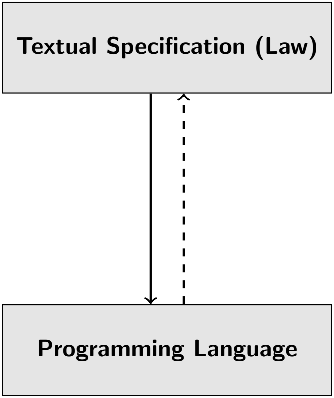
Specificaton of Voting Protocols

Example. Termination condition in the Hare-Clark Act

1. If, after a calculation under clause 3 (3), 6 (4) or 9 (2) (d), the number of successful candidates is equal to the number of positions to be filled, the scrutiny shall cease.
2. If, after a calculation under clause 3 (3) or 6 (4) or after all the ballot papers counted for an excluded candidate have been dealt with under clause 9 –
   (a) the number of continuing candidates is equal to the number of positions remaining to be filled; and
   (b) no successful candidate has a surplus not already dealt with under clause 6;

   each of those continuing candidates is successful and the scrutiny shall cease.

The Translation Problem



- human translation from text to source code
- validation by translating source code to text?

How can we ensure that this process is trustworthy?

Approaches to the Translation Problem

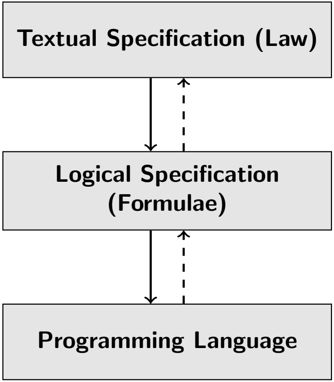
New South Wales 2015 Election: Trust through closed source

- closed-source software developed by Scytl
- used previously in 2011, not without problems
- backed by the NSW electoral commissioner
- correctness specified in purchase contract

ACT Legislative Assembly: Trust through Open Source

- open-source software developed by Software Improvements
- published report on software design, bugs are acknowledged

Intermediate Level: Logical Specification



- Translating Logic → Text provides validation
- Know how to relate programs and specifications

How Trustworthy is This Approach?

Example: Hare-Clark in HOL

```
!seats cands ballots state cand rem_cands COUNT TRANSFER_TO.
(COUNT_HCT seats cands ballots = NEXT_STAGE state)
  /\ MEM cand cands
  /\ LENGTH (FIRSTS_FOR cand ballots) > QUOTA_VAR state
  ==> ?t pre post. (COUNT_HCT seats cands ballots = NEXT_STAGE pre)
    /\ (COUNT_HCT seats cands ballots = NEXT_STAGE post)
    /\ (TIME_VAR pre = t) /\ (TIME_VAR post = SUC t) /\ t > t0
    /\ !rcvr_pre rcvr_post transval.
     (FST rcvr_pre = FST rcvr_post)
      /\ (FST rcvr_pre <> cand) /\ MEM (FST rcvr_pre) cands
      /\ ~MEM (FST rcvr_pre) (ELECTED_VAR pre)
     (*a*)
      /\ (COUNT rcvr_pre = LENGTH (
     FILTER (($= (FST rcvr_pre)) o HD o
       (STRIP_BALLOT (EXCL_VAR pre) (ELECTED_VAR pre)))
     (FIRSTS_FOR cand ballots)))
     (*b*)
      /\ (transval =
        ((LENGTH (FIRSTS_FOR cand ballots) - QUOTA_VAR state) , LENGTH
        (FIRSTS_FOR cand ballots)))
      /\ VALID_TRANS_VAL transval
     (*c*)
      /\ (TRANSFER_TO rcvr_pre =
         (COUNT rcvr_pre * (FST transval)) DIV (SND transval))
     (*d*)
      /\ (SND (SND rcvr_post) = (transval
          , FILTER (($= (FST rcvr_pre)) o HD o
         (STRIP_BALLOT (EXCL_VAR pre) (ELECTED_VAR pre)))
         (FIRSTS_FOR cand ballots)
         , clause6)
     ::(SND (SND (rcvr_pre))))
    /\ (TOTAL_COUNT rcvr_post
         = TOTAL_COUNT rcvr_pre + TRANSFER_TO rcvr_pre)
    ==> !rem_cand.
     ((FST rem_cand = FST rcvr_pre
```

Australian National University

Higher-Order Logic as Intermediate Level

Text to Logic

- familiarity with HOL required

Logic to Implementation

- standard verification problem

Trust in People

- to ascertain the correctness of the logical specification
- to ascertain due dilligence of verification

Trust in Technology

- your hardware
- your compiler and your proof checker

Single Transferable Vote on a Single Slide

begin/1 :
$begin(S, H, U) \otimes$
$!(Q = U/(S{+}1) + 1)$
$\quad \multimap \{!quota(Q) \otimes$
$\qquad tally\text{-}votes(S, H, U)\}$

tally/1 :
$tally\text{-}votes(S, H, U) \otimes$
$uncounted\text{-}ballot(C, L) \otimes$
$hopeful(C, N) \otimes$
$!quota(Q) \otimes !(N{+}1 < Q)$
$\quad \multimap \{counted\text{-}ballot(C, L) \otimes$
$\qquad hopeful(C, N{+}1) \otimes$
$\qquad tally\text{-}votes(S, H, U{-}1)\}$

tally/2 :
$tally\text{-}votes(S, H, U) \otimes$
$uncounted\text{-}ballot(C, L) \otimes$
$hopeful(C, N) \otimes$
$!quota(Q) \otimes !(N{+}1 \geq Q) \otimes$
$!(S \geq 1)$
$\quad \multimap \{counted\text{-}ballot(C, L) \otimes$
$\qquad !elected(C) \otimes$
$\qquad tally\text{-}votes(S{-}1, H{-}1, U{-}1)\}$

tally/3 :
$tally\text{-}votes(S, H, U) \otimes$
$uncounted\text{-}ballot(C, [C' \mid L]) \otimes$
$(!elected(C) \oplus !defeated(C))$
$\quad \multimap \{uncounted\text{-}ballot(C', L) \otimes$
$\qquad tally\text{-}votes(S, H, U)\}$

tally/4 :
$tally\text{-}votes(S, H, U) \otimes$
$uncounted\text{-}ballot(C, [\,]) \otimes$
$(!elected(C) \oplus !defeated(C))$
$\quad \multimap \{tally\text{-}votes(S, H, U{-}1)\}$

tally/5 :
$tally\text{-}votes(S, H, 0) \otimes$
$!(S < H)$
$\quad \multimap \{defeat\text{-}min(S, H, 0)\}$

tally/6 :
$tally\text{-}votes(S, H, 0) \otimes$
$!(S \geq H)$
$\quad \multimap \{!elect\text{-}all\}$

defeat-min/1 :
$defeat\text{-}min(S, H, M) \otimes$
$hopeful(C, N)$
$\quad \multimap \{minimum(C, N) \otimes$
$\qquad defeat\text{-}min(S, H{-}1, M{+}1)\}$

defeat-min/2 :
$defeat\text{-}min(S, 0, M)$
$\quad \multimap \{defeat\text{-}min'(S, 0, M)\}$

defeat-min′/1 :
$defeat\text{-}min'(S, H, M) \otimes$
$minimum(C_1, N_1) \otimes$
$minimum(C_2, N_2) \otimes$
$!(N_1 \leq N_2)$
$\quad \multimap \{minimum(C_1, N_1) \otimes$
$\qquad hopeful(C_2, N_2) \otimes$
$\qquad defeat\text{-}min'(S, H{+}1, M{-}1)\}$

defeat-min′/2 :
$defeat\text{-}min'(S, H, 1) \otimes$
$minimum(C, N)$
$\quad \multimap \{!defeated(C) \otimes$
$\qquad transfer(C, N, S, H, 0)\}$

transfer/1 :
$transfer(C, N, S, H, U) \otimes$
$counted\text{-}ballot(C, L)$
$\quad \multimap \{uncounted\text{-}ballot(C, L) \otimes$
$\qquad transfer(C, N{-}1, S, H, U{+}1)\}$

transfer/2 :
$transfer(C, 0, S, H, U)$
$\quad \multimap \{tally\text{-}votes(S, H, U)\}$

[elect-all/1 :
$!elect\text{-}all \otimes$
$hopeful(C, N)$
$\quad \multimap \{!elected(C)\}$

Linear Logic as Intermediate Level

Text to Logic

- need to understand linear logic

Logic to Implementation

- automated proof search
- proofs are independently verifiable certificates

Trust in People

- to ascertain the correctness of the logical specification
- to understand and operate proof checker
- to ascertain correctness of proof checker?

