Experiences with Ada

in the safety-critical communication & ground control systems of the **Neuron UCAV**

DEFENCE AND SPACE

L. Pabón, A. Jiménez, and J. M. Martínez

12-16 June 2017

AIRBUS

The Neuron UCAV

A brief description of the technological context

The Neuron UCAV

The **Neuron** is an experimental *Unmanned Combat Aerial Vehicle* (UCAV) developed with international cooperation of France, Greece, Italy, Spain, Sweden and Switzerland.

The project is led by **Dassault Aviation**, with the collaboration of the aerospace industry partners **Alenia** (Leonardo), **EADS-CASA** (Airbus), **HAI**, **RUAG**, **Saab** and **Thales**.

Watch video at:
https://youtu.be/n9EJ-WAK4lw

The Neuron UCAV (cont.)

The main objective of the program, launched in 2003, was to acquire the necessary knowledge and experience for future UAS programs, by developing a Pan-European large size, stealth, autonomous UCAV platform to demonstrate the maturity and the effectiveness of technical solutions.

The aircraft made its maiden flight in December 1^{st} 2012, and more than 100 flights have been performed since then (and still ongoing) to test the capabilities of the system.

Watch video at:
https://youtu.be/n9EJ-WAK4lw

Aircraft specifications

General characteristics:

- Crew: 0
- Length: 9,5 m (31 ft 2 in)
- Wingspan: 12,5 m (41 ft 0 in)
- Empty weight: 4.900 kg (10.803 lb)
- Gross weight: 7.000 kg (15.432 lb).
- Power plant: 1 × Rolls-Royce/Turboméca Adour/Snecma M88
  40 kN (8.992 lbf) thrust

Aircraft specifications (cont.)

Performance:

- Maximum speed: 980 km/h (609 mph)
- Service ceiling: 14.000 m (45.900 ft)

Armament:

- 2 × 230 kg (500 lb) guided bombs

Stealth:

- Low radar profile due to the delta wing shape and special coating.

Ada-powered systems contributed by Airbus Defence & Space

Ada is a key part of the embedded, real-time, safety-critical systems contributed by former EADS-CASA (now Airbus Defence & Space) to the Neuron program:

- The ***Data Link Management Software*** **(DLMS)** is the subsystem in charge of managing the communications between ground and air segments, controlling the transmission and reception of information (control, telemetry and video) through the data links.

Ada-powered systems contributed by Airbus Defence & Space (cont.)

- The ***Global System Monitoring & Control* (GSMC) *Commands & Health Monitoring*** are two ground HMI subsystems in charge of the command and control of the air vehicle, the collection and display of real-time telemetry, and the safety validation of the flight plan.

The software (>300.000 SLOC) is developed to be certifiable according to RTCA/DO-178B up to assurance levels D and C respectively.

Hardware and platform context

The DLMS and GSMC subsystems are part of an ***Integrated Modular Avionics*** **(IMA)** architecture, and built upon the following elements:

- Single board computers based on PowerPC Freescale™ processors.
- Wind River VxWorks® 653 real-time operating system.

Hardware and platform context (cont.)

- Multiple and heterogeneous I/O interfaces:
  - Serial
  - Discrete lines
  - Ethernet
  - Flash file system
  - Video graphics
- Board support packages (BSP) and device drivers provided by the hardware manufacturer.
- ARINC 653 partitions & API for space and time segregation.

Operational context

The DLMS and GSMC operational requirements entail the software dealing with **functional** and **non-functional** aspects such as:

- Communicating and synchronizing with multiple stakeholders.
- Generating and displaying “complex” graphical HMI.
- Performing computationally intensive geodetic calculations.

Operational context (cont.)

- Handling different messaging models (including the exchange of batched non-atomic information).
- Adjusting to a prioritized preemptive scheduling policy, with consistent timing, sizing and throughput.
- Strict synchronous timings for certain processes, while allowing the entry and *rendezvous* of asynchronous tasks without interference.
- Being robust against external or environmental failures.

The Ada contribution

Profile and impact of Ada in the Neuron DLMS and GSMC

The chosen Ada route

Ada development **toolset**:

- AdaCore GNAT Pro High-Integrity Edition for ARINC 653.
- Wind River™ VxWorks 653 Workbench.
- IBM® Rational® Rhapsody® for Ada.
- VectorCAST/Ada™.

The chosen Ada route (cont.)

Rather **conservative approach** and restrictive Ada playground:

- Ada 95 compatibility.
- Run-time libraries limited to GNAT Pro *Zero Footprint* and *Cert* profiles.
- Compiler options mostly tuned for safety vs. performance.
- Rules enforced with global pragmas and/or verified with GNATcheck.

The chosen Ada route (cont.)

- Coding standard as a selection of rules and recommendations from:
  - Global directives for software development within the Neuron program.
  - ISO/IEC-TR-15942 guidelines.
  - MC/DC alleviation considerations.
  - Moderation of complexity metrics.
  - APEX API (No Ada tasking features).

Sticking to a *bare-bones* Ada profile

As a result of all the **safety-oriented** and determinism-seeking decisions, the range of allowed language features gets pretty constrained:

- No dynamic memory allocation (except during initialization).
- No allocators.
- No subprogram access.
- No unchecked access.
- No discriminated records.
- No class-wide operations.

Sticking to a *bare-bones* Ada profile (cont.)

- No generic units.
- No unconstrained objects (secondary stack).
- No streams.
- Stack size limitations.
- Enforced variable initialization.
- Regulated propagation & capture of predefined exceptions.
- No GOTO statements (or semantic equivalences).
- …

Other software challenges and demanding activities

Beyond safety assurance, other **technical topics** have also had special relevance during the development:

- Integration on ARINC 653 partitioned environments:
  - Quite painless due to AdaCore GNAT Pro APEX & high-integrity features.

- Geodetic calculations on large data sets:
  - Computationally-intensive algorithms required the dimensioning of memory structures and execution times based on the worst case scenario.

Other software challenges and demanding activities (cont.)

- Generation of graphical interfaces (HMI)
  - Seamless bindings & wrappers of the underlying APIs allow full control of the OpenGL-SC constructions to be handled from the Ada side.
- Drivers in polling mode (no interruptions):
  - Some performance headaches, which required exigent optimizations.
- 100% structural coverage achievement requirements:
  - No unjustified dead/deactivated/inline code is allowed.
  - Full bidirectional traceability between requirements, design, code and V&V.

© Dassault Aviation

Perceived advantages of the Ada choice

A good balance between **low-level control** and **safety vigilance**:

- It allows to define rich & safe type taxonomies.
- It favors consistent data typing and integrity across the projects.
- It encourages well-defined arithmetic conventions, precision and accuracy.
- It gives natural control over physical and functional ranges.

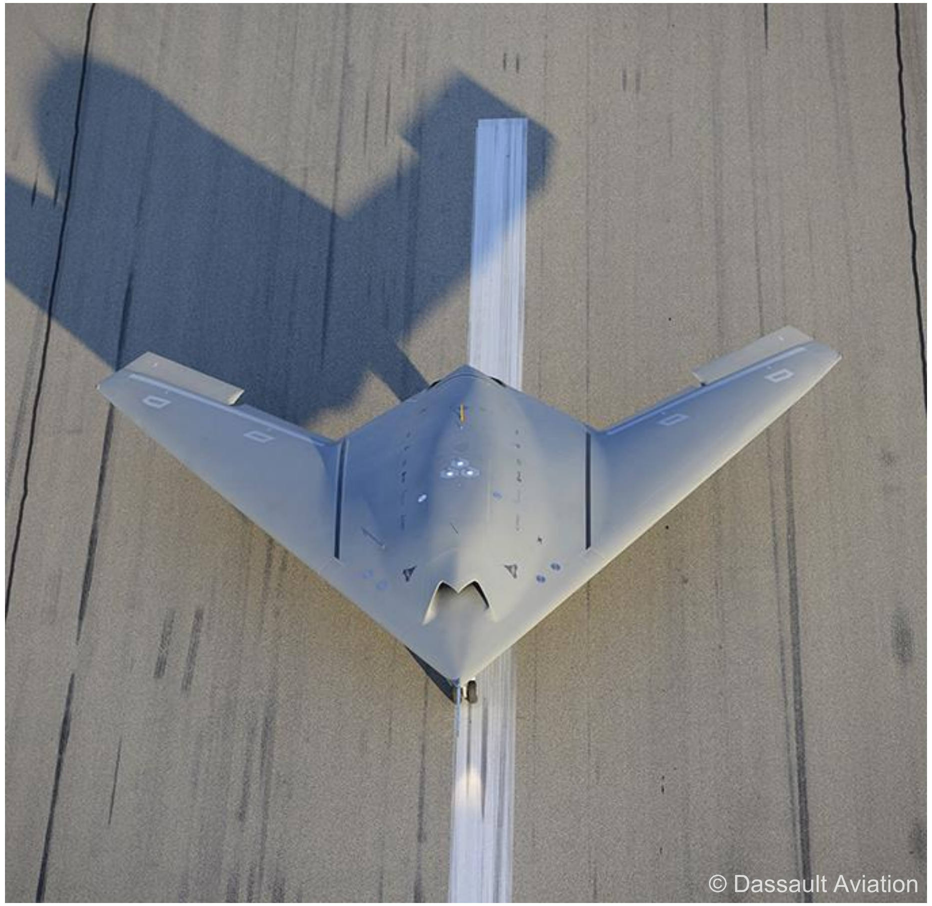
Perceived advantages of the Ada choice (cont.)

- It enforces obedience to data and buffer boundaries.
- It provides good control and warnings about data memory alignments.
- It helps to hold back the impact of I/O data loss, corruption & delay.
- It eases error prevention, detection and fault-tolerance.
- It seamlessly binds with foreign drivers, libraries and APIs.
- Long-term source code maintenance has been quite comfortable.

Perceived advantages of the Ada choice (cont.)

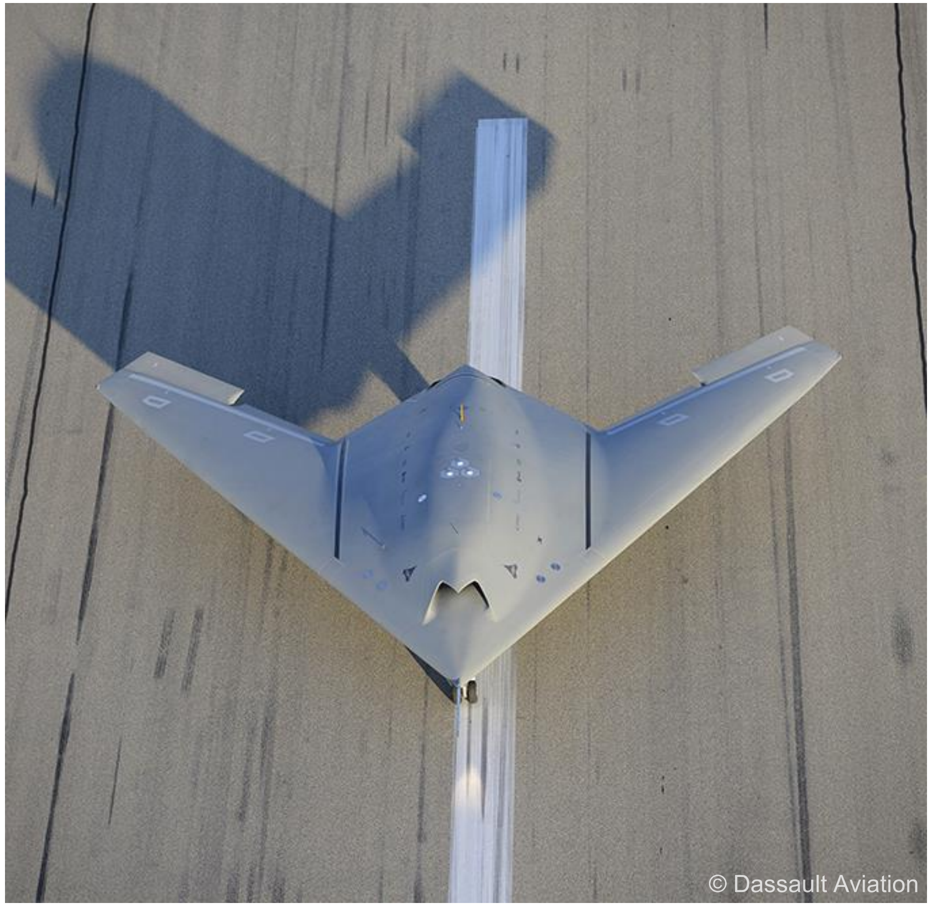
Didactical expressiveness of the language:

- Gentle slope learning curve for Ada-inexperienced developers.
- It helps in communicating the intention and the form.
- It assists in shaping clear and unequivocal interfaces.
- It decreases propensity for unseen code errors.
- It facilitates peer code reviews.



Perceived advantages of the Ada choice (cont.)

- It helps in transmitting good confidence to the certification authorities about the containment of safety risks and the compliance of rules and constraints at the coding phase.
- It tends to be verbose (especially if no *renames* are allowed), but it pays off due to all the previous virtues.
- Coding obstacles or efforts usually came from self-imposed restrictions, and not from the Ada semantics and features.



A few drawbacks too

Ada still seems to stand as a niche option, so we found some disadvantages regarding the **maturity or availability of tools** (in comparison with the support for other programming languages):

- Design and source code round-trip engineering not working properly on the (by then) available version of Rhapsody® for Ada.
- Several difficulties during the initial setup of unit testing and structural coverage analysis with the early versions of VectorCAST/Ada™.

A few drawbacks too (cont.)

- Support to Ada in popular continuous integration platforms was still marginal and limited to basic features.
- We missed the availability of a few advanced tools that have been present for some time in other environments (such as DSM tools for dependency analysis, architecture and code refactoring).
- It is not so easy as for other languages to build *relaxed* parser-based tools to aid in tasks such as automated traceability or documentation.

Overall assessment and evolution

The general experience with Ada in the Neuron project has been **satisfactory**, as Ada has been perceived neither as a risk nor as a concern, but as a solid foundation and a lifeline in many aspects.

Ada still is the **main choice for subsequent projects**, with more reason now that existing support has improved its maturity, and new interesting tools are joining the party.

Overall assessment and evolution (cont.)

After the Neuron, our Ada playground has **evolved** towards:

- The use of more modern versions of the compilers and tools.
- The adoption of broader and more flexible language features.
- The definition of better templates for project structures.
- The refinement of coding standard rules.
- Reusability, by writing less code while improving portability between:
  - Big/Little endian architectures.
  - Different types of computer processors (including multi-core).
  - Different operating systems (including emulated ARINC 653 support).

Thank you

AIRBUS