ARIANE-5-based Studies on Optimal Integrated Modular Avionics Architectures for Future Launchers

Björn Annighöfer, Çelen Nil**, Johannes Sebald** and Frank Thielecke**
**Hamburg University of Technology / SYSTAR Innovation*
Nesspriel 5, 21129 Hamburg, Germany
***Airbus Defense & Space, Airbus-Allee 1*
28199 Bremen, Germany

Abstract
The core avionics of the Ariane 5 launcher was developed in the early 90s and might be not optimal with respect to today's requirements and technological advances. In a study within the PREPARE project formal modeling and mathematical optimization are used to "rethink" the avionics. Based on the Integrated Modular Avionics (IMA) concept the best candidates for a future avionics architecture are derived. The results show improvement potentials up to 50%. The most promising architectures are analyzed in detail and relations between objectives and architectural artefacts are discovered. The study results and the used method seem important for future launcher development.

1. Introduction

The avionics system of the ARIANE 5 was designed in the early 90s. Studies to streamline the avionics architecture were carried out in the work-package Core Avionics of the PREPARE project, co-funded by the DLR. Research and pre-development activities aim at new concepts and advanced technologies to enhance the performance and processes of the future launcher. The launcher's avionics architecture shall be re-designed by adopting existing technologies of other industry sectors, namely Integrated Modular Avionics (IMA) and a formal model-based and algorithm-aided avionics architecting method.

The concept of IMA is standardized avionics hardware, which is shared by multiple system functions. A low number of hardware types reduce cost. Resource sharing reduces the number of devices and eases the creation of cross-system functions. Levering the full benefits of IMA, however, requires complex planning. Owing to standardization and decoupling of hardware and software, IMA architectures have a high degree of freedom. Multiplied with the high number of functions of the launcher avionics, a manual planning of the optimal device setup and function allocation seems hardly possible [1].

This is not the first article proposing IMA for the launcher. Monchaux [2] proposed an IMA-like avionics system and Jeremy [3] suggested a new Ethernet-based bus technology. In both works a certain platform is proposed and the benefits are analyzed. In this study, a model an algorithm based approach was used to retrieve the objectively optimal IMA platform from the scratch.

Model-based and automated planning of IMA architectures is an active field of research. Foerster [4], Gamatie [5], Delang [6], and Lafaye [7] proposed models for architecture planning. Most often the scope is dynamic analysis rather than architecture planning. Some use AADL [8] an architectural model. AADL, however, seems not rigid enough for automated architecture design. In addition to avionics models, methods and algorithms for automated planning of IMA architectures were proposed. Sagaspe [9], Al Sheikh [10], and Bieber [11] developed methods for optimal application distribution and scheduling. Salomon [12] proposed a method for automated redundancy structure generation. Methods for avionics network planning and scheduling have been proposed by Zhang [13], Al Sheikh [14], Carta [15], and Charara [16]. These approaches have only been demonstrated for small scale examples. Moreover, novel and contradicting objectives as energy need or feed-through cables shall be optimized. Therefore, a multi-objective approach is required as demonstrated in [17]. In this study an avionics architecture meta-model [18] highly integrated with a optimization framework [19] from aircraft avionics planning is adopted.

A domain specific model for avionics architectures is used to capture the systems and requirements, such as resource needs, safety segregations, and required sensors. A functional decomposition prepares the systems model for being distributed on IMA hardware. An anatomy model represents the sensor, actuator, and avionics installation locations, as well as possible cable routes. A hardware model holds devices, links, and hardware properties. Since there is not a single IMA concept or packaging, four possible IMA platforms are compared, including remote I/O, cabinets, and dual-lane modules. An optimization method called "device type optimization" is used to derive automatically the optimal number of IMA devices, their locations, and the sizing of devices in parallel. For each platform the optimal architecture for each of the design objectives is calculated and compared in a trade-off analysis.

In summery the following major issues are addressed in this design studies.

- Reducing the number of devices and device types by sharing resources
- Reducing the number of wires by a common avionics bus
- Reducing cable length by relocating and resizing devices
- Make improvements in non-traditional design objectives

This article is structured as follows:

- Chapter 2 recalls the current launchers avionics system, the scope of the study, and the model-based planning approach.
- Chapter 3 creates a formal architecture model of launcher functions and anatomy.
- Chapter 4 defines the IMA platforms to be investigated and design objectives to be optimized.
- Chapter 5 presents the results of twelve optimizations in terms of objectives and architecture.
- Chapter 6 discusses the results and the impact for the launcher
- Chapter 7 concludes the article.

