

RELIEVING PRESSURE FOR UAV SOFTWARE DEVELOPMENT

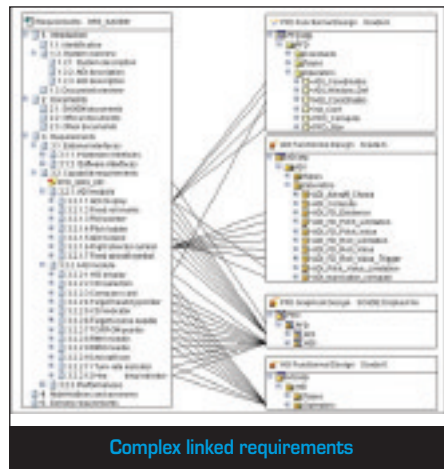
The demanding standards expected in UAS/UAV software development can be alleviated using FMBD and SCADE, as Jim McElroy reports

The explosive global growth in the requirements and development of Unmanned Air Systems (UAS) and Unmanned Air Vehicles (UAVs) represents both a significant business opportunity and significant risk for those supplier companies willing to engage and address this burgeoning market. There are political and technical challenges ahead that increase both the process complexity of developing such systems and the complexity of the applications themselves. We will focus our energy in this paper on some of the key issues software developers of UAS systems face and how Formal Model-based Development (FMBD) with SCADE can alleviate development pressure under stringent standards such as DO-178B Level A and mitigate risk while making the software developer more productive. FMBD with SCADE has been proven enormously successful on UAVs such as the Neuron (Dassault Aviation), Barracuda (EADS) and the Watchkeeper (U-TacS).

UASs – The Application Domain

UASs are being created for many different types of applications, both civilian and military. Common objectives for these systems include Intelligence, Surveillance, Reconnaissance (ISR), search and rescue, environmental monitoring, transportation, and payload deployment. To achieve these system level objectives, UASs and UAVs are built from numerous potentially complex subsystems which all may need to work cooperatively to achieve an overall mission task. In essence they are a system of systems. Examples of subsystems include, but are not limited to, flight controls which may offer capabilities such as autonomous take off, sense and avoid, mission flight, inertial navigation, global positioning, fuel control and management, power management, mission execution, and landing. If the air vehicle is a weapons deployment platform, then stores and weapons management are added to the basic platform of capabilities, in addition to any of the other sensor and control applications

required to deploy the weapon or payload. These are all examples of applications that “may” primarily reside in the air vehicle itself. On the ground there may be control station software and display software which potentially interacts with people as well as monitors air vehicles, satellites, and other ground control stations. As the system-level requirements expand for UASs so does the complexity of the application software and the process of developing it under safety-critical standards such as DO-178B Level A.



In addition to application complexity there are number of process related issues that developers must deal with. To name a few, at a high-level, communication is often a problem. Communication from the customer to the systems engineering team, from the systems engineers to the software engineers, and from software to other parties in the hardware and validation teams. In today's global economy, communication among the potentially disparate development organisations is also a major issue. Often, requirements come in the form of the written word which is subject to interpretation and can be quite ambiguous. Formal models solve this problem by eliminating ambiguity and increasing communication by raising the level of abstraction and increasing visual understanding. Formal model-based

development offers executable specifications and visual simulation, eliminating any ambiguity and increasing design visibility and intent.

Standards-based development is by definition, restrictive in nature. Under DO-178B up to Level A, there are a number of objectives that must be achieved to be compliant under the standard. For FMBD, there is a comprehensive handbook available on the Esterel Technologies web site entitled “Efficient Development of Safe Avionics Software with DO-178B Objectives Using SCADE Suite” located at: www.estereltechnologies.com/technology/handbooks/do-178b-for-avionics.html. It is simply not possible to give this topic justice in this short paper therefore I refer the reader to the above mentioned handbook to understand the process and value SCADE FMBD brings under DO-178B Level A.

