

# ADDRESSING SYSTEM RECONFIGURATION AND INCREMENTAL INTEGRATION WITHIN IMA SYSTEMS

Francisco Ferrero <sup>(1)</sup>, Ana Isabel Rodríguez <sup>(2)</sup>

<sup>(1)</sup>GMV AD S.A., Isaac Newton, 11, Tres Cantos, Madrid, 28760, Spain, Email: fferrero@gmv.com

<sup>(2)</sup>GMV AD S.A., Isaac Newton, 11, Tres Cantos, Madrid, 28760, Spain, Email: airodriguez@gmv.com

## ABSTRACT

Recently space industry is paying special attention to Integrated Modular Avionics (IMA) systems due to the benefits that modular concepts could bring to the development of space applications, especially in terms of interoperability, flexibility and software reuse. Two important IMA goals to be highlighted are system reconfiguration, and incremental integration of new functionalities into a pre-existing system.

The purpose of this paper is to show how system reconfiguration is conducted based on Allied Standard Avionics Architecture Council (ASAAC) concepts for IMA Systems. Besides, it aims to provide a proposal for addressing the incremental integration concept supported by our experience gained during European Technology Acquisition Program (ETAP) TDPI.7 programme. All these topics will be discussed taking into account safety issues and showing the blueprint as an appropriate technique to support these concepts.

## 1. INTRODUCTION

Non-IMA avionics systems (e.g. federated systems) often include avionics units on which functional software is bound to the underlying hardware. It is common to communicate those units via several data buses through more than one communication standard. Some consequences are frequent and long maintenance operations, low availability, and high software-hardware coupling. As aircraft systems are becoming increasingly larger and more complex, current mission and operational requirements are also becoming more constraining. And market availability of components is getting so short that systems are often becoming obsolete during their development.

In ASAAC, the IMA Core System is viewed as a single entity that includes several hardware modules (known as processing resources) connected by a unified network through no more than one communication standard (see [1]). Each of those hardware modules can allocate several applications of different safety critical levels running (even concurrently) on the same operating system sharing resources. Hence, several processing

resources can be used to construct any avionics system regardless of size and complexity, communicating with sensors and actuators thanks to a unified network interface.

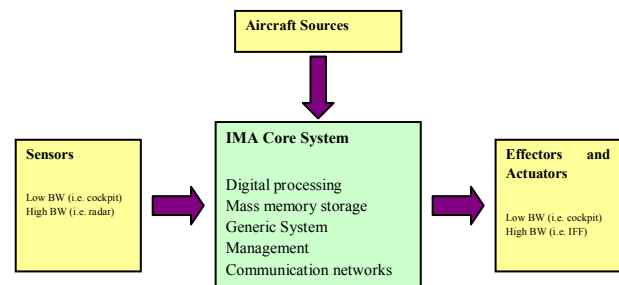


Figure 1. IMA System definition

In this paper, blueprints are shown as an appropriate mean to conduct system reconfiguration, incremental integration and system incremental upgrade. Section 2 provides a brief overview of the most important ASAAC concepts needed to understand system configuration and incremental integration. Section 3 describes how system reconfiguration is addressed by ASAAC standards from a technical point of view. Safety considerations will be also taken into account debriefing the most important issues regarding the production of safety evidences that will enable aircraft certification. Section 4 links topics discussed in previous section to the incremental system upgrade process, including also which safety matters should have in mind to permit the certification of the new upgraded system. Finally, in section 5, synergisms between ASAAC and system development in space domain are highlighted.

## 2. ASAAC FOUNDATIONS

This section provides a very brief description of the main concepts necessary to understand system reconfiguration and incremental system upgrade issues.

### 2.1. Three-layer stack

In the introductory section it was already depicted how the IMA system looks like. It is important to clearly state what an IMA System is according to ASAAC

standards. The IMA Core System and other external subsystems, like actuators and sensors, form part of the IMA System. The IMA Core System provides system functions and it is formed of the IMA Platform and the application software. The IMA Platform is composed of a set of Common Functional Modules (CFM, fully described in [3]) interconnected along the aircraft using a standardized communication protocol. CFMs could be considered like the computational nodes hosting applications processes.

IMA Platform and application software are not physically but logically isolated within the system. Logical interactions between applications and the IMA Platform are clearly identified in [8]. The IMA Platform interacts with applications and with input/output (I/O) devices (external subsystems, sensors, actuators, etc.). The communication of applications with I/O devices represents another logical intersection to take into account. Those logical intersections make IMA Systems difficult to analyze in terms of safety and integrity. It is necessary to establish some kind of mechanism to divide the problem into smaller pieces. The ASAAC three-layer stack sets up this mechanism.