2. Background, Aims and Methods

The aim is to identify the optimal IMA-based avionics architecture for the ARIANE 5 and future launchers. Therefore, it is necessary to review the current avionics systems, define the general scope and approach of the study and to introduce the computer-aided architecting method.

2.1. Current Avionics Systems

The Ariane 5 has a federated **avionics system**. There is approximately one device per function. All cross communications are realized by bus connections or separate physical wires. It is divided in the flight controls and management system (SSCV), the telemetry system (SSTM), the power distribution system (SSPE), and the safety system (SSSA). (cf. [20, 3, 2])

- The **SSCV** is mission critical. It stabilizes the launcher during flight and controls the different stages of the flight, as well as the precise delivery of the payload. Its main component is the On-Board Computer (OBC), which computes high-level control and mission management commands. Its main peripherals are inertial reference systems, engine control units, and structure separating pyrotechnics. Most components of the SSCV are located in the upper stage on a ring shaped structure, called the Vehicle Equipment Bay (VEB). The devices communicate over a common MIL-STD 1553 bus. Additional status inputs are received from various temperature, pressure, flow, vibration, and flow sensors from all over the launcher. These mostly analog or discrete inputs are acquired by data conditioning and acquisition units located in each structural compartment. The SSCV is fully redundant, such that each component, except some sensors and actuators, and the bus exist twice.
- The **SSTM** is not mission critical. Its purpose is to collect various sensors and status data during flight and to sends it down to the ground control station. Trajectory, position, vibration, and shock data is used to prove a successful mission to the customer and for offline engineering. The main component is the telemetry data conditioning and acquisition unit. It provides an RF link to the ground and is capable to extract and transmit data spied from the MIL-BUS. In addition, it listens to many sensors of other systems by additional physical wires connected to remote conditioning units.
- The **SSSA** ensures safety in case of catastrophic failures during the first mission phases. If errors such as a wrong orientation or trajectory are detected it initiates a safe neutralization of the entire launcher. The SSSA is fully redundant and independent of other systems. It furthermore provides on- board means to support an accurate localization and ground trajectory tracking during the first flight phases.
- The **SSPE** system controls the power distribution during different flight phases and the switching of individual devices. Its main components are redundant power distribution boxes. The SSPE supplies devices from SSCV, SSTM, and the payload from various batteries or the ground power supply.

In total the current avionics system is composed of 44 devices and approximately additional 1100 sensors and actuators. These are connected by two MIL-bus systems, some proprietary bus systems, and much analog and discrete I/O.

2.2. Study Scope and Approach

Current avionics devices are all composed of computing, communication, and physical I/Os. These are also the main building blocks of IMA modules. However, in IMA the processor, memory, bus connection, and I/Os are shared. It is, therefore, assumed that functions of one system, which are currently distributed over several devices, can be hosted in one IMA module. Hardware sharing might not only occur in one system, but for example SSCV and SSTM share a lot of sensors and information, such that putting them on the same hardware would save a lot of wires. Moreover, by combining functions and wirings the optimal locations of devices may change, such that **relocation and resizing** of devices might save additional wire length.
Another point of improvement is the inter-device communication. The current MIL-BUS is robust, but limited in its bandwidth and not flexible in topology signal distribution. Therefore, it is assumed to be insufficient in future. AFDX and TTEthernet suggested as possible candidates for a **common uniform avionics communication bus**.
Last, traditional avionics functions are design primarily for function and optimized for simple design objectives as mass. By using a computer-aided design methodology **non-traditional design objectives** as cost, or operational reliability might be improved without any drawbacks.

The focus of the optimization is on the SSCV and SSTM system. Since SSSA and SSPE have a low number of functions that can be distributed. Moreover, the SSSA is completely independent of other systems.

The general approach for the study is (1) to carry out a functional decomposition of the current avionics system in redistributable blocks, (2) to build a formal model of functions, hardware and anatomy, (3) to define potential IMA target platform templates in their basic physical properties, (4) implement the design objectives, (4) apply a device type definition and sizing optimization algorithm to derive the set of Pareto optimal architectures, and (6) to select the most promising candidate for further investigations.

2.3. Computer-aided Avionics Architecting

For the study an implementation of computer-aided architecting method is used documented in [21]. This method is based on a meta-model for avionics architectures and its requirements, and a set of optimization routines for single and multi-objective architecture optimization. From the optimization routines the so called device type optimization is utilized. Model and optimization are implemented in the Avionics Architect, which was used for the study.

2.3.1. Avionics Architecture Model

For the design studies an avionics architecture meta-model published in [18] is used. The meta-model is based on six basic layers. The five most important layers are definitions, systems, hardware, installation, and mapping.