Formal Model-Based Development

First, we have to start by defining Formal Model-Based Development (FMBD) with SCADE, herein after referred to only as FMBD, which consists of a rigorous unambiguous graphical semantic which combines data and control flow as well as a formal action language used to represent the architectural and behavioural aspects of a software-centric system. Granted, it's not enough to simply be able to represent system specifications, architectures, and designs graphically, but the formal semantics of the language must be rich enough to enable automation such as design verification, requirements traceability and impact analysis, model-test-coverage, documentation generation, and qualified or certified code generation. For UAS software developers, these key enabling technologies, when combined with good modular design and open-architecture development practices, ensures correct-by-construction, high quality, on-time deployment of more maintainable, more reusable, UAS components.

continues on page 32

From a process perspective, in addition to the pure technical advantages offered by FMDB, this approach facilitates communication and collaboration among the constituent development team members. With FMDB, designs are easily shared, verified at the graphical level, simulated, and documented. No longer do teams have to merely rely on the written word as a form of communication. Now, unambiguous models describe the exact intent of the System Under Design (SUD). Potentially the most important benefit to FMDB is the ability to easily and rapidly determine the impact of requirements changes, throughout the process, and respond to those changes in a formal, deterministic, model-based development manner where results are easily observed, reported, and analysed.

With FMDB design flaws are caught much earlier in the development cycle with model-based verification technology, when they are less costly to fix

In this environment where UASs may include a collection of unmanned aircraft, control stations, data collection stations, communication with other assets, and collaboration between different companies and agencies, there is a clear need for better forms of communication and better methods for system development, testing, and deployment. Formal models enable a more agile development process where quick, adaptive design changes can be easily constructed, analysed, verified, and then deployed on potentially new target architectures with safety and security built-in.

It's important to differentiate formal model-based development from informal methodologies and tools. Although a paper can be written on this subject alone, I'll summarise it this way. There is no question that some informal tools and methodologies are extremely flexible, powerful, and applicable to a wide range of design exploration. Often these informal methodologies enable experimentation of ideas and concepts which is truly valuable. However, that said, when push comes to shove, the best path to safety-critical application development and certification is a straight no nonsense correct-by-construction design flow. With informal methodologies, often the developer is free to put whatever they want into the design, as long as it makes sense to them. With this informal process 'garbage in' often leads to 'garbage out'. With this approach, the choice is to go through lengthy, costly testing, debug, and model-refinement cycles. Formal model-

based development is extremely efficient in that it eliminates this costly process through correct-by-construction design, verification, and automation technologies such as qualified code generation.

Because formal graphical models are more easily understood and requirements can be linked into the graphical model, the ability to effectively respond to requirements changes improves dramatically. Because of this tight coupling between the requirements and the design, maintenance of the system is dramatically improved as design modifications are made at the graphical level where they can be executed and verified prior to deployment.

Testing is obviously a primary concern for developers using traditional manual software development methods as this is potentially

the most time-consuming and costly activity in the development process. FMDB eliminates low-level unit testing for all code automatically generated through qualified code generation, dramatically shortening the verification of the software and the certification of the complete system. FMDB integration with tool-chain testing tools facilitates the testing of the complete application either on the host or the target hardware. This is extremely valuable in that often the software team must continue to work even when target hardware may not be yet available.

With FMDB, documentation is a by-product of the design process. Automatically generated and always up to date, the design documentation with FMDB is customisable and easily generated. Furthermore, because SCADE FMDB has been developed in close cooperation with the certification authorities of EASA, Transport Canada, and the FAA, certification evidence is provided, eliminating the need for the user to create this material on their own. Again, this shortens the time-to-certification.

FMDB Summary Workflow

A. Requirements

Often the requirements for UASs and UAVs are captured using either informal tools such as Word, Excel, PowerPoint, or specialised COTS tools such as DOORS or RequisitePro from IBM. Regardless of the authoring tool, FMDB enables the linking of requirements to the design models and throughout the development process. Most importantly, this makes the development

team more productive and nimble as this technology can be used to easily determine impact of requirements changes throughout the design and the product life-cycle. This gateway between the requirements tool and the formal modelling environment ensures consistency, completeness, and traceability of requirements throughout the development process. An example of how requirements can be linked to model elements is included in the diagram.

