EYE IN THE SKY

A small engineering team designed, verified, generated and integrated 125,000 lines of code to control an unmanned aerial system using ANSYS SCADE in one-third the time required had the code been written in C.

By Giuseppe Cinà, Flight Control Systems Manager, Piaggio Aero Industries, Genoa, Italy
Amar Bouali, V.P. South Europe, Turkey, MEA Operations, Esterel Technologies, a wholly-owned subsidiary of ANSYS

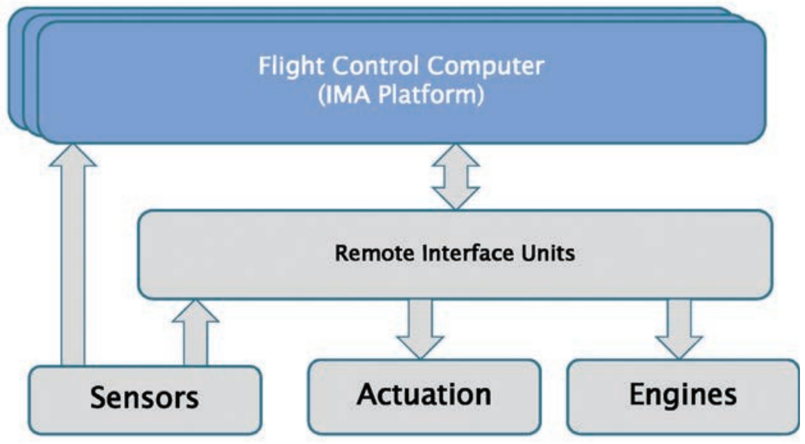
▲ P.1HH HammerHead shown in first flight

The use of unmanned aerial systems (UASs) for intelligence, surveillance and reconnaissance (ISR) missions has shown explosive growth. As their value continues to be demonstrated, this growth shows no sign of slowing. The UAS sector must address a number of key technical and manpower challenges in developing autonomously controlled aircraft. Engineers from Piaggio Aero faced the challenge of transforming the company's conventional manned P.180 Avanti II executive jet into a UAS. The vehicle command-and-control architecture needs to be certified against first-generation requirements while supporting a design road map that foresees growing functionality to support different configurations. This job had to be done with a strictly controlled number of engineers to limit overhead and succeed in a very short time. Piaggio engineers accomplished these goals with a new development process in which ANSYS SCADE models were created from scratch, or, if Matlab/Simulink® models were available, they were translated via the SCADE Suite Gateway for Simulink®. From the SCADE model, the embedded source code was generated automatically with the SCADE KCG qualified code generator. The vehicle control and management system (VCMS) — the digital infrastructure performing aircraft command and control — was tested continually, first at the model phase, then on the host, and

The first flight of the aircraft was successfully completed less than two years after the project began.

finally in the target environment so that the team could identify problems and correct them at the earliest possible point.

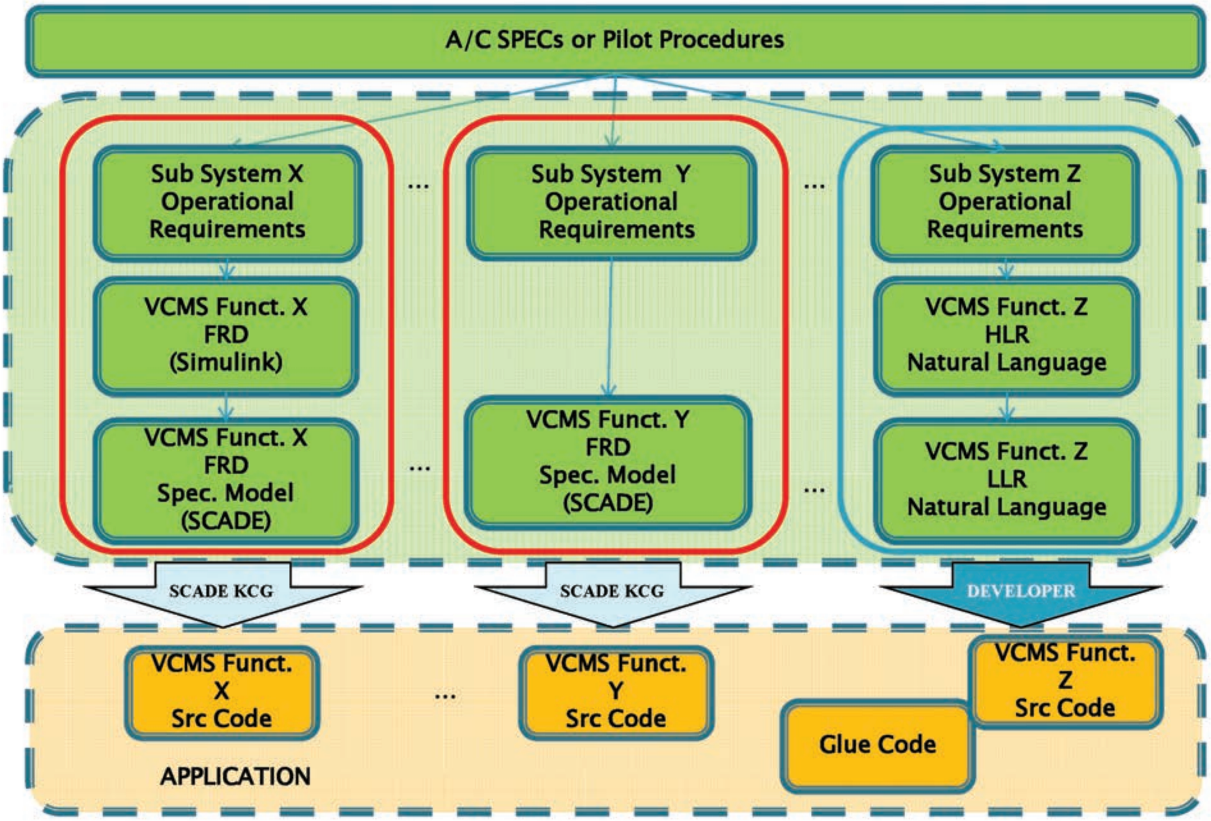
Piaggio Aero Industries S.p.A. is a multinational aerospace manufacturing company headquartered in Genoa, Italy. It designs, develops, constructs and maintains aircraft, aero-engines and aircraft structural components. Powered by two Pratt & Whitney Canada PT6-66B turboprop engines, Piaggio's newly developed P.1HH HammerHead will provide sophisticated standoff (deployed at a distance) capabilities for any surveillance and security need. The VCMS manages flight control, propulsion, electrical power generation and distribution, landing gear, braking, ice detection and protection, navigation and communications systems. Partitioning techniques were used to create a segregated environment in which software applications of each function run without interfering with each other to avoid propagating failures.



VCMS architecture

DEVELOPING SOFTWARE REQUIREMENTS

In the first months of the project, the team developed the engine and flight control laws; it also created the other requirements for the embedded software. High-level requirements for the VCMS were available in several different formats. Systems engineers collected some requirements in textual form as functions, interfaces and redundancies. Other requirements were captured in text from operating manuals such as the P.180 Pilot Operational Handbook. Control laws, algorithms and equations involved in flying the plane were written, simulated and validated in MathWorks® Simulink. Requirements were generated in the IBM®


A/C SPECs or Pilot Procedures
Sub System X Operational Requirements
VCMS Funct. X FRD (Simulink)
VCMS Funct. X FRD Spec. Model (SCADE)
...
Sub System Y Operational Requirements
VCMS Funct. Y FRD Spec. Model (SCADE)
...
Sub System Z Operational Requirements
VCMS Funct. Z HLR Natural Language
VCMS Funct. Z LLR Natural Language
SCADE KCG
SCADE KCG
DEVELOPER
VCMS Funct. X Src Code
...
VCMS Funct. Y Src Code
Glue Code
VCMS Funct. Z Src Code
APPLICATION

P.1HH development process

Rational® DOORS® requirements management environment. Test cases were also written in DOORS and linked to operational requirements using the SCADE Requirement Management Gateway. For each test case, test steps and expected results were defined.

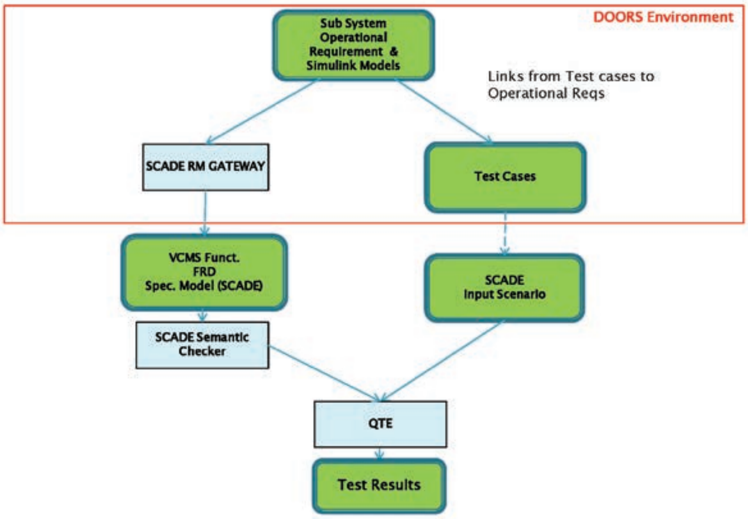
For this project, the software must comply with DO-178B, the de facto standard used to qualify all avionics software by the FAA, EASA and other certification authorities. Piaggio selected ANSYS SCADE as the development environment for the VCMS, since SCADE automatically generates source code from the model and minimizes the effort required to verify that the source code corresponds to the system model. The ANSYS SCADE KCG code generator is qualified as a DO-178B development tool, so conformance of the code to the input model is trusted, eliminating the need for verification activities related to the coding phase. SCADE's model-based methodology enables system engineers to model each function autonomously and check its performance on a host computer before the real hardware is available.

CREATING MODELS

SCADE models were created based on functional requirements from scratch by systems engineers for the textual documents, and automatically via the Simulink Gateway for the existing Simulink models. The SCADE Requirements Management Gateway was used to link the requirements to the embedded system design in the SCADE model. Engineers employed the SCADE Semantic Checker to verify the semantics of the model. Problems were identified and resolved in the host on the PC environment rather than in the much more expensive and complicated target hardware environment. A small portion of the code, primarily low-level layers such as input/output, was developed in C using traditional methods.

To ensure that the Simulink model was correctly translated to the SCADE environment, Simulink test vectors were translated into the SCADE environment. The test cases were translated to SCADE Input Scenarios. Then the test vectors were run in both Simulink and SCADE, and the results were compared to ensure that the translated SCADE model had the same functional behavior as the original Simulink model.

SCADE USAGE FOR UNMANNED AIRCRAFT VEHICLES
esterel-technologies.com/83piaggio


DOORS Environment
Sub System Operational Requirement & Simulink Models
Links from Test cases to Operational Reqs
SCADE RM GATEWAY
Test Cases
VCMS Funct. FRD Spec. Model (SCADE)
SCADE Input Scenario
SCADE Semantic Checker
QTE
Test Results

SCADE model validation process

SOFTWARE VERIFICATION

DO-178B verification requires proof that the functional tests performed by the test vectors fully cover model functionality. The SCADE Model Test Coverage (MTC) tool checked the model coverage and identified several areas that were lacking. Additional tests were designed and performed to provide the needed coverage.

Verification activities exponentially increase as the number of inputs of each model grows and as the number of models increases. In the early stages of the project, test vector generation, validation and configuration were issues. SCADE LifeCycle Qualified Test Environment (QTE) provided a solution by automatically running the tests in the host environment, comparing the results to the expected values and highlighting any errors.

Similar activities were performed on the target computer, sending test vectors into the executable code generated from the models. Piaggio engineers wrote a simple test application tool that runs on the target and plays a role similar to QTE by running the application with the SCADE input scenario then comparing results with the output generated by the same application and input on the host.

SYSTEM INTEGRATION

Models to handle different functional aspects of the VCMS were progressively integrated on the host computer to build a virtual VCMS to check interoperability of the applications well in advance of system integration. These verification activities were used to identify and solve many integration problems even before performing system integration on real hardware. As a result, the problems found during system integration were small in number, and all were mainly due to hardware/software/subsystem integration issues rather than to design errors. Once system integration was completed and the final tests executed, data from the real world were fed into the test vectors to further verify the model.

The entire project was completed in about 18 months, starting with model development performed directly by the system engineers and proceeding to compilation, integration and verification of the approximately 125,000 lines of source code that comprise the VCMS. The working team — in terms of equivalent full-time manpower — was limited to less than 20 engineers (system and software) who worked in tight coordination from the early design stages up to final system integration to meet the challenging target. As a result, the VMCS was developed and verified in an estimated one-third the time that would have been required had the code been handwritten.

The first flight of the aircraft was successfully completed in November 2013, less than two years after the project began. The VCMS worked perfectly. The P.1HH configuration will grow through incremental software releases that will add new functionalities to expand mission capabilities of the P.1HH.