- The **definitions layers** contain objects that are building blocks of an architecture, e.g. device types, wire types, and resource types. Most important it contains the platform definition, i.e. which device types are available, which resources are provided by the device types, and which functions can be hosted.
- In the **systems layer** functions are modeled logically. It is a static model of atomic functional blocks called tasks. In addition, safety and performance constrains can be assigned to tasks. To host tasks it is necessary that the device provides the resources the task requires (CPU, memory, I/O, etc.). Tasks communicate by signals that also require resource (e.g. bandwidth) and must be mapped to networks or bus systems. Modeling of fixed system sensors, actuators or other equipment is carried out as peripherals, which can be hardwired to certain tasks.
- The **hardware** layer contains the devices and interconnecting topology. Devices are instance of device types, e.g. end-systems or network switches. Links between devices express point-to-point connections.
- The **installation** or **anatomy** layer expresses a graph-like topology of installation locations and cable routes. Installation locations are rooms for devices. Cable routes are connections between location that can be used to put links or peripheral wires. Therefore, locations can be assigned witch generic infrastructure resources consumed by devices, like sufficient space for device installation. Cable routes are assigned with a fixed length. Cable route segments can split or join at cable route joints.

- Systems, hardware, and installation layers are independent. The **mapping** layer is used to define an operational avionics architecture. The mapping layer references one systems, one hardware, and one installation layer. For each object of the referenced layers an assignment object exists. For instance, device assignments put devices from the hardware model to locations from the installation model. Task assignments bind tasks to devices. For the same input different mappings can be created.

2.3.2. Multi-Objective Device Type Optimization

The inputs for the study are the functions and peripherals of the current avionic system, basic properties for the new IMA avionics devices such as dimensions and mass, and the launchers anatomy. The aim is to identify the optimal number, position, and resource provision of the devices for multiple, partially contradicting objectives. Based on the model, the device type optimization as published in [22] seems most suitable.

The general procedure of device type optimization is depicted in Figure 1. The inputs are the system tasks with its resource requirements. Moreover, device type templates must be available that specify which functions and resources a device can host. The maximum number of resources per device is limited. A typical limit - also used in this study - is the electronic board area in the device and the area required by a certain resource (e.g. CPU or I/O). The third input is the vehicle anatomy with the possible installation locations for the devices, which are restricted by infrastructure resources. From this information an overdimensioned max-architecture is derived, that contains each device as often as possible if the device type would be considered on its own. It is neglected that location capacities are exceeded. The same applies for the resource instances on the device types. Finally, the tasks are allocated to the device with the constraint that no device limits and location capacities are exceeded.

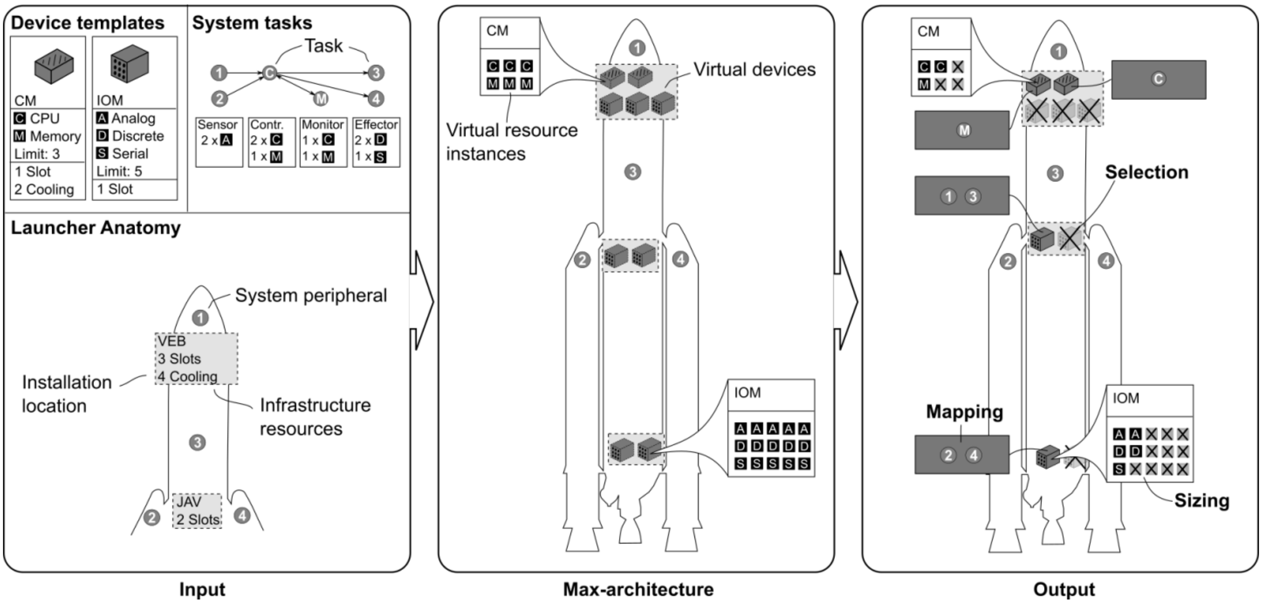

Figure 1: Flexible framework of eight optimization routines for avionics architecture optimization.