Each CFM implements a standardized three-layer stack composed by the following layers:

- Module Software Layer (MSL): provides functions to the operating system to control and monitor the underlying hardware devices.
- Real-Time Operating System (RTOS), GSM functions and the Run-Time Blueprint (RTBP) represent the platform core services.
- Application and AM function: implement the IMA Core System functions.

Next figure depicts the interfaces and components of the three-layer stack described in [2].

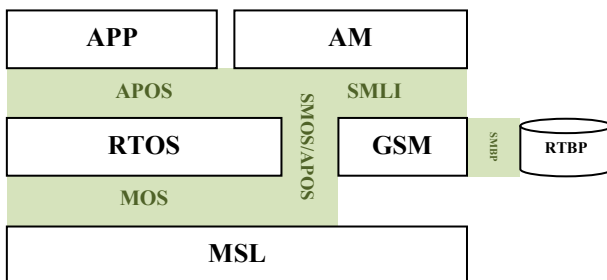


Figure 2. ASAAC three-layer stack

Application software interfaces with the RTOS using the Application to Operating System (APOS) interface, which makes applications independent from lower layers. The RTOS is independent from the underlying hardware thanks to the Module to Operating System (MOS) interface. The hardware provider supplies this layer to control and monitor hardware devices. Other interfaces, like System Management to Operating System (SMOS) and System Management to RTBP (SMBP) allow system management functions to control and monitoring module and access the information stored in the RTBP, respectively.

This separation into three layers allows separating safety arguments of the application software and the IMA Platform, as pointed in [8]. This separation would be possible if failure modes does not propagate to the application level, it means, the IMA Platform must be designed to safely support this separation, also necessary to support the arguments for incremental system upgrade and certification.

## 2.2. System management hierarchy

GSM hierarchy is responsible for implementing fault detection, health monitoring and configuration management services [4]. Five functions form part of the GSM hierarchy, and an additional function running at application level is introduced to manage application-triggered events:

- Fault Management (FM function): responsible for identifying, masking, confining and localizing faults.
- Health Monitoring (HM function): responsible for the health status of the system.
- Configuration Management (CM function): responsible for establishing the initial configuration of the system, reconfiguring when required after event reception, and performing power-up and power-down procedures.
- Security Manager (SM function): not related with reconfiguration, responsible for system security policies like authentication, encryption and decryption.
- Application Management (AM function): responsible for managing the functionality of the system at application level in terms of mode selection.

The impact of the system reconfiguration could be much localized depending on the nature of the event that may trigger the reconfiguration. This can be seen in [4], where the concept of the hierarchy of GSM is defined at different levels:

- Aircraft (AC) level: controls/oversees the entire IMA Core System.
- Integration Area (IA) level: controls/oversees a section of the IMA Core System in a similar way the aircraft level does and the modules assigned to that integration area. The resource elements assigned to a particular integration area provide functions that are logically grouped (e.g. navigation, mission, or radar).
- Resource Element (RE) level: represents the lower system management level and it is responsible for managing and controlling computational Processing Elements (PE) (i.e. a microprocessor, a digital signal processing chip, and field programmable gate array).

Figure 3 depicts an example of logical system architecture. AC level controls different IAs, each of them responsible for managing and controlling different aircraft functions, e.g. guidance and navigation, mission computer or air conditioning control. There are two different types of IA: system or functional. A system IA manages several lower IA. In the opposite, a functional IA manages one or more RE. System IAs can only manage other IAs, while functional IAs only manage REs. A RE is a logical concept, not a physical one, which is linked to a particular PE in a CFM. If a CFM has more than one PE, several REs can be mapped to the same physical module. In this logical system architecture, applications are mapped to each RE.

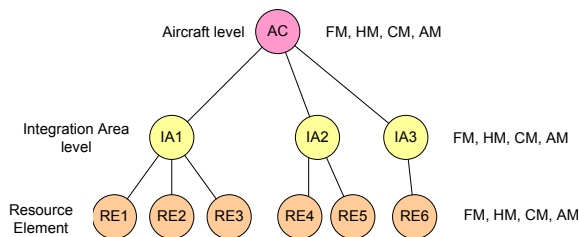


Figure 3. System management hierarchy

The IMA system will be reconfigured in response to a limited range of events triggered by FM and AM functions in presence of hardware, software or logical failures. This management hierarchy allows the fault

confining and avoids propagation to other modules within the IMA Core System.

### 2.3. Blueprints

When an event is received, GSM functions require enough information to find out the optimal configuration to be changed to. This optimal configuration could be pre-established at system design phase or on-line computed by optimization algorithms. This is the role of the blueprints within the IMA system, especially the RTBP. According to [6] and [7] the following blueprints form part of the blueprint model:

- Software blueprint: provides information about the resource consumption of the application in terms of computing power, memory (static and dynamic), and communications. It may also include some modelling parameters necessary to model application functional and non-functional behaviour.
- Hardware blueprint: describes the resource provision for each hardware module in terms of computing power, memory, and communications.
- Configuration blueprint: describes how application software is mapped onto hardware at physical level (describes where an application will be executed on) and logical level (describes the resource element an application is mapped to and the integration area this resource element is managed by).
- System blueprint and RTBP: describes the IMA Core System so that it can be loaded onto the relevant platform. It picks the relevant information from previous blueprints in order to provide an IMA Core System model where software/hardware mappings have been checked for consistency and integrity. The RTBP represents the loadable, executable version of the SBP.
- Mapping rules: optimise the software and hardware mapping against a set of constraints.

It must be noted that ASAAC standards and guidelines do not mention software, hardware or configuration blueprints at any moment. Only RTBP is mentioned in [2] like some kind of database storing all data necessary to manage the system.

Within TDP1.7 ETAP programme (hereafter ETAP programme), blueprints were modelled as Extensible Mark-up Language (XML) files with well-defined

semantic and syntax. The software blueprint is called Application Blueprint (ABP), and each CFM that forms part of the IMA Platform provides its Resource Blueprint (REBP), corresponding with the hardware blueprint. The Configuration Blueprint (CBP) models the whole system configuration and establishes the logical system architecture. It also defines both PE to RE and application to RE mappings, and system clock hierarchy. The integration process of these three blueprints into the System Blueprint (SBP) picks the relevant information up from each single ABP and REBP, and maps application to resources according to the information provided by the CBP and the mapping rules, particular for the target environment. The SBP is the input of the blueprint compiler and describes the IMA Core System both at software and hardware levels.

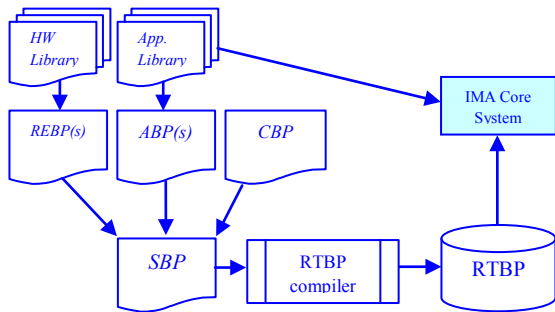


Figure 4. Blueprint development process

A development process based on blueprints enhances software reuse as well improves maintainability and flexibility. It could be possible to have libraries of applications and hardware modules to produce new systems.

### 3. SYSTEM RECONFIGURATION

Reconfiguration is the capability to adapt system functionally to the changing external or internal conditions of its environment, [6]. One of the benefits of IMA systems is this ability of moving from one configuration to another one in response to a limited range of events, so that system functions can be provided along time. It is therefore necessary to take into account that:

- System must be able to detect faults, either software or hardware.
- Once an event is triggered, the appropriate configuration to move to should be known.

- The system reconfiguration shall be done in a safely manner.

Most of the events are due to faults occurred within the IMA System, so in order to provide an effective fault-tolerance management it is necessary to identify correctly faults origin and type. In ASAAC standards, the FM and HM functions, part of the GSM functions, share this responsibility. The FM function monitors the system and report hardware and software faults and failures to the HM function.

The information confined within the RTBP is used by the FM and HM to determine if it is required to trigger the system reconfiguration process. During the blueprint development process, when producing the CBP, events to reconfigure are defined, while REBP provides fault masking information and error identification. The compilation of the REBP, ABP and CBP into the SBP provides enough information to the GSM functions to make decisions when faults or logical failures are detected.

#### 3.1. Reconfiguration process

Both higher performance and system integration characterise this new generation of control systems, as well as increased resource sharing. In order to exhibit fault tolerance in presence of errors giving any operational scenario, fault detection and health monitoring become a central point in IMA systems. It has to be possible to reconfigure this kind of systems to provide continued functionality when a system component fails (either hardware or software).

