AdaCore CASE STUDY

# Software Improvements: Using Ada to Implement a Secure Electronic Voting System

adacore.com

## Summary

**Electronic election systems need to be demonstrably secure in order to ensure the protection of votes, the privacy of voters, and the prevention of interference. That's why Software Improvements selected the Ada 2012 programming language and AdaCore's GNAT Pro Ada development environment to upgrade the Australia Capital Territory's Electronic Voting & Counting System (eVACS®), a public-facing system that demands a high level of security.**


## The Role and Importance of Election Systems

In democratic societies, election processes are of paramount importance. They translate individual voter choices into collective decisions, requiring high-level software engineering standards and precision in every detail.

Reliability, transparency, and security are all essential characteristics due to the high stakes the public has in voting system integrity. Even a minor vulnerability can erode trust in the entire electoral process; choosing the right software development technology (in particular, a programming language that is intrinsically safe and secure) is critical.

In the past, many electronic voting systems have struggled with essential security requirements - including vote protection, privacy maintenance, and interference avoidance. However, one electronic voting and counting system used in the Australian Capital Territory (ACT) has stood out as a notable exception - eVACS®.

## The Evolution of the eVACS® System

The Electronic Voting & Counting System (eVACS) was originally developed in 2001 by Software Improvements Pty Ltd of Canberra, Australia, for the ACT Electoral Commission, known as Elections ACT, and specifically for the Australian Capital Territory's 2001 Legislative Assembly Election. The original system comprised three main components: Electronic Voting, Data Entry of Paper Votes, and Counting (using the Hare Clark algorithm[1]).

Since its inception, eVACS has undergone substantial modifications in line with the evolving requirements of modern elections. The progression from its foundational model in 2001 to today's contemporary structure illustrates the adaptability of both the system and its development team.

**Initially, the backbone of the eVACS system was implemented in the C programming language. In 2006, however, Software Improvements transitioned to Ada to leverage the robustness and adaptability intrinsic to the language as well as its rich array of features that are aligned with Software Improvements' usage of model-driven development.**

**Customer:**
Software Improvements Pty Ltd offers services that support high-integrity systems and software development.

**Challenge:**
To upgrade the Electronic Voting & Counting System (eVACS) software used by the Elections ACT for Legislative Assembly Elections in the Australian Capital Territory and deliver a more secure voting system.

**Solution:**
Software Improvements selected Ada 2012 and AdaCore's GNAT Pro Ada development environment for the eVACS upgrade.

**Result:**
By using Ada 2012, Software Improvements was able to upgrade the eVACS platform to provide easier setup, voting, and measurement capabilities, as well as to mitigate evolving security vulnerabilities.

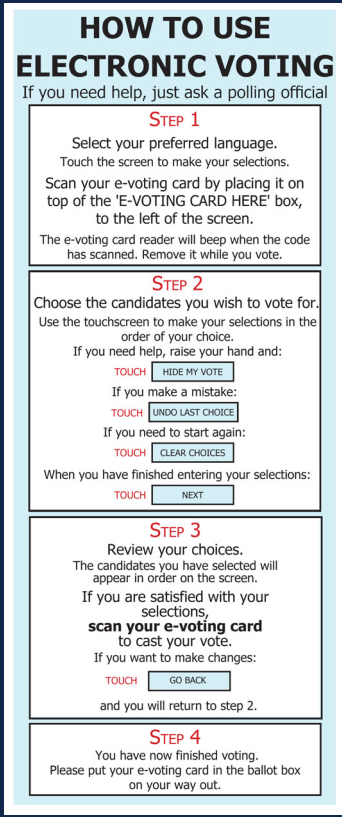
[1]Hare Clark is typical of a multi-member, preferential election system - probably unheard of in most other jurisdictions. In the USA it might be called a multi-member, instant runoff election.

Dr. Clive Boughton, Principal Consultant at Software Improvements Pty Ltd, illustrates what was at stake: “Ada 2012 was chosen as the main programming language for eVACS® 2020 because most other mainstream languages do not possess adequate built-in constructs to easily enable the safe/secure operations we needed.”

This change involved designing a flexible model for running elections using a specific system design approach (the Shlaer-Mellor methodology) and then creating an entirely new tool that could turn these system designs into usable computer code in the Ada programming language. This pioneering initiative exhibited substantial versatility, demonstrating compatibility with any structured model. This revamped system made its debut in the 2007 Australian Federal election and has been proving its value across multiple election events ever since.

“It’s a much easier thinking process to go from a model into Ada than it is from a model into C or, any other lower-level language,” said Dr. Boughton.