Task allocation is a combinatorial optimization problem expressed as a **multi-objective binary program (MOBP)**. A MOBP looks for a solution vector x that minimizes multiple cost vectors

$$f_1^T x, f_2^T x, \dots \qquad \textbf{(1)}$$

such that

$$Ax \le b \qquad \textbf{(2)}$$

$$x \in \{0,1\}^n . \qquad \textbf{(3)}$$

In device type optimization x encodes the allocation of tasks, the usage of device and device types, as well as the resources required per device type. The inequalities ensure that all resources, capacity, device limits, and segregations are hold. For further detail please refer to [22].

Instead of a single solution the Pareto optimum is calculated. Formally, this is the set of solutions for that no other solution exists that is superior in all objectives (dominant). For the engineer it is the set to the best trade-offs. An example is given in Figure 2. The Pareto optimum is obtained using an iterative single-objective optimization routine [23] named Pareto-front-sampling as depicted in Figure 3.

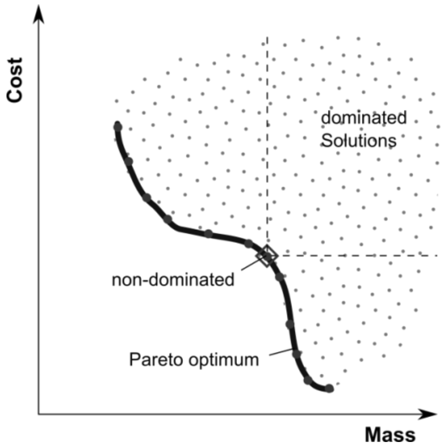

Figure 2: Pareto optimum

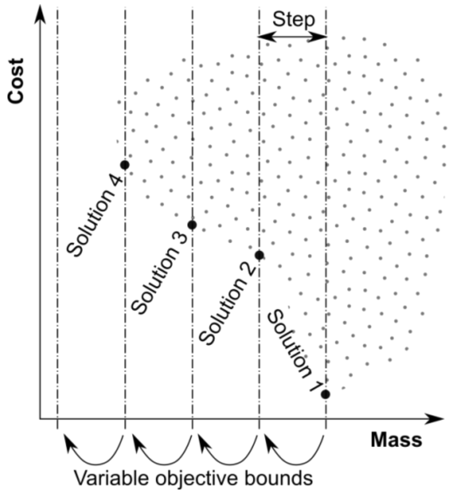

Figure 3: Pareto-front-sampling

2.3.3. Avionics Architect

Meta-model and optimizations are implemented in the **Avionics Architect** (AA) [24]. It is a planning environment for early verification, evaluation and optimization of avionics systems. In the Eclipse based software avionics architectures and system requirements can be modeled. It is possible to automatically verify architecture aspects and to carry out on-the-fly evaluations. Architecture variants can be compared. Optimizations are implemented in MATLAB and controlled from Eclipse. MATLAB reads the native model files and converts them into mathematical optimization problems. It has configurable interfaces to COTS MILP solvers, as CPLEX, GUROBI or SCIP. The results are stored in updated architecture models. In addition, the AA provides helper functions for architecture visualization, analysis, and report generation.

3. Launcher Avionics Architecture Model

An avionics architecture model is the input for the design studies and optimization. Important are the system functions and peripherals, as well as the launchers anatomy and constraints to be compliant to.

3.1. Anatomy Model

The Ariane 5 launcher is composed of six main structural elements. As depicted in Figure 4 these are the **main stage (EPC)**, two **boosters (EAP1 and EAP2)**, a **spacer (ETF)**, the **upper stage (ESC)**, and the **payload compartment (COIFFE)**. For the design studies an anatomy model is required, that defines possible installation locations for future avionics devices and the positions of the remaining devices, actuators, and sensors. In addition, the routes for cables between the locations are needed. The main locations for avionics equipment are the ESC, the upper part of the EPC (EPC_JAV) and lower part (EPC_JAR). The avionic compartment of the ESC is divided in two separated opposite half rings with the locations VEB_z and VEB_zn. Those half rings form the Vehicle Equipment Bay (VEB), which hosts most of the core avionics equipment. Moreover, ESC, EPC, and EAPs have a lot of equipment in the engine structure (EBS_BM and EPC_BM) and nozzles (EPC_MOT and EAPx_MOT). The spacing between installation locations, the cable routes, and joints have been derived for the Ariane 5 ME concept with a height of 61

m. The length and joints of the launcher are depicted in Figure 4. In total there are 20 locations for new IMA devices and 158 additional locations for peripherals. 221 cable routes interconnect the locations.

