

# SECURITY CHALLENGES IN UAV DEVELOPMENT

*Chris Constantinides, Principal Security Systems Technologist*

*Paul Parkinson, Senior Systems Architect*

*Wind River, Alameda, CA*

## Abstract

The expanded development and diverse mission operations of unmanned air vehicles (UAV) have exposed INFORMATION SECURITY (INFOSEC) and COMMUNICATION SECURITY (COMSEC) concerns that are not easily addressed in traditional federated or currently deployed integrated modular avionics (IMA) systems. The need to operate military UAVs in civil airspace communicating over unclassified links to foreign air traffic control systems and keep sensitive and/or classified information separated without increasing space, weight and power (SWaP) poses challenges to UAV systems architecture. In this paper the MILS (Multiple Independent Levels of Security) software architecture will be discussed in relation to how it can fulfill these disparate UAV system design requirements.

## The Advent of Unmanned Systems

In recent years, there has been very rapid growth in the development and deployment of unmanned air vehicles (UAVs). These unmanned systems are being used in a very diverse range of roles, from urban reconnaissance through to high altitude long endurance (HALE) operations. In many cases, program developments have been driven primarily by operational requirements to deploy systems in environments deemed to be too hazardous or too hostile for human operators. However, the development of these unmanned systems has not overcome the technical challenges to fulfill these operational requirements along with producing additional, tangible benefits.

UAVs do not need to be encumbered by life-support systems for a human operator, often

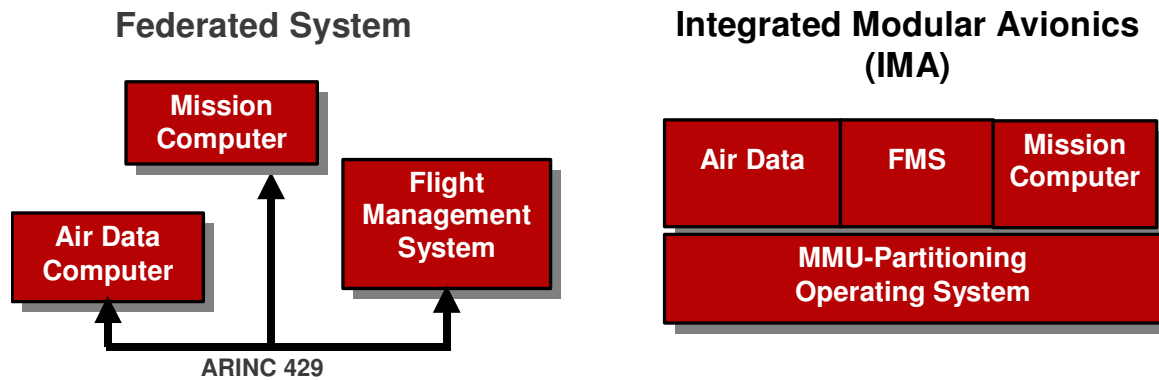
enabling the design to be physically smaller and lighter than a manned vehicle. In the case of military UAVs, this can contribute towards a reduced radar cross-section (RCS) resulting in a lower probability of intercept (LPI) by hostile forces. This can also reduce the need to use stealth, supersonic or hypersonic capabilities currently used for high-altitude spy planes (like the U-2, SR-71 and successor aircraft).

UAVs are also capable of performing much longer, extended missions than those restricted by the limits of an individual, human operator. This is because they provide the ability to co-ordinate operation through a number of remote operators working in shifts. This capability provides military planners with the ability to have increased loiter time on target, which can be invaluable for a changing situation on the ground.

There is also the additional benefit that military UAVs can be used in theaters which present a higher-risk of intercept than would be acceptable for manned aircraft, including airborne reconnaissance to provide invaluable battlefield intelligence. They can even be used as active decoys, penetrating deep into enemy territory in offensive air operations [1]. Armed combat variants are now being deployed (usually referred to as unmanned combat air vehicles or UCAVs

## Modular Avionics in the Extreme

Aircraft systems are becoming more and more complex in order to implement more and more advanced functionality. As a result, the software content in these systems continues to grow at an astonishing rate. For example, in the 1980s, the



**Figure 1. Avionics architecture**

software content in the avionics systems of military fast jets was around one hundred thousand source lines of code (SLOC), and this has increased significantly in recent years. It has been estimated that the F-35 Lightning II will have around seven million SLOC.

The increasing complexity of these systems has required a corresponding increase in system performance and, therefore, physical footprint in terms of space, weight and power (SWaP). However, there are pressing requirements to reduce the physical footprint of systems within aircraft and, especially, within UAVs due to their physical size constraints. So a significant increase in processing density is essential.

This problem is being addressed through the adoption of integrated modular avionics (IMA). This architecture comprises common computing platforms which can host multiple applications concurrently (Figure 1). This has been proven to save space, power and weight. If an open standards-based software architecture is used instead of a closed proprietary software implementation - which can be the case even with some COTS implementations - it can provide additional benefits in terms of improved modularity, interoperability and software portability.

The ARINC 653 software architecture [2] is an example of an open standards-based approach and has become pre-eminent in recent years. It provides a specification for an application executive for integrated modular avionics systems and is based upon the principles of robust-temporal and spatial partitioning. These are fundamental requirements to enable multiple applications with differing levels of

safety-criticality to run concurrently on the same processor; without this a system with mixed criticality would need to have all of the software safety-certified to the of the most critical application, which would have a significant impact on certification efforts and costs.

ARINC 653 defines an application executive, or APEX, which provides an application programming interface (API) of 51 routines to enable the development of portable applications on an IMA platform, supporting temporal and spatial partitioning along with communication between applications in different partitions through well-defined ports. ARINC 653 also defines a health management framework, which can be used to provide a hierarchical framework for error detection and recovery. This framework provides an increased level of fault tolerance, which can be used to achieve improved operational availability.

The ARINC 653 standard has been refined in recent years through an iterative cycle of standardization and development experience gained through the safety-critical IMA programmes where it's been used. The efforts to produce guidance for safety certification of IMA systems under RTCA DO-178B [5] / EUROCAE ED-12B [6] has also involved an iterative cycle, with experience from real programmes and new requirements feeding back into the standardization process. This has led to a role-based development approach to facilitate the certification process being defined in the common guidance documents DO-297 [3] and ED-124 [4] - produced through the collaborative efforts of joint RTCA and EUROCAE working groups. This role-based development approach, involving platform provider, system integrator, and

application developers, can be supported in ARINC 653 implementations.

## Implications of Secret Mission Operation on Unmanned Systems

UAVs contain a number of systems which perform different functions. These include avionics systems, mission systems and sensor systems. These use firewalls, trusted communications and encryption for the secure transmission and reception of information between networks, without transforming the information during its transport. This is known as Communications Security (COMSEC). There is likely to be data communication between the avionics systems and mission systems which passes positioning information, bearing, altitude, etc, for example. However, information may also need to be transformed securely between applications, subsystems or networks. This is known as Information Security (INFOSEC), and may be required in addition to ComSec.

Let's consider a hypothetical scenario involving a UAV which is tasked with a Top Secret reconnaissance mission (Figure 2). The UAV mission systems may contain Secret or Top-Secret data, including the mission flight plan and there is a risk that it could disclose classified Secret or Top Secret information to the Unclassified, civil air traffic control (ATC) system. This is a very real concern because the communications security (COMSEC) implementation is not concerned with the type of data which is exchanged between networks, only whether the networks and applications can actually communicate.

The UAV could take off within controlled airspace and have to interact with civil ATC over unencrypted links, operating 'in the black'. The UAV remote pilot could then request a vector out of controlled space from ATC in order to perform its reconnaissance mission and, after this point, all flight information will be Top Secret or 'in the red', including all communication which will be over links with strong encryption.

Upon completion of the reconnaissance mission, the UAV could re-enter controlled airspace and interact with civil ATC again. The transition back from red to black is particularly challenging

because this will now involve systems containing Top Secret data communicating over unencrypted links to an unclassified ATC. The flight plan and logged data for the red part of the mission must be retained for analysis/debrief, but must not "leak out" during return to base.

The UAV mission systems are likely to utilize IMA platforms due to the constraints of space, weight, power and heat, as outlined earlier. In this environment, it is likely that it will not be practical to have separate systems to provide physical isolation between black and red data. This threat must instead be addressed through an InfoSec implementation which is Multi-Level Secure (MLS).

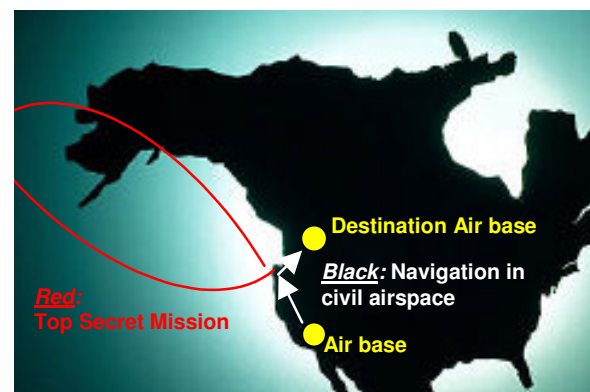
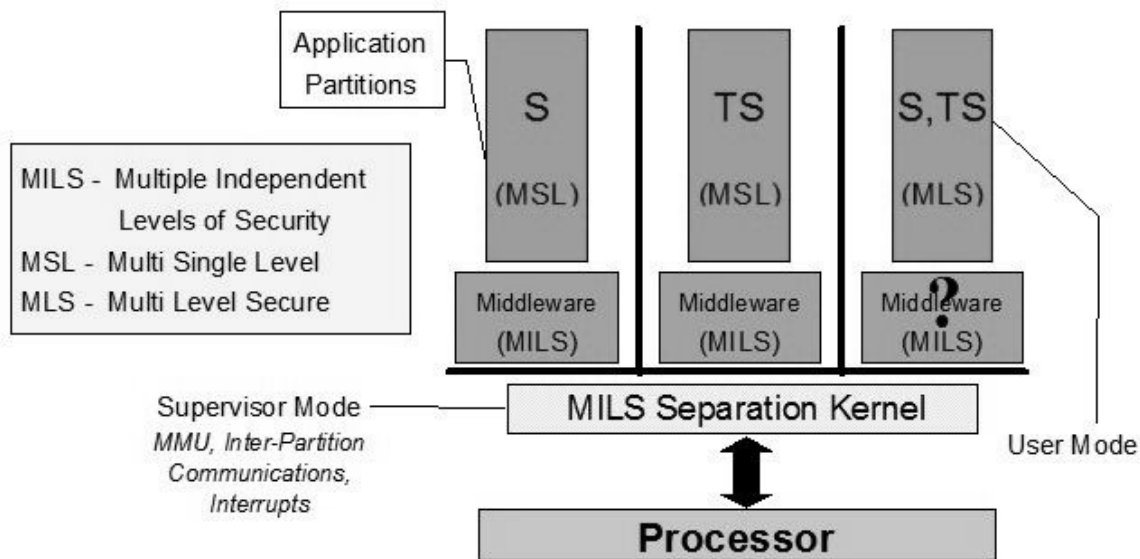


Figure 2. UAV mission

## Multiple Independent Levels of Security

In recent years, the Common Criteria (ISO-15048) [7] has become widely accepted, following the pioneering collaborative efforts in information security by the United States, Canada, United Kingdom, France, Germany, and the Netherlands in the 1990's. This standard defines the criteria and assurances required for different types of systems and threats, including Multi-Level Secure systems holding information at multiple levels of security classifications. In the scenario described earlier, when the UAV is returning from its reconnaissance mission, it will have retained its Top Secret flight plan and mission log. It will be communicating over unencrypted links to an unclassified ATC, resulting in Unclassified and Top Secret data being present on the same system with a high Evaluation Assurance Level requirement. The Multiple



**Figure 3. MILS architecture**

Independent Levels of Security (MILS) architecture has evolved to provide a solution to overcome the prohibitive security certification costs and effort associated with previous, monolithic implementations. It does this by dramatically reducing the amount of security-critical code and dramatically increasing the scrutiny of that code to the appropriate security level. The most current accepted MILS architecture [8] (Figure 3) comprises of a MILS separation kernel which is able to host multiple applications of different security classifications and/or safety levels in separate partitions.

In the MILS architecture, the separation kernel is responsible for enforcing the system security policies. The system security policies define both the space partitioning of the individual applications along with the information flow between these individual applications. The permitted data flows are defined in a security policy database which is used by separation kernel's reference monitor to enforce these policies and detect, and if need be, limit the damage by attempted violations.