The eVACS evolution didn’t stop there. In 2019, an adaptation of the Ada code reflecting the latest system model was undertaken, prompted by some new requirements. The system’s resilience and adaptability were quickly evident; for example, the user interface was updated without difficulty, transitioning from a mobile keyboard to a touchscreen, while preserving keyboard accessibility for visually impaired users. These advances were made while preserving a steadfast focus on reliability, security, and user-friendliness.




**“Ada 2012 was chosen as the main programming language for eVACS® 2020 because most other mainstream languages do not possess adequate built-in constructs to easily enable the safe/secure operations we needed.”**

**- Dr. Clive Boughton, Principal Consultant at Software Improvements Pty Ltd**

## Security Measures in the eVACS System

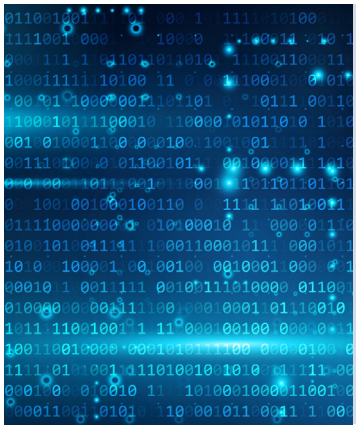
Given its crucial role, eVACS calls for a rigid security apparatus to ensure people’s trust in voting. To this end, the Software Improvements Pty Ltd team has crafted a diverse portfolio of strategies to fortify voting integrity and system credibility.

Operating as a closed system, eVACS is resilient to external threats. A cornerstone of this security architecture involves the distribution of eVACS via a fortified USB that, upon insertion, purges all pre-existing data (including the operating system) and permits the installation of eVACS alone, preventing unauthorized intrusion or outside meddling.

Further security layers include the issuance of unique barcodes to voters opting for electronic voting, thus preserving the secrecy of the ballot and curbing unauthorized access. Within polling locations, servers are securely locked away in cabinets, prohibiting access to the executing system and vote database, thereby nipping potential system interference in the bud. All operations on servers are purely menu-driven, and a two-factor authentication protocol is deployed, further shoring up security within this hermetically sealed system.

As Dr. Boughton stated, “We deployed the second Ada iteration in the 2007 Australian federal election, creating an incredibly reliable system. Yet, few grasped why it was so robust or effective. There was a misconception that it was inherently superior simply because it was computerized. People were often unaware of the potential electronic vulnerabilities. To ensure the reliability, we engineered a model compiler (written in Ada) that translates a model into Ada.”

The potent security measures ingrained within the eVACS system, crafted using the safety- and security-centric Ada programming language, have enabled the Software Improvements Pty Ltd team to uphold trust and integrity in the electoral process amidst the rising tide of cyber threats.



> **A strong, executable modeling language, like Shlaer and Mellor's, emphasizes data, state, and behaviour. It allows for early architectural planning, efficient change management and model verification before translation. Combined with Ada, it enables the integration of security and safety directly into system design, simplifying maintenance and adaptation, thereby significantly reducing workload and enhancing system robustness.**
>
> **- Dr. Clive Boughton,**
> **Principal Consultant at Software Improvements Pty Ltd**

## The Interplay of Modeling and Ada in System Development

The birth and evolution of the eVACS system were contingent on two fundamental elements: the skill of modeling, and Ada's precise semantics. Modeling served to untangle the intricate architecture of the system, carving out domains as standalone components, thereby easing the path of development. In their endeavor to simplify the process, the team engineered a model compiler for Ada, leading to a significant reduction in workload and maintenance effort.

The synergy between Ada and the Shlaer–Mellor modeling method proved instrumental. Ada's support for modularization (packages), object orientation, and concurrency is a natural match for the building blocks of Shlaer Mellor modeling (such as class diagrams, state transition diagrams, and precise action language), simplifying domain incorporation. Leveraging Ada's rich type-system, concepts at the model level were directly translated, unlike in C, expediting the transition from design blueprints to implementation.

"A strong, executable modeling language, like Shlaer and Mellor's, emphasizes data, state, and behavior. It allows for early architectural planning, efficient change management and model verification before translation," said Dr. Boughton. "Combined with Ada, it enables the integration of security and safety directly into system design, simplifying maintenance and adaptation, thereby significantly reducing workload and enhancing system robustness."