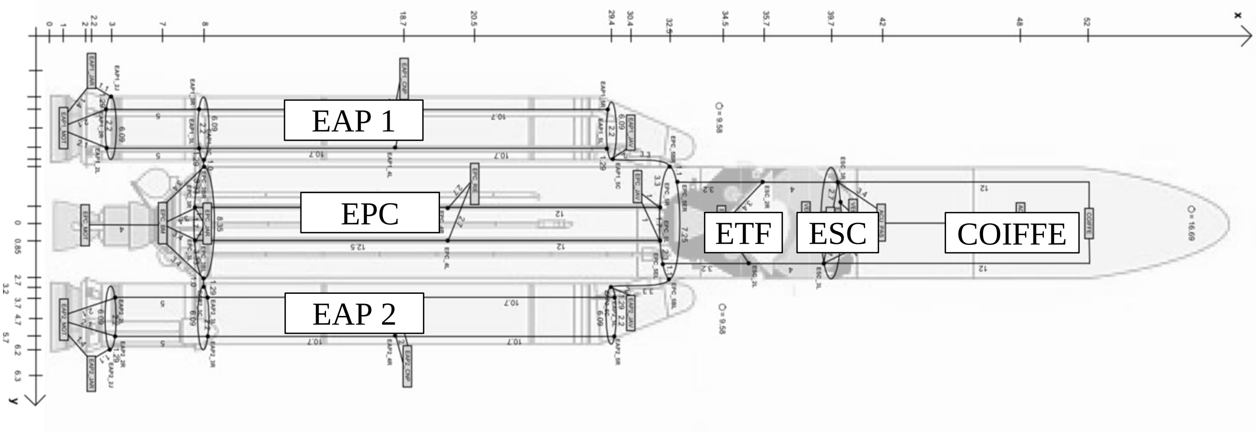

Figure 4: Dimensions and lengths of the anatomy model

3.2. System Functions Model

A model of the Ariane 5 system function is created by defining which a device will be replaced. Completely replaced devices are modeled as tasks that can be redistributed. Devices that remain as legacy equipment, are modeled as peripherals and I/O tasks

The main controller of the **SSCV** is the On-board Computer (OBC). The major functions of the OBC are flight control, pyro technique commanding, electro valve control, telemetry, and fault detection. To relocate the functions of the OBC it is divided in eight sub functions listed in **Table 1**. A static CPU load is estimated in relation to the current OBC. In percentages it is given how much computing time is spent on the individual functions in a fixed period of time. In addition, the OBC and other replaced SSCV components communicate with 505 sensors and actuators. Considering redundancy these result in 620 I/O tasks requiring different types of analog, discrete, pyro, electro valve interfaces.

Table 1: Functional decomposition of the OBC applications

Function	**Description**	**CPU load [%]**
OBC_FC	Flight control	20
OBC_OBT	On-board time	5
OBC_TM	Telemetry	10
OBC_SO	Sequential order of pyro techniques	15
OBC_EV_EBS	Electro valve control of ESC	10
OBC_EV_EPC	Electro valve control of EPC	10
OBC_FDIR	Fault detection and isolation	10
OBC_PW	Control of power distribution	5

The main function of the **SSTM** is the acquisition, recording, and downstream of functional and non-functional data during the flight. In case of a new IMA-like avionics architecture, it could be possible to distribute the telemetry data acquisition and pre-processing on distinct devices. To enable a distributed mapping of the SSTM, the SSTM processing is decomposed in six functions listed in Figure 5. Each function collects the date in the individual section of the launcher. CPU load is estimated in comparison to the OBC to 10% for each task. Peripherals of the SSTM result in additional 479 I/O tasks requiring different analog, discrete, and RF interfaces.

In **summary**, the modeling and decomposition of the launcher's systems resulted in a systems model of 1140 tasks, 986 peripherals, and 1135 signals. From the 1140 tasks 22 are computational task of the OBC and SSTM decomposition. The remaining 1118 are sensor data acquisition or actuator control. 184 I/Os are electro valve drivers.