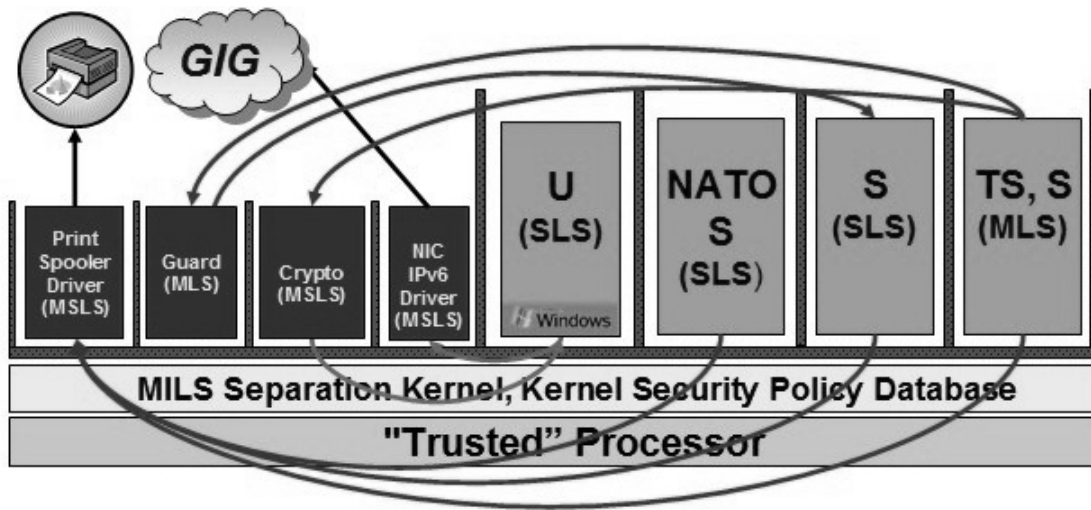
In a high assurance system implemented using the MILS architecture; the design can significantly reduce the possibility that no implicit communication could occur through covert channels, which can occur through variations in timing and resource availability. A correctly designed high assurance system that enforces

temporal and spatial partitioning also aids in the prevention of threats like covert channel communications. Additional techniques should also be used to prevent covert communication channels through utilization of the system hardware components. These capabilities would ensure that the UAV in our example keeps Top Secret mission data isolated securely from the unclassified application communicating with the civil ATC.

### MILS Implementation Architecture

In order to understand the principles and implementation of a MILS system, it is key to understand the different components and how the components are layered in order to create assurance. In the MILS architecture described earlier, the separation kernel (based on the Separation Kernel Protection Profile [9]) enforces the security policies defined for the system.

The separation kernel itself is built on four fundamental security policies of: Information Flow, Data Isolation, Periods Processing and Damage Limitation. These four policies create an architecture that allows creation of additional components that are Non-Bypassable, Evaluatable, Always Invoked and Tamper Proof. (NEAT). These components form the basis for local application specific Reference Monitors which can be created. Thus a system can be architected to



**Figure4. MILS System**

create layers of assurances that use the facilities and integrity of the separation kernel to create application specific security policies (aka Application Reference Monitors).

The MILS architecture provides the ability to create highly robust systems, when used in conjunction with hardware and proper security architecture system design [10] (Figure 4). The processor and other system hardware components must not interfere with the operations of the system or violate any of the security policies which are being enforced by the separation kernel. Proper system design and analysis must be undertaken to understand the hardware components functionality and interaction with the system. Using a system without understanding the hardware can lead to false expectation of the security assurance of the system.

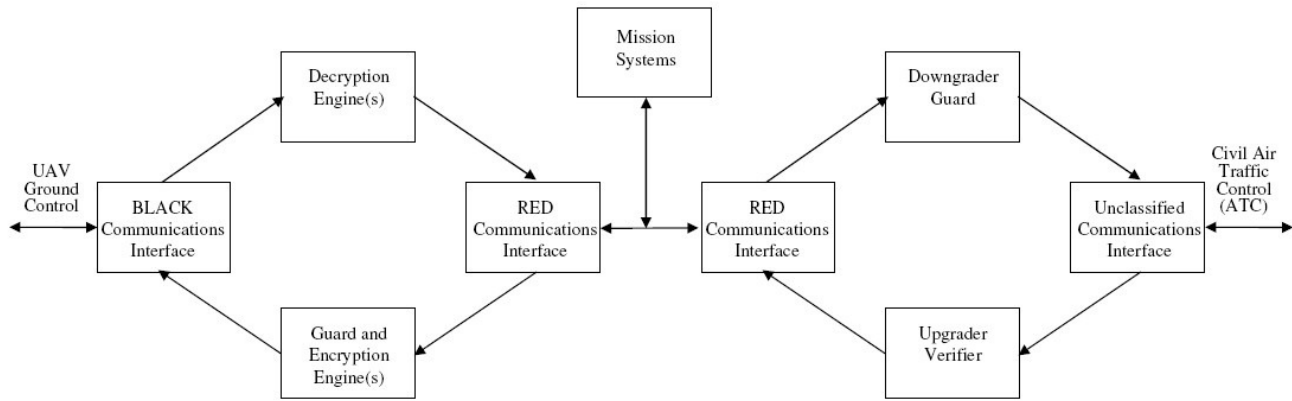
In high assurance systems, the initial trusted entity provides the foundation for additional layers of trust created by the system. This initial and continued trust entity is provided by the hardware. If the hardware is not utilized properly, this can lead to a security breach and covert channels. The different system hardware components, including caches, Direct Memory Access (DMA) and memory addressing components, etc., must be properly utilized by the both the hardware and software architecture. Failure to do so will compromise the trust and assurance of the system.

For example, if a DMA operation is started by a low assurance or untrusted partition and the

memory address range accessible to the DMA operation is unbounded, the operation could move information to or from another partition's memory space to which it should not have access. Even if the DMA operation does not violate memory space partitioning, if the DMA operation does not complete its operation in the time defined by the system security policy, this action could have an impact on the assurance of the systems. Vital expected operation of the system may not operate as defined by the system security policies. The effect on the system could reduce or eliminate execution of critical components like Security Audit Logging and Monitoring which can leave a system open to a variety of threats and compromises.

In a MILS system, the objective is to utilize the partitioning architecture to create layers of assurance. This means that the system should be architected to reduce the amount of security and critical functionality that need to be scrutinized. This is accomplished by putting unique functionality into different partitions, and by defining the informational flows and execution to these functional partitions so that code can operate exactly as defined in the system security policy database.

In integrated modular avionics (IMA) systems based on the ARINC 653 software architecture, device drivers and Operating System services reside and execute in supervisory mode. This approach is appropriate for systems which will undergo safety-certification, even up to DO-178B Level A, but for



**Figure 51. UAV Security Solution**

systems which also need to undergo security certification there are additional considerations.

In a high assurance MILS architecture, it is almost a necessity that all drivers and Operating Systems services be moved to a non-supervisory execution space (Figure 4). This bounds the impact of these devices on the system as described by the system security policies. All communications to and from the partitions and even execution are statically designed into the system with rigorous spatial and timing boundaries.

The cascading impacts of different applications' execution and interdependencies are well defined in high assurance MILS system. Hardware and system operation must be closely scrutinized on its impact to the system and violation of system security policies enforced by the separation kernel. System hardware functionality must be understood and controlled in a secure fashion. What this means is that functionality, such as device drivers or systems services, needed by low assurance partitions that could impact the operations of the separation kernel or execution of high assurance partitions will need to go through another layer of assurance so that the low assurance function does not effect the system.

Taking our previous example of UAV reconnaissance mission which could take off within controlled airspace and have to interact with civil Air Traffic Control (ATC) over unencrypted links, operating 'in the black'. There is vital information that must be given to and acquired from the ATC, like position, which has different value of the data

depending on location in the mission. As information is transferred in controlled space airspace to the ATC for position for the initial request to leave controlled airspace could be considered unclassified, at that time. Later in the reconnaissance mission when the UAV position is classified as Top Secret, the release of its position to ATC would be a breach of security. The functional architecture in the UAV Security Solution (Figure 5) can greatly minimize the risk that the UAV position, or any other data, could leak out.

The key to the above security architecture is that it can be implemented in a MILS layered assurance architecture and could prevent the leakage of data. The UAV Security Solution shows how the architecture breaks down the functionality of the components providing separation and defining the functional informational data flow. The security functionality can be decomposed into small pieces which enables those pieces to be heavily scrutinized. The functional blocks where multi-level security data could exist, such as the Downgrader Guard and Upgrader Verifier functions block, could then be evaluated to the appropriate high assurance levels thus reducing the security evaluation costs. This requires careful design to ensure that each piece maintains trust and operates as intended by the system security policies.

One challenge of this MILS system is to enforce correct temporal execution of the system security policies. In the above example, proper design must be undertaken to ensure that the Unclassified

Communications Interface, cannot compromise the systems operation. Hardware resources like DMA and high-speed communications buses, which are used in many instances for transferring information, must be under the control of the proper security policy so that they cannot covertly influence the operation of the separation kernel enforcement of the system security policies. This means that only application partitions defined in the system security policy would have access to that resource, and that all other applications that would need to utilize the resource would need to have the proper authorization. The authorization would be minimally defined by the Partition Information Flow Policy in the system security policy database and the local security policies of the application identified to own the resource.

The use of the MILS architecture in creating high assurance systems is currently the best route to achieving affordable yet highly secure systems. The MILS architecture utilizes the concept of robust temporal and spatial partitioning of ARINC 653, and expands on the architecture by requiring stricter management of all system components that could be a threat to the system enforcement of its security policies. This approach shows that security can be achieved without sacrificing space, weight and power (SWaP) requirements.

## The Future

The use of unmanned systems, both UAVs and UCAVs, is continuing to increase rapidly and the knowledge and experience gained through their deployment is being used in ever-more challenging missions. In the near future, unmanned systems will be used for highly sensitive reconnaissance missions and offensive operations, which will place stringent security requirements on the classified data contained in the UAV systems. The MILS software architecture provides a viable route for the successful development and deployment of these systems.

## References

- [1] Tice, Capt. Brian P., USAF, "Unmanned Air Vehicles, The Force Multiplier of the 1990s", Air Power Journal, Spring 1991. URL <http://www.airpower.maxwell.af.mil/airchronicles/apj/apj91/spr91/4spr91.htm>
- [2] ARINC Specification 653-1, ARINC. URL [https://www.arinc.com/cf/store/catalog\\_detail.cfm?item\\_id=632](https://www.arinc.com/cf/store/catalog_detail.cfm?item_id=632)
- [3] "Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations", DO-297, RTCA. URL <http://www.rtca.org/onlinecart/product.cfm?id=382>
- [4] "Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations", ED-124, EUROCAE.
- [5] "Software Considerations in Airborne Systems and Equipment Certification", DO-178B, RTCA. URL <http://www.rtca.org/onlinecart/product.cfm?id=341>
- [6] "Software Considerations in Airborne Systems and Equipment Certification", ED-12B, EUROCAE.
- [7] ISO-15408, "Information Technology – Security Techniques – Evaluation Criteria for IT Security", URL <http://www.iso.ch>
- [8] Vanfleet, M., US National Security Agency, 25th July 2002, "MILS Architecture", Open Group Security Forum.
- [9] "U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness", URL [http://niap.bahialab.com/cc-scheme/pp/PP\\_SKPP\\_HR\\_V1.03.pdf](http://niap.bahialab.com/cc-scheme/pp/PP_SKPP_HR_V1.03.pdf)
- [10] "Wind\_River\_Presents\_MILS\_Aug2008.ppt" (slide 38), derived from Calloni, Dr. Ben A. "Information Assurance via OMG / TOG Standards: A Necessary Step for Affordable, Secure Cross Domain Interoperability", URL <http://www.omg.org/docs/omg/07-06-08.pdf>

## **Email Addresses**

Chris Constantinides is a Principal Security Systems Technologist for Wind River. He has been in or serviced the Military and Aerospace Industry for over 20 years. Chris is a contributor in the MILS community standards and a member of the Real Time Embedded Systems Forum in The Open Group. Prior to joining Wind River, Chris has worked for McDonnell Douglas Corporation and Boeing Corporation on numerous programs that span from fighter aircraft to commercial jets to missiles. Chris has also been a part of advanced research and development for both of these companies doing system and sub-system hardware and software design. Chris can be contacted via email at: [Chris.Constantinides@windriver.com](mailto:Chris.Constantinides@windriver.com)

Paul Parkinson is a Senior Systems Architect with Wind River, working with customers in the Aerospace & Defence sectors in the UK. Paul's professional interests include Integrated Modular Avionics (IMA) and Intelligence Surveillance Target Acquisition Reconnaissance (ISTAR) systems. Paul blogs on A&D industry issues on the Wind River website at URL <http://blogs.windriver.com/parkinson>. Paul can be contacted by email at: [Paul.Parkinson@windriver.com](mailto:Paul.Parkinson@windriver.com)

*27th Digital Avionics Systems Conference  
October 26-30, 2008*