B. Design, Verification, and Implementation

As the title of this section suggests, with FMDB the implementation is actually a by-product of the design and verification process. With this approach, the process of waiting until the end to test the application is eliminated. With FMDB design flaws are caught much earlier in the development cycle with model-based verification technology, when they are less costly to fix. Furthermore, because of the formal nature of the models and the qualified code generation, further analysis of the model can predict worst case execution time (WCET) and stack usage of the application itself on the target hardware.

C. Design for Reusability

One of the fundamental benefits of FMDB is how easy it is to think in terms of components and modules when constructing and verifying the system at the graphical level. Ultimately, with this model-based approach models of software components are naturally designed and verified to address the requirements. These components are easily traced to requirements and are well-defined in terms of their interfaces and behaviour. This subject of modularity and design for re-use is another paper topic on its own, however this proven FMDB methodology ensures that model-based components, with their associated requirements and documentation, are more easily created and maintained in formal models than traditional manual coding and development methodologies. The modularity provided by FMDB increases UAS platform agility, platform extensibility, and component reuse in multiple aircraft and ground systems, while at the same time reducing cost of deployment and maintenance of the software.

UAS Software – The Path Ahead

As identified in the "United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047", the goal of the USAF is to develop UASs that are 'increasingly automated, modular, and sustainable'. In addition, the expansion of capabilities and

theatres of operation for UASs means that there is a requirement for the UASs to operate in environments with both commercial and military aircraft, where in the U.S.A. that means to also operate within the National Air Space (NAS) and under the authority of the Federal Aviation Administration (FAA). Here the FAA insists that UAS operations have an equivalent level of safety to those of manned aircraft. Therefore, sense and avoid is a basic requirement. In order to achieve this goal, UASs must become more automated and "smarter", being able to provide intrinsic capabilities such as "sense and avoid". There is significant debate as to the levels of autonomy that could and should be granted to UASs but the immediate future shows that UASs will be capable of taking off, executing pre-programmed flight plans, and landing without any human interaction. Clearly, there are significant political, technical, and in some cases moral reasons for keeping humans in the flight decision loop however UAVs will no doubt become increasingly self-sufficient within their theatres of operation.

The modularity provided by FMBD increases UAS platform agility, platform extensibility, and component reuse in multiple aircraft and ground systems

To achieve these goals the USAF requires an open architecture with clearly defined, non-proprietary interfaces, and developed under enforceable standards. Both civilian and military UAVs must receive certificates of authorisation from the FAA to fly in the National Air Space (NAS) and to achieve those certificates, systems must provide key safety capabilities such as "sense and avoid". Looking ahead, the USAF states "Future UAS should be multi-mission, all-weather, net-centric, modular, open architecture and employ leveraging appropriate levels of autonomy." It's clear that formal modular development methodologies will be required to meet these demands in a timely and efficient manner.

Summary

Formal Model-based Development with SCADE addresses a number of key issues developers face when constructing safe, reliable, and sustainable UASs and UAVs. Communication among each of the interested parties and development organisations is a key capability that is improved dramatically with unambiguous formal graphical models. In addition, formal models present a clear and concise view of how the requirements link to the design. More importantly, with FMBD, the impact of requirements changes is easily determined and visualised. As a result, the development team can respond quickly and confidently to those changes through graphical design and executable verification. Through FMBD, developers can leverage intrinsic component-based development through modular design strategies. This modular approach enhances design reusability and platform extensibility. Certainly, a major benefit of FMBD is the enablement of design verification, qualified code generation, and documentation generation. Each of these technologies enabled by FMBD produces applications which are correct-by-construction, eliminating design flaws early, producing code which is safe, reliable, and documentation which is always accurate. As in other safety-critical application areas such as rail transportation and medical instrumentation, formal model-based development is a key element to solving UAS/UAV software development challenges.

.....
The author is Esterel Technology's vice-president of corporate marketing and business development

“Let Sidney
find it
for you...”



SITEFIND
The industry specific search engine

www.sitefindonthenet.net