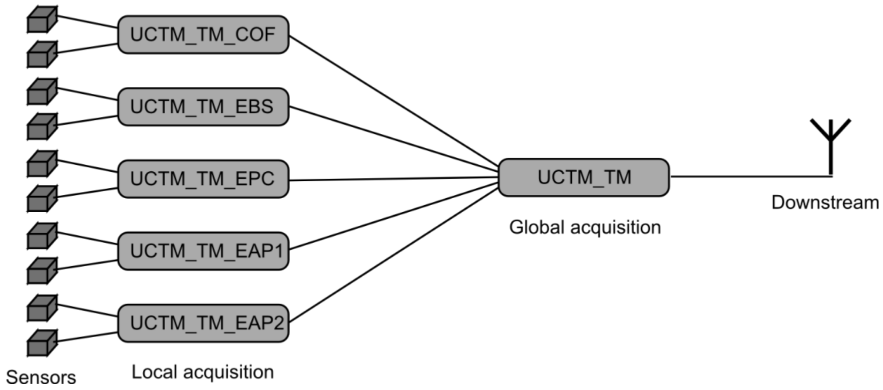

Figure 5: Acquisition structure after decomposition of SSTM acquisition functions

3.3. Constraints

Three additional issues have to be considered to retrieve a valid mapping in respect to the launcher by automatic methods.

3.3.1 Segregation

The SSCV system is redundant. Most device, sensors and actuators exist twice or are at least connected twice. To ensure a fully operational redundancy, a segregation constraint is introduced. This means that no pair of functions from SSCV1 and SSCV2 is allowed on the same device.

3.3.2 Symmetry

For all segregated functions it is in addition assumed that the allocation of both lanes must be symmetric. Therefore, many functions of the SSCV are mapped either symmetrically in left and right locations, or symmetrically on the left and right lane of dual-lane modules.

3.3.3 Structural changes

It must be considered that the structure of the launcher changes during the flight. Sections are separated. It must be ensured that no function is mapped to a device that is separated before the function has fulfilled its purpose. In the model location constraints have been used to prohibit such false mappings. For instance EPC functions are allowed on EPC or ESC devices, but not on EAP devices. EAP functions are allowed on all devices.
The same issue applies to cables. Even if a correct mapping of functions is assured, a cable could be routed through a section separated too early.

4. Launcher Avionics Optimization Study

There are four platforms and four objectives to be optimized. Three objectives are used as primary optimization objectives. Four objectives are used for the evaluation.

4.1. Potential IMA Platforms

Four common IMA platforms from current aircraft or IMA research are suggested for an IMA-based Ariane avionics system. All platforms require a common avionics bus named Launcher Data and Communication Network (LDCN). Dimensions of the four platforms are chosen similar to aircraft IMA module, i.e. approximately 3kg per module and 120000 mm^2 electronic board area. The electronic board area of the resource is derived from the existing avionics devices as they are. These resource sizes are approximately ten times that of current aircraft resources.

4.1.1. Platform 1: Computing- und I/O-Modules – CIOM

Computing and I/O Modules (CIOM) provide computing resources and physical input and output interfaces (I/O). Computing and I/O are shared between functions. CPU, memory and buses are used by functions in parallel. Primitive I/Os, as analogs or discretes are used exclusively, but exist several times. Each CIOM is a stand-alone device with housing and power supply. Figure 6 depicts an example of a CIOM-based architecture. Core Processing and I/O Modules (CPIOM) of the Airbus A380 and A350 are examples for a CIOM platform.

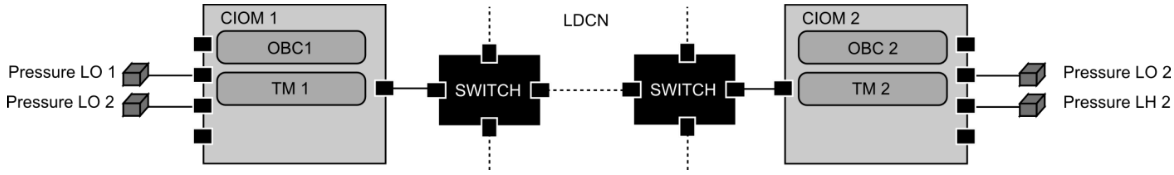

Figure 6: Example of a CIOM-based architecture

4.1.2. Platform 2: Separated Computing and I/O – CM + IOM

A distributed IMA platform is created by separating computing and I/O resources on different device types. Computing Modules (CM) provide computing and memory resources. System sensors, actuators and other equipment are connected by I/O modules (IOM). CMs and IOMs communicate over the LDCN. Computing and I/O is currently separated in all actual developments of the next generation IMA (IMA2G) [25]. It enables new redundancy concepts and helps to save cable lengths and mass. Figure 7 shows an example architecture with CMs and IOMs. This concept is partially implemented with the Common Remote Data Concentrators (CRDC) of the Airbus A350.