**Ada excels in implementing complex systems where reliability, maintainability, and efficiency are critical. Its constructs make it easier to catch errors early, and its syntax makes the code readable even to non-specialists. Unlike other languages, where errors such as buffer overruns can persist in operational code, Ada incorporates continuous verification as a design principle.**

## The Utility and Efficacy of Ada

With its efficient management of complexity, readable syntax, and built-in verification capabilities, Ada proved to be the ideal choice for a high-integrity system like eVACS. Its logical structure offers a solid basis for teaching fundamental programming principles. Consequently, programmers trained in Ada adopt a method of logical thinking, enhancing their adaptability in a technologically varied workplace.

Dr. Boughton notes that, although Ada is a large language, it is easy to learn, and its design helps programmers to focus on task-specific constructs, effectively simplifying what may seem intricate. For example, a member of his team who was unfamiliar with Ada was able to understand and adapt the system to the updated 2019 requirements in just two months.

"Ada has evolved and has become more sophisticated," he explains. "However, there's no need to grasp it in its entirety. Concentrate on the aspects that are pertinent to your tasks. It's not necessary for every system to utilize every Ada feature."

Furthermore, Ada's strengths are especially useful in the post-development stages. It encourages a robust system architecture, helping developers manage software complexity and making maintenance more efficient.

## Adaptability Unveiled: The Ada Advantage

The eVACS path from 2006 to 2019 demonstrates Ada's benefits in supporting the product's evolution. The initial development of the model compiler required two dedicated personnel. A decade later, the upgrade was quick and painless: Ada made it possible to adapt the original code to match the updated requirements, avoiding the need to start afresh.

"Indeed, we have a solitary coder, not an Ada expert, but one who is versed in its style," explained Dr. Boughton. "Ada's clarity helped him swiftly

> **Several academic groups actively argue against electronic voting systems. It was incumbent upon us to demonstrate the potential for electronic systems to be well executed and secure.**
>
> **- Dr. Clive Boughton,**
> **Principal Consultant at Software Improvements Pty Ltd**

understand what was needed. He pruned and reworked the existing code, constructing an operational system within a few months. A testament to Ada's efficacy."

The system's interface underwent a major upgrade. Initially, eVACS limited navigation across the ballot to column-by-column and candidate-by-candidate. With the new version, and thanks to Ada's many benefits, the system has overcome these limitations.

**The eVACS ease of adaptation demonstrates Ada's true potential: a language that facilitates refactoring, promotes quick learning, and offers a seamless transition in upgrading a software codebase to meet new requirements—a distinct advantage in meeting the rigorous demands intrinsic to building high-integrity systems.**

## The Impact and Success of the eVACS System

The eVACS success is evidenced by the increasing adoption rates, ironclad reliability, and transformative impact on the voting process. From its modest origins, serving a niche segment of the population, the system has now become the voting medium of choice for 70% of voters in the Australian Capital Territory, demonstrating the public's faith in the system's integrity.

Since its revamp in 2006, eVACS has proven its worth across approximately five election cycles, revealing its robustness and dependability. Its debut triggered a paradigm shift in the voting process, marking a departure from the traditional paper ballots to an electronic format while ensuring that its design caters to everyone, including those with disabilities.

But the influence of eVACS extends beyond the voting booth. It has paved the way for future advances in secure, high-integrity software systems and stands as an example of the many benefits of the Ada programming language. Dr. Boughton expresses this sentiment aptly. "Several academic groups actively argue against electronic voting systems. It was incumbent upon us to demonstrate the potential for electronic systems to be well executed and secure,"

With this mindset, the Software Improvements Pty Ltd team has shown eVACS to be a pioneering innovation in the realm of electronic voting systems, with a future of sustained growth.

## Looking Ahead: Ada's Continued Role and Promising Future

Dr. Clive Boughton and his team at Software Improvements Pty Ltd will be using Ada to streamline the eVACS system in several areas. A key focus is the vote counting process, originally implemented in C and later migrated to SQL, which is scheduled for a transition to Ada to exploit its ability to implement critical software.

> **Unquestionably, plans for future system enhancements are in motion, I'm eager to work toward a 'universal election model' using Ada and tailor it for commercial viability. To me, this language truly delivers.**
>
> **- Dr. Clive Boughton,**
> **Principal Consultant at Software Improvements Pty Ltd**

Based on user feedback and evolving electoral demands, eVACS will likely see the need for further enhancements. Continuing the role they played in making eVACS a success, Ada and AdaCore's tools will be a major factor in helping Dr. Boughton's team meet these new challenges. For systems that need to be reliable, secure, efficient, and maintainable, Ada is the right choice.

AdaCore

adacore.com