As noted in [6], three steps will have to be performed for a successful reconfiguration:

- Determine the appropriate configuration when a set of particular components, either software or hardware, fail.
- Determine the events that will trigger the need for a reconfiguration.
- Implement a mechanism that allows system transferring to a new one in a safely manner.

First two points are addressed during the design phase of the blueprint-based development process. System hazards are identified and system configurations are defined to mitigate system failures. Events are also outlined in this phase.

Mechanisms are implemented by the IMA Platform core services, particularly performed by GSM functions. The RTBP provides all the information that GSM functions require for reconfiguring the system in presence of well-known events. GSM functions therefore deal with the safely configuration transfer.

As introduced in section 2, system logical architecture is built around the IA concept. Reconfiguration also makes extensive use of this concept: sometimes the nature of triggered events impacts locally to a particular IA, and should not be propagated to other IA. Others, the entire IMA Core System should be reconfigured affecting the AC level to all lower IAs. It is clear that the IA concept provides some kind of soft partitioning as it implements a mechanism for fault isolation and recovery at system level, in contrast to spatial and temporal partitioning, as described in [9].

Therefore, safety issues of system reconfiguration have a two-fold view:

- Blueprints, as they should model dependable and safety requirements derived from RAMS analysis. Implicitly, the blueprint-based process must address this safety issues when producing blueprints.
- GSM functions, as they should allow transferring to a new configuration safely.

### 3.2. Safety-related aspects

The IMA concept permits much greater flexibility, as the system is able to fulfil different roles thanks to reconfiguration, as well as improving reliability and availability. However, if certification is difficult even in federated systems, it is even harder to demonstrate system safety on IMA due to the huge number of permutations of configurations. In order to deal with this complexity, [7] provides an approach to gain experience with IMA safety considerations:

- Step one: manual reconfiguration, which consist of a limited number of well-defined configuration options (generally a small number). If system has to deal with several roles, one configuration could be applied to one system role.
- Step two: static reconfiguration (on ground), which increases the number of possible configurations so it is more difficult to select them manually. As the number of permutations is higher, the assurance that the configuration is error free is lower.

- Step three: dynamic reconfiguration, the most difficult one, where reconfiguration is performed whilst the aircraft is flying. Decision-making is now based on an optimization algorithm that calculates the optimal configuration given a particular scenario. It has also to take into account the need for a safe transition between configuration states.

The blueprint-based development process defines activities to assure blueprint integrity, but it is still a problem how this integrity can be demonstrated for the certification process. Our experience during ETAP programme is extremely limited regarding this matter since we did not follow previous steps to gain enough knowledge. Our prototype is initially based on dynamic reconfiguration, and our optimization algorithm is replaced by the RTBP. All configurations and transition events were pre-established after system safety analysis, and GSM functions are responsible, especially CM function, to perform reconfiguration. Although main objectives of the programme were not focused on certification issues, we tried to incorporate them to the process. And it can be said that we put the craft before the horse.

Three major safety subjects must be covered in order to deal with certification issues:

- It must be demonstrated the integrity of blueprints along the process. And maybe the most important, demonstrating that the CBP is safe.
- FM and HM GSM functions shall need to be qualified to demonstrate that they detect, isolate and identify uniquely system failures.
- Finally, configuration transitions must be safely managed by GSM functions.

From [6] and [7] it is introduced the difficulties we have to face to deal with previous issues. Appropriate and qualified tools shall have to be developed to manage such complex configurations, and demonstrate blueprints integrity and safety along the development process. Modelling tools have a relevant importance here, as they could be able to model system dependability and safety issues, while generating automatically blueprints for all system views, i.e. application, platform and configuration views.

#### 4. INCREMENTAL INTEGRATION CONCERNS

Blueprints, the three-layer stack and the logical system hierarchy concept allow the development of the incremental integration and qualification concept. Main objective is the reduction of life cycle costs and improve overall system performance in terms of configurability and modularity. One of the most attractive promises is related to future system evolutions. It must be possible to re-use certification evidence for those parts of the IMA Core System that were not modified during the upgrading process. Otherwise no benefits are envisaged regarding migrating to IMA concepts from current federated approach.

Some areas of avionics system certification that are mainly affected by the use of IMA concepts are described in [8]. It is highlighted one of them that was repeatedly present in our work during ETAP programme: separating system functions in different hardware modules does not longer physically provide isolation. There are many logical interactions between the IMA Platform, application software and I/O devices. “Domino” failure effect could arise if logical interactions are not properly addressed as part of the dependability and safety plan performed at system level.

ETAP programme was also concerned on the incremental integration and qualification of IMA systems, and the activities necessary to implement these concepts into the blueprint-based development process. Our experience in upgrading a pre-existing IMA Core System consisted of adding a new application to an existing system, recording the necessary effort to apply system modifications.

First, system analyses need to be re-worked considering the interactions of the new functionality with the old one(s), i.e. already qualified. This analysis must include a Common Cause Analysis (CCA) so we can anticipate possible domino effect in case of error or failure. With the new function(s), new functional and non-functional requirements must be addressed on the CBP. Moreover, the system management hierarchy may be modified if new CFM is added to the IMA Platform, or new IA are considered after adding new functions. In order to simplify the scope of our work, this paper considers only the addition of a new application, maintaining the IMA Platform unchanged from the first IMA Core System version. This means that the CBP is only

modified in the functional IA that managed the RE where the new application is mapped. If the IMA Platform is not changed, REBP remain exactly the same ones used in the pre-existing system. The ABP is produced independently within each application development. Thus, old ABPs have not to be changed also.

However, at integration and qualification phase, many upgrades may be performed in order to include the new system function. Modifications on SBP are only applied to those PE that were affected due to changes in the management hierarchy. As explained, REs are mapped to PEs as defined in the CBP and is managed by a particular IA.

New/modified IA required new test scenarios to be developed, but we also had to perform regression testing on those IA that showed some kind of interactions with the new/modified one. This effect can be also extrapolated to the qualification process: it is necessary to demonstrate that the new functions do not disturb the old ones, although they are running on different CFM. Therefore, in order to reduce integration, verification and validation activities, it can be inferred that:

- Interactions between functional IA should be limited at a minimum. This would limit the regression testing necessary to integrate new or modified IA
- It seems sensible to highlight the enormous importance of the system design phase. An optimal logical architecture will allow the incremental integration and qualification to be real.

In this context, first conclusion derived from previous analysis is that the IA concept improves modularity in terms of change isolation and fault management. This property is important for IMA systems, characterized by a very high integration of system’s components.

The second important conclusion is referred to the necessity of strong partitioning at module level, as addressed in [8]. In fact, ASAAC software standard [2] does not specify any partitioning mechanism explicitly. The IMA Platform must guarantee that timing requirements are really covered, and no undesired effects arise when sharing common resources between applications mapped to the same CFM. Otherwise it would be almost impossible to present evidences of absence of undesired effects due to resource sharing.

Finally, blueprints have an important role in system development. Their integrity must be demonstrated so that each system configuration meets system requirements. Appropriate modelling techniques and tools are required to assist engineers on the assessment of the blueprint integrity and safety. For example, supporting the analysis at very early stages of system design, and the assessment of feasibility of the system configuration.

## 5. SPACE AND ASAAC SYNERGIES

Nowadays the space industry is paying special attention to current technological trends, especially those related to Interoperability, Composing and Reuse. The analysis of domains, which differ from the space one, provides an important chance in order to exchange experiences. One specific area of interest is modular distributed architectures, namely ASAAC [1], ARINC 653 [9] or AUTOSAR [10].

Space and Avionics systems share the same schedule and cost pressures and challenges:

- Mass and power (volume) saving,
- Reducing integration and validation costs,
- Integration to be independent from SW-HW life cycle
- Potential different criticality of application with the same computing node.

Space community agrees on the IMA concept, although with different technical rational and organisation and business cases. During ADCSS 2007 [11], concrete IMA technologies were identified as candidate to be applied for Space:

- Partitioning (spatial and temporal). Essential for run-time isolation of safety and security issues.
- Reference Architecture.
- Open standard definition of:
  - HW/SW Isolation.
  - Application / Executive abstraction of the platform interface.
  - Communication services (data exchange).

IMA impacts on the spacecraft avionics concerning the necessity for a definition of an open reference architecture, processes and tools, and new hardware technologies to standardise platform interfaces

(computer and HW/SW interface remain too much specific).

ESA has supported different IMA related activities as the Time and Space Partitioning working group and several ARINC related studies [12].

Additionally, the European Space Technology Harmonisation Dossier for On-Board software aimed to define families or reference architectures of space software. In the frame of this harmonization dossier, the activity "Framework for Domain engineering" (GSTP) was initiated [13]. One conclusion of the DOMENG study has been to promote the use predefined software components and architectures to eliminate redesigning and rebuilding the same software structures over and over again. DOMENG has highlighted the fact that the architectural models must be evaluated considering Dependability and safety issues, schedulability and tool supports.

System reconfiguration as presented in this paper may introduce an effective and efficient solution to deal with dependability and safety requirements.

ASAAC blueprint-based development process is certainly closed to current development approaches for Space based on the Model Driven Engineering paradigm. This is the case of ASSERT [14] and its automatic model transformations chains. There is a clear similitude between ASSERT and the concepts here explained: the Functional Viewpoint model corresponds to the Application Blueprints (ABP) here presented, Configuration Blueprint is related to the Configuration View, and the System Blueprint is very close to the ASSERT Deployment view. Synergies between both worlds exist and benefits can be derived from them.

Based on our experience, other attractive subjects for the Space domain are:

- ASAAC standard for CFM [3] hardware modules improves the interoperability of hardware modules between different architectures. This standard defines interfaces and basic functionality so that hardware modules are interoperable each other and open to multi-vendor markets and COTS technologies. The immediate consequence, an important cost reduction on hardware development.
- ASAAC standard for Software [2], and particularly the blueprint approach presented in this paper,

enhance software reusability. Applications are modelled by their Application Blueprints, at both functional and non-functional levels; re-use is supported by the ABP technique.

- The ASAAC three-layer stack [2] introduces a necessary separation of safety arguments for applications and hardware platform. With this approach Space system qualification may be simplified as the amount of qualification testing is reduced.

## 6. CONCLUSIONS

In the context of the ETAP programme, two applications were developed in parallel, but the integration, validation, verification, and qualification of both systems were performed separately. The second IMA Core System version was updated with the new application in order to exercise the actual effort on integration and qualification.

Although safety issues were not deeply covered, technical aspects and several safety considerations were derived.

The IA concept allows us to isolate changes in the system and confine regression testing on those old functions affected by the upgrade process. However it is strongly necessary to consider spatial and temporal partitioning at module level so that separation of safety arguments for applications and IMA Platform to be maintained.

It is also highlighted the importance of an efficient system design regarding the logical architecture selection, especially if it is intended future updates. IA interactions must be limited to a minimum in order to reduce the effort on integration, qualification and certification phases.

Finally, it is pointed the necessary adoption of Model Driven Engineering (MDE) to support system design and integration activities. Appropriate and qualified modelling tools must be able to generate blueprints, check their integrity and verify they fulfil dependable and safety requirements. Early resource consumption must be provided so the optimal system configuration, represented by the CBP, can be found without comprising additional iterations from integration phases to design phases due to resource exhaustion problems.

## 7. REFERENCES

- [1] ASAAC (2002). ASAAC Phase II Stage 2, *Final draft of Architecture Standards*.
- [2] ASAAC (2002). ASAAC Phase II Stage 2, *Final draft of Software Standards*.
- [3] ASAAC (2002). ASAAC Phase II Stage 2, *Final draft of CFM Standards*.
- [4] ASAAC (2002). ASAAC Phase II Stage 2, *Final draft of proposed guidelines for system issues – Volume 1: System management*
- [5] ASAAC (2002). ASAAC Phase II Stage 2, *Final draft of proposed guidelines for system issues – Volume 4: System Configuration /Reconfiguration*
- [6] Nicholson M. *Health monitoring for Reconfigurable Integrated Control Systems*. Department of Computer Science, University of York.
- [7] Jolliffe G., Nicholson M. *Exploring the possibilities towards a preliminary safety case for IMA blueprints*. Department of Computer Science, University of York.
- [8] Nicholson M., Conmy P., Bate I., McDermid J. *Generating and maintaining a safety argument for Integrated Modular Systems*. Department of Computer Science, University of York.
- [9] ARINC SPECIFICATION 653-1 *Avionics Application Software Standard Interface*, 2003.
- [10] *Automotive Open System Architecture (AUTOSAR)* [www.autosar.org](http://www.autosar.org)
- [11] ADCSS/07 - ESA Workshop on Avionics Data, Control and Software Systems (ADCSS) – ESA/ESTEC October 2007
- [12] ADCSS/08 - ESA Workshop on Avionics Data, Control and Software Systems (ADCSS) – ESA/ESTEC October 2008
- [13] *DOMENG Study Final Report* GMV-DOMENG-RP-011 Issue 1 April/09
- [14] *ASSERT Automated proof-based System and Software Engineering for Real-Time systems* - [www.assert-project.net](http://www.assert-project.net)