AVIONICS magazine

Sunday, August 1, 2010

Product Focus: Software

Avionics software programmers challenged by integration, certification, testing issues for software-centric aircraft

By Barry Rosenberg

Airplane avionics systems are wholly dependent on software, but rarely does anyone but a software engineer wonder whether the programs that keep the plane aloft are anything but perfect in their operation.

But as software programs become more complex, the testing and certification processes that ensure the viability of those systems must be equal to the task.

“On the one hand, we have all these exciting new technologies, and on the other hand, we have concern about how to meld those technologies with the requirements of certification while retaining confidence that our increasingly complex software will keep us up in the air,” said Robert Dewar, co-founder, president and CEO of AdaCore, based in New York City. “It is always interesting that on the one hand people tend to think of computers as unreliable and full of bugs. Yet people seem quite willing to step on a plane, where you’re entrusting your life to very complex software.

“That is an interesting dichotomy, so one of the things I like to push is for more companies in avionics to think about how to write reliable software,” he added.

Dewar considers it remarkable that there’s never been a loss a life due to a software bug on an airliner, though there have been some frighteningly close calls, such as an incident involving Malaysia Airlines in 2005. In that instance, FAA eventually determined that a software error permitted the air data inertial reference unit aboard a Malaysia Air 777 flying between Perth, Australia, and Kuala Lumpur, Malaysia, to accept data from a failed accelerometer, causing the aircraft to make uncommanded maneuvers. The pilots regained control of the aircraft, and FAA eventually issued an emergency airworthiness directive for all 777 operators to install upgraded software.

Object Oriented

The use of object-oriented programming techniques is playing an increasingly important role in software development and testing of complex avionics systems, software experts say. In object-oriented programming, the data structure includes not only the data but also the function. Together, the two become an “object.” Such objects help in the modification of software because new objects can “inherit” characteristics of older objects.

Until recently, software certification was always test oriented. A company would prepare a large set of tests, and then conduct them one by one. Standards for testing would ensure that the tests are complete and traced to requirements.

It’s different in the object-oriented world, where the use of mathematically based techniques, or formal methods, is considered important in software development and verification.

“In the case of object-oriented techniques, it looks to us like you are going to have to use formal methods, to some extent,” Dewar said. “DO-178C doesn’t mandate formal methods, but it certainly leans in that direction,” he added, referring to the modification of RTCA DO-178B, which guides avionics software development, to address the use of object-oriented software.

“Basically, the technical issue is the following: when you use object-oriented programming and inheritance you are supposed to make sure that the inheritance makes sense. If you have a cow derived from an animal then the cow is still supposed to look like an animal, not something completely different. And it is hard to verify those required properties purely by testing. It can be done. It’s not impossible, but difficult. So we think formal methods are going to play a more important part.”

And then there are certain elements that can never be mathematically proved. For example, it’s an important requirement that cockpit displays be non-confusing. But how do you create a mathematical formula that tests “non-confusing?” That doesn’t minimize its importance, though, requiring some type of human input into the equation.

Another way of looking at object-oriented programming: it is all about not knowing at the time the program is written what will happen at run time.That concept, however, is diametrically opposed to the certification process, which is all about knowing exactly what will happen during a run.

“Those two [issues] are not happy friends,” said Dewar. “Yet the object-oriented techniques are so genuinely powerful that we want to be able to use them. Furthermore, the current generation of programmers doesn’t seem to be able to think any other way, partly because they’ve been trained in Java and C++. The current generation of college students, in particular, tends to be Java-only folks.”

There’s no denying the critical role that Java programming language can play in avionics systems, as proven by safety critical systems using Java that have been developed by companies such as Aonix, of San Diego, which competes against AdaCore. Dewar acknowledges that.

“Java is a very dynamic language.... Don’t worry about allocation of storage, just let the garbage collector do it. Everything is an object, so the Runtime system will figure out what is going on. It can load methods dynamically and pieces of libraries dynamically. So as it goes along, it figures out what pieces of software are needed and loads them. This is all very well, but all the things I mention don’t sound happy in a certified environment. And so, that is a real challenge.”

COTS Challenge

The use of commercial-off-the-shelf (COTS) software also provides a challenge for the avionics industry.

“One thing that continues to challenge us is the integration and certification of COTS software,” said Tim Budden, president of Esterline Control Systems AVISTA, based in Bellevue, Wash. “And as our software applications continue to grow so do things like partitioned operating systems, where an application will sit on top of Wind River’s partitioned operating system, for example. The cost to get that piece of software certified is huge, even when you’re taking an existing set of source code and creating a whole certification package for it.”

For Boeing’s 787 Dreamliner, for example, GE Aviation developed the partitioned hardware/software platform, with a variety of different suppliers like Esterline AVISTA tasked with developing the software that resides on a card.

That’s changed the dynamics a bit in how industry has or has not worked together in the past. Normally, companies would only have to worry about their own system, with integration being the specification that lets one box talk to another.

“Now you have to take your software and go a level deeper to make it work on somebody else’s hardware,” said Budden. “You actually have to have engineering interaction with other teams in other companies. I think that was an interesting dynamic on the 787.

“I wouldn’t say that it is necessarily a new problem,” he added. “But with the way the size of software grows, it continues to be a big cost. With portioned RTOSs (real-time operating systems) and others it grows with the system in terms of size and complexity.”

Object-oriented programming and COTS software provide opportunities as well as challenges. The same is true of model-based software development.

“Model-based development has grown over the past few years and what companies are finding, though, is that it has good application in some areas, and in other areas it is not as beneficial to them,” said Budden. “It works well in systems that have control algorithms and control laws like flight controls that lend themselves better to the type of system that the tool sets were originally developed, for example. Some companies are trying to take that further and expand it to other systems like displays that aren’t as control law based, and they’ve had mixed success with that.”

A couple of the more interesting software opportunities for the industry are ones it is being forced to make, either through government intervention or industry adoption of a new piece of hardware, such as electronic flight bags.

“NextGen and SESAR are going to basically provide a lot of different applications that are going to have to be written to control the interaction and communication between aircraft, and provide the new functionality that will be required to work in the sky,” said Gary Gilliland, senior manager of business development for safety products with LynuxWorks, of San Jose, Calif. “There is a plethora of different applications that will have to be written, such as security software that will validate messages from other aircraft.”

As always, the challenge is the cost of installing these new systems in aircraft and getting them certified.

“You have to decide whether you’ll write these applications and put them on new hardware, and find a place in the aircraft that can take this new hardware,” said Gilliland. “That’s the straightforward old-timey way to do it. The second challenge is to take partition-capable RTOSs and put this software in a different partition. Depending on what operating system you use, you can reduce your certification costs by using an existing system, assuming you have the bandwidth to add these applications. Or you could add another board to the chassis if you have an open slot.”

EFBs are not mandated for aircraft. In fact, many airlines have yet to make a business case for their installation. That might be, though, because they are focusing too much on the hardware instead of the software. It is software that permits own-ship position on an EFB, for example, that supports the business case. Without the software, the hardware is like a brick in the cockpit.

“When you add EFBs in the cockpit all you’re doing is adding weight,” said Mark McCausland, president of Ultramain Systems, of Albuquerque, N.M., which manufactures integrated maintenance and logistics software for airlines. “A lot of airlines are focusing on the hardware. Putting hardware in that cockpit doesn’t do you a bit of good.

“What you need to do is focus on the software first. That is where you get the gain. Focus on that first, and then circle back to the best hardware. A lot of people are doing it the other way around.”

Hi-Lite Project Launch

Industry partners AdaCore, Altran Praxis, CEA LIST, EADS Astrium Space Transportation, INRIA ProVal and Thales Communications in May officially launched the Hi-Lite project, an effort supported by French government agencies. The project is designed to increase the use of formal methods in developing high-integrity software, particularly to meet the latest DO-178C avionics software standard.

The $5.3 million, three-year project aims to create formal verification tools for the Ada and C programming languages to reduce the need for physical testing of high-integrity software systems.

“As high-integrity systems get larger and more complex, formal methods provide a cost-effective solution that decreases the need for testing and speeds up project completion,” said Arnaud Charlet, AdaCore’s Hi-Lite project leader.

“We aim to make formal verification faster and easier to use across large, multi-language projects that need to meet certification criteria, such as the forthcoming DO-178C standard,” he said.

Companies

AdaCore www.adacore.com

AgiLynx www.agilynx.com

AIM GmbH www.aim-online.com

Aircraft Management Technologies www.flightman.com

Aonix www.aonix.com

Astronautics Corporation of America www.astronautics.com

Avionyx, Inc. www.avionyx.com

Boeing www.boeing.com

Cobham www.cobham.com

DAC International www.dacint.com

Data Device Corp. www.ddc-web.com

EMS Aviation www.emsaviation.com

ENEA www.enea.com

ENSCO, Inc. www.ensco.com

Esterline www.esterline.com

Excalibur Systems, Inc. www.mil-1553.com

Freescale Semiconductor www.freescale.com

Gables Engineering www.gableseng.com

Gallium Visual Systems Inc. www.gallium.com

GE Intelligent Platforms, Inc. www.ge-ip.com

General Dynamics www.gdcanada.com

Green Hills Software www.ghs.com

IMS Flight Deck www.imsconsultants.com

Intel www.intel.com

Jeppesen www.jeppesen.com

Kongsberg Gruppen www.kongsberg.com

Kontron www.kontron.com

Lufthansa Systems www.lhsystems.com

LynuxWorks www.lynuxworks.com

Mercury Computer Systems www.mc.com

Objective Interface Systems, Inc. www.ois.com

Presagis Inc. www.presagis.com

Quantum3D, Inc. www.quantum3d.com

Real-Time Innovations www.rti.com

RMS Technology, Inc. www.rmstek.com

Sagem Avionics www.sagemavionics.com

SYSGO AG www.sysgo.com

TechSAT GmbH www.techsat.com

Teledyne Controls www.teledyne-controls.com

TTTech www.tttech.com

Ultramain Systems Inc. www.ultramain.com

Vector Software www.vectorcast.com

Wind River www.windriver.com

WSI www.wsi.com

Print This Window Close Window