Trust in Technology

- your hardware
- (proof checkers are verified to machine level)

Example: Domain-Specific Logics for STV

$$(\mathsf{Ax})\,\frac{}{(b,q,s) \vdash \mathsf{state}(u,a,t,h,e)}$$

- $u = b$, $a = \mathsf{nas}$, $t = \mathsf{nty}$, $e = []$
- h pairwise distinct, $C = \bigcup h$

$$(\mathsf{C1})\,\frac{(b,q,s) \vdash \mathsf{state}(u,a,t,h,e)}{(b,q,s) \vdash \mathsf{state}(u',a',t',h,e)}$$

- $\mathsf{eqe}((f{:}fs), u', u))$, $f \in h, t(f) < q$,
- $\mathsf{add}(f, f{:}fs, a, a')$ $\mathsf{inc}(f, t, t')$

$$(\mathsf{El})\,\frac{(b,q,s) \vdash \mathsf{state}(u,a,t,h,e)}{(b,q,s) \vdash \mathsf{state}(u,a,t,h',e')}$$

- $c \in h, t(c) = q, \mathsf{len}(e) < s$
- $\mathsf{eqe}(c, h', h)$, $\mathsf{eqe}(c, e, e')$

$$(\mathsf{Tv})\,\frac{(b,q,s) \vdash \mathsf{state}(u,a,t,h,e)}{(b,q,s) \vdash \mathsf{state}(u',a,t,h,e)}$$

- $f \notin h$
- $\mathsf{repl}((f{:}fs), fs, u, u')$

STV in Coq

```
Inductive Pf (b: list ballot) (q: nat) (s: nat) : Node -> Type :=

  ax :  forall  u a t h e,
  (forall c: cand, In c h) ->
  PD h ->
  u = b ->
  a = nas ->
  t = nty ->
  e = nbdy ->
  Pf b q s (state (u, a, t, h, e))
| c1 : forall u nu a na t nt h e f fs,
  Pf b q s (state (u, a, t, h, e )) ->
  eqe (f::fs) nu u ->
  In f h ->
  t f < q ->
  add f (f::fs) a na ->
  inc f t nt ->
  Pf b q s (state (nu, na, nt, h, e))
| el : forall u a t h nh e ne c,
  Pf b q s (state (u, a, t, h, e)) ->
  In c h ->
  t(c) = q ->
  length e < s ->
  eqe c nh h ->
  eqe c e ne ->
  Pf b q s (state (u, a, t, nh, ne))
| tv : forall u nu a t h e f fs,
  Pf b q s (state (u, a, t, h, e)) ->
  ~(In f h) ->
  rep (f::fs) fs u nu ->
  Pf b q s (state (nu, a, t, h, e))
| ey : forall u nu  a t h e,
  Pf b q s (state (u, a, t, h, e)) ->
  eqe [] nu u ->
  Pf b q s (state (nu, a, t, h, e))
| tl : forall u a t h nh e c,
  Pf b q s (state ([], a, t, h, e)) ->
  length e + length h > s ->
  In c h  ->
  (forall d, In d h-> t c <= t d) ->
  eqe c nh h ->
  u = a(c) ->
  Pf b q s (state (u, a, t,nh, e))
| hw : forall w u a t h e,
  Pf b q s (state (u, a, t, h, e)) ->
  length e + length h <= s ->
  w = e ++ h ->
  Pf b q s (winners (w))
| ew : forall w u a t h e,
  Pf b q s (state (u, a, t, h, e)) ->
  length e = s ->
  w = e ->
  Pf b q s (winners w).
```

Australian National University

Domain Specific Logics as Intermediate Level

Text to Logic

- basic maths knowledge suffices

Logic to Implementation

- provably correct proof search
- proofs are independently verifiable certificates

Trust in People

- to ascertain the correctness of the logical specification
- to implement and run a proof checker

Trust in Technology

- your hardware
- your compiler

Aspects of Trust

Width of trust base: pool of possible scrutineers

- Domain specific: all 1st year undergraduates
- Other approaches: need PhD in Logic

Height of trust base: technological degree of certainty

- Domain specific: trust hardware and compiler
- Other approaches: down to machine level

What should we aim for?

- large *social* trust base: many people can ascertain correctness?
- small *technological* trust base: just hardware?
- role of certificates? role of formalism?
- how much trust can we reasonably afford?