Counting Votes with Formal Methods

Bart Jacobs

in collaboration with
Engelbert Hubbers, Joseph Kiniry, and Martijn Oostdijk

Security of Systems Group
Department of Computer Science, University of Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
www.cs.kun.nl/sos

This abstract provides some background information about the electronic voting experiment that is planned in the Netherlands for the European Elections of 2004, and about our own involvement in the infrastructure for this experiment. The talk will elaborate further about the computer security issues involved, especially with respect to the use of formal methods for vote counting software.

Remote Voting

Since the late 1990s voting in the Netherlands proceeds largely via voting machines. These are dedicated computers that record and store votes. These machines are under control of local governments, who put them up in voting stations on election days. These voting machines (and all new versions of them) have undergone independent evaluation before being admitted. However the internal mechanics is secret. Also, the evaluation reports are not public. Nevertheless, at the time of introduction, these machines were uncontroversial. They have been used in several elections, without causing problems. Currently, such machines are the subject of much discussion, see for instance [2].

In 2002 the parliament of the Netherlands adopted a temporary law that went a step further than voting via computers at voting stations. The law allows experimentation with what is called location-independent voting. It has resulted in plans to allow voting via internet and phone in the 2004 elections for the European Parliament. This involves a one-time, limited experiment, largely intended to explore the possibilities and to gather experience with the required techniques and procedures.

Low-Tech Approach

These electronic elections are set up for expatriats. They already have the possibility to participate in elections via voting by (ordinary) mail. To keep things simple, the approach in the electronic elections is modeled after this voting by mail. Hence, participants in the electronic elections are required to register explicitly in advance. Upon registration, they have to submit a copy of their pasport and provide a self-chosen pincode, for authentication.

The whole organisation is fairly low-tech, and involves various codes for voter identification & authentication, and for candidate selection. In total, thousand different ballots (with different candidate codes) will be distributed randomly, in order to ensure confidentiality.

C. Rattray et al. (Eds.): AMAST 2004, LNCS 3116, pp. 21–22, 2004.

The complicated registration procedure thus prevents national adoption of this approach. And because there is no national electronic identity card (yet), more high tech, crypto-based authentication procedures are not an option.

Organisation

The plans for these electronic elections have been elaborated by the Ministry of Internal Affairs, mostly in 2003. There has been an open bidding to build the software for these elections, and to run it as a service. This bidding has been won by LogicaCMG. Also, a panel of independent experts has been set-up, for feedback. The main advice from this panel[1] was to run the project as open as possible, and to compartimentalise it maximally, so that fraud is difficult without cooperation of several parties. The Ministry owns the copyright on the software, and organises its own evaluations – again by several parties. For instance, our group has participated in an evaluation of the robustness of the webservers, during an experiment in nov. 2003. Also, the intention is to make the source code available for inspection[2], but it will probably not appear on the internet, like earlier in Australia[3].

Vote Counting

Late into the project the Ministry decided to open another separate, much smaller bid for the counting of the votes. It has been won by our group, on the basis of a proposal that involves annotating the Java source code with correctness assertions from the Java Modeling Language JML [1]. These annotations are checked both with the runtime assertion checker [4] and with the newest version of the Extended Static Checker, ESC/Java 2 [3], developed in part by our group. Details will be in the talk.

References

1. L. Burdy, Y. Cheon, D. Cok, M. Ernst, J. Kiniry, G.T. Leavens, K.R.M. Leino, and E. Poll. An overview of JML tools and applications. In Th. Arts and W. Fokkink, editors, *Formal Methods for Industrial Critical Systems (FMICS'03)*, number 80 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2003.
2. D.L. Dill, B. Schneier, and B. Simons. Voting and technology: Who gets to count your vote. *Commun. ACM*, 46(8):29–31, 2003.
3. ESC/Java2. Open source extended static checking for Java version 2 (ESC/Java 2) project. Security of Systems Group, Univ. of Nijmegen `www.cs.kun.nl/sos/research/escjava/`.
4. G.T. Leavens, Y. Cheon, , C. Clifton, C. Ruby, and D.R. Cok. How the design of JML accommodates both runtime assertion checking and formal verification. In F. de Boer, M. Bonsangue, S. Graf, and W.-P. de Roever, editors, *Formal Methods for Components and Objects (FMCO 2002)*, number 2852 in Lect. Notes Comp. Sci., pages 262–284. Springer, Berlin, 2003.

[1] Which included the current author.

[2] At the time of writing the details are still unclear, but inspection will probably require an explicit request/registration.

[3] See `www.elections.act.gov.au/EVACS.html`