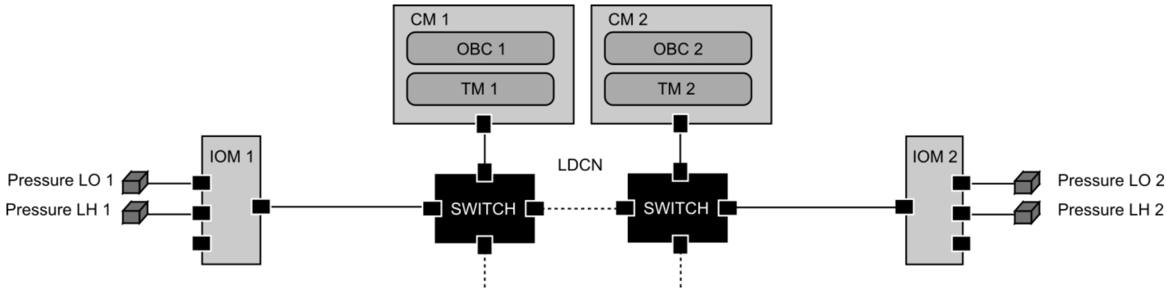

Figure 7: Example of an architecture with separated computing and I/O modules

4.1.3. Platform 3: IMA-Cabinets – CB + IOB

Further reductions of housing and housekeeping (e.g. power supply) mass are possible with cabinets. A cabinet is a standardized chassis. This chassis is equipped with standardized blades, which provide computing (CB) or I/O (IOB) resources. By choosing the installed blade types individually for each cabinet position it is possible to save housing mass. In addition, the number of spare resources caused by standardization is reduced. Within aircraft VITA 46/48 VPX [26, 27] seems currently most desired. Figure 8 shows sample architecture with cabinets composed of one computing and two I/O blades each.

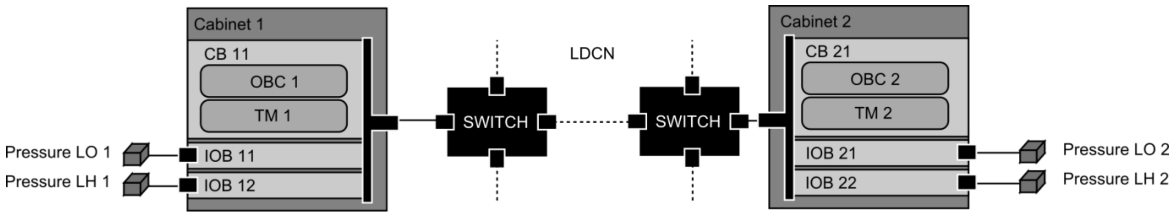

Figure 8: Example of IMA-cabinets equipped with computing and I/O blades

4.1.4. Platform 4: Dual-Lane Computing and I/O Modules – DCIOM

A dual-lane computing and I/O module (DCIOM) is a special variant of a cabinet. It is unmodifiable cabinet and has two blades. Chassis and housekeeping are simpler, since a DCIOM is not mechanically configurable. The basic concept is to have two lanes of a redundant function in one chassis. On each electrical board one lane of the system function is executed. In current research projects DCIOM boards are developed, that provide computing and I/O resources. Both boards in one device provide the same resources. An example is given in Figure 9.

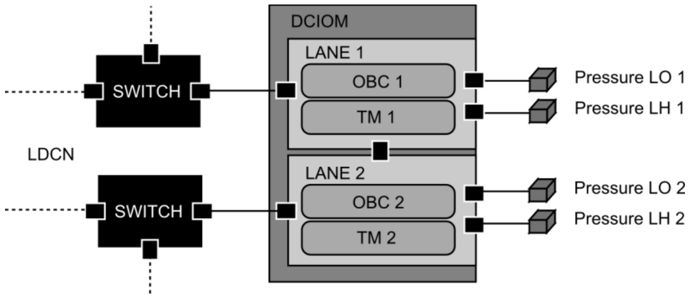

Figure 9: Example of a dual lane module architecture

4.2. Objectives

Mass, energy need, feed-through cables, and center of gravity have been selected as objectives to be implemented for device type optimization.

4.2.1. Mass

The **mass** of the avionics system sums up all device masses, cables, peripherals, and fixations. Cable mass depend on the cable lengths and wire types. In case of cabinet based avionics, also the masses of all utilized cabinet housings must be taken into account. Summing up the mass objective it is calculated as

$$m^{tot} = m^D + m^W + m^C \tag{4}$$

where m^D is the mass of all devices. m^C is the mass of optionally used cabinet housings. m^W is the mass of LDCN and peripheral wires. m^C calculates as $l_i \cdot \overline{m}_j^W$. For each cable i, its length l_i, and the specific mass $\overline{m}_j^W$ of its wire type j.

4.2.2. Simplified Energy Need Calculation for Staging

Not all parts of the launcher and, therefore, not all avionics components reach the same height. For instance all components located in the boosters fly only up to the first separation point. This is a static modelling, since implementation does not consider the mass flow rate due to unknown figures relating propellant, change in velocity and hence specific impulse etc. However, the energy need has a direct impact on propellant amount required to reach a certain height with given masses.

The energy need is calculated as acceleration work plus lift work. Both depend on the components mass, but also on the height and velocity reached at the end of each flight phase, i.e. the separation. Energy need defined calculated as

$$E^{tot} = E^L + E^A. \tag{5}$$

The lift work is calculated as

$$E^L = \gamma m m_e \left(\frac{1}{r_e} - \frac{1}{r_s}\right) \tag{6}$$

with the gravitational constant γ, the earth mass m_e, and the earth radius r_e. m is substituted by m^D, m^W and m^C for the individual components. r_s is the earth radius plus the reached height. h_s, i.e. $r_s = r_e + h_s$.

The acceleration work is

$$E^A = \frac{1}{2} m v_s^2 \tag{7}$$

m is the mass of the individual components and v_s is the velocity reached at the separation point. In this study the heights and velocities from a standard geo-transfer-orbit flight (GTO) from [28] are used.

4.2.3. Feed-through Cables

During the separation of structures electrical connections and busses are cut. It is desired to have as low as possible feed-through cables at separation points, e.g. the connections between boosters and the EPC. This objective is calculated by marking some cable routes as separation points. The number of wires through these routes summed up over the launcher is minimized. The objective of separated cables is calculated as

$$SC^{tot} = \sum_{SP_i \in \{SP_1,\ldots,SP_n\}} \sum_{w_j \in SP_i} c_{w_j}\,. \tag{8}$$

c_{w_j} is the number of conductors of cable j in separation point SP_i. This is summed up over all separation points of the launcher, i.e. $\{SP_1, \ldots, SP_n\}$.

4.2.4. Center of Gravity - CoG

The **center of gravity** objective is defined as the radial offset from the YZ-center of the launcher.

$$COG^{tot} = \sqrt{y_{COG}^2 + z_{COG}^2} \tag{9}$$

The offsets in Y and Z direction are calculated as

$$y_{COG} = \frac{1}{f_m} \sum_i m_i y_i \tag{10}$$

and

$$z_{COG} = \frac{1}{f_m} \sum_i m_i z_i \tag{11}$$

respectively. The mass of each component m_i (device, cable, or cabinet housing) is multiplied with its y or z coordinate (y_i and z_i). The summed up momentums are divided by the total mass of all components f_m. For cables the momentum is calculated for each segment inside a cable route. The position of the cable is calculated as the center between both end-points of the cable route.

4.2.5. Auxiliary Objectives

Two secondary objectives are used to obtain architectures of higher quality. Secondary objective means that this objective is only optimized if the optimum in the main objective leaves room for improvement without changing the main objective value. This is a so-called lexicographic optimization. First, the **distance of controller functions** to the connected peripherals is minimized. Second, the **scattering of systems** is minimized. Details on that are published in [29].

4.3. Optimization Setups

All optimizations are carried out using the **device type optimization routine**. Pre-calculations have shown that obtaining the full Pareto-Optimum is infeasible for the current size of the model and the study duration. Therefore, the Pareto optimum is estimated using the three optima for the single objectives. This results in three independent optimization runs per platform. Moreover, one device type per platform shows to be insufficient. For each platform two independent I/O devices types (T1 and T2) with equal properties are used. Each optimization has either mass, energy need, or feed-through cables as objective plus the auxiliary objectives. In total twelve optimizations are carried out and twelve architectures are calculated. The architectures resulting from each optimization are evaluated in all four objectives and compared.

5. Results

Twelve optimizations were carried out. The results are summarized in Table 2. The values of the four objectives mass, energy need, feed-through cables, and center of gravity offset are given. For each objective and contribution the minimum is underlined. Moreover, the optimization time is given in seconds. For each optimization there was a time limit of five hours. In the four optimizations minimizing the number of feed-through cables it was possