



# Ada: The Maginot Line of Languages

-or-

**One language to rule them all, One language to find them,  
One language to bring them all and in the darkness bind them.**

(with apologies to J.R.R. Tolkien)



During World War I, more than one million French citizens were killed, and another estimated four to five million were wounded. Many French politicians and generals thought that the Treaty of Versailles (which ended the war, and was supposed to punish the defeated countries and prevent further conflict) was insufficient protection. France was justifiably concerned that the treaty was really just an armistice and that war would ultimately resume (as it did – World War II). To protect France, many influential politicians and generals were in favor of an aggressive set of fortifications. There were many studies and meetings, and based on the consensus of opinion, the Maginot Line was built.

The Maginot Line, named after French minister of defense André Maginot, was a line of concrete fortifications, tank obstacles, machine gun posts, and other defenses which were built along the Italian and German border. The French thought that these fortifications would slow down attacking forces, allowing the French time to respond. Two places the Maginot line did not extend were the Ardennes Forest (which was thought impassable) and the Belgium border, because Belgium and France had recently signed an alliance.

When World War II began, the Germans did not view the Ardennes Forest as impenetrable. More than a million troops and 1,500 tanks crossed Luxembourg, Belgium and then moved straight through the Ardennes. On May 10, 1940, the German advance started. The French government had to abandon Paris on May 13. The conquest was swift and decisive.

History has sometimes viewed the Maginot Line as something that was ineffectual. However, this viewpoint, in my opinion, is vastly incorrect. The Maginot Line did exactly what it was supposed to do – prevent a direct attack upon France’s Eastern border. The few places upon the Maginot Line that were directly attacked by German troops held out well. The concept was sound, the execution was just incomplete. There is a history lesson to be learned here.

The theme of this issue is Ada 2005. Now, for those of you who don’t know me, I’m an Ada zealot. I taught one of the first U.S. Air Force-approved Ada training courses back at Keesler, AFB in 1984. I taught Ada at the Air Force Academy starting in 1986.

Back in the 1980s, there were literally hundreds (possibly thousands) of programming languages running around. Every defense program and contractor used their own language (or variation of a language). Most projects were in assembly language of some type, making projects hard to maintain and upgrade. The initial vision of Ada was to provide a common high-order programming language that would allow Department of Defense (DoD) software that was cheaper and quicker to develop and easier to maintain. Ada can be described as a language that has facilities for real-time response, concurrency, hardware access, and reliable run-time error handling. In support of large-scale software engineering, it emphasizes strong typing, data abstraction and encapsulation. Nothing bad in this list – in fact, everything in this description sounds pretty good, doesn’t it? So good, in fact, that back in 1983, Richard DeLauer, then Under Secretary of Defense for Research and Engineering, sent out a memo directing that:

The Ada programming language shall become the single common programming language for Defense mission-critical applications. Effective 1 January 1984 for programs entering Advanced Development and 1 July 1984 for programs entering Full-Scale Engineering Development, Ada shall be the programming language.

The problem was that back in 1983 there weren’t many compilers, tools, or experienced programmers. Compilers were slow and tended to consume all the resources of even high-end computers. The general feeling among us Ada zealots was that the DeLauer memo was premature and actually worked against the cause of Ada. Because of the lack of tools, compilers, and trained programmers, many developers either received a waiver from the Ada mandate or simply ignored the memo. Sort of like the Maginot line – folks just went around it.

However, time has been good to Ada. It has been updated several times, and the actual intent of the DeLauer memo (that high-level languages be used to develop DoD software) has long since been met. Back in the 1980s, as I said, there were literally hundreds of languages being used. Today, most software is created using a relative few languages. C++ (nobody uses C anymore) and Java are probably most used, and according to trends, C++ usage is going down while Java is on the rise; Java provides almost all of the same safety features (strong typing, data abstraction, encapsulation). There are quite a few of us who hold the opinion that Ada strongly influenced Java – and that Java has C++ syntax, but Ada semantics. Ada is still widely used outside of the United States, and Ada is used worldwide in the avionics industry.

Ada is still a viable force in avionics simply because it’s very good at what it was designed to do – provide high-quality code in safety-critical environments. It has run-time features such as real-time and parallel processing that are hard to find in any other language.

It’s all about safety and security – the same things that the Maginot Line was designed to give. And, just like the Maginot Line, it all lies in the execution.

— David A. Cook, Ph.D.

The AEGIS Technologies Group, Inc.

<dcook@aegistg.com>

## Additional Reading

1. Much of this research comes from <<http://europeanhistory.about.com/library/weekly/aa070601a.htm>> and <[http://en.wikipedia.org/wiki/Maginot\\_Line](http://en.wikipedia.org/wiki/Maginot_Line)>.
2. See <[www.people.ku.edu/~nkinners/LangList/Extras/langlist.htm](http://www.people.ku.edu/~nkinners/LangList/Extras/langlist.htm)> for a list of more than 3,000 languages.
3. See <<http://oop.rosweb.ru/>> under Language List, then Ada.
4. See “Evolutionary Trends of Programming Languages.” This excellent article can be found at <[www.stsc.hill.af.mil/crosstalk/2003/02/schorsch.html](http://www.stsc.hill.af.mil/crosstalk/2003/02/schorsch.html)>.
5. “An Empirical Study of Programming Language Trends,” Dios et. al., IEEE Software, May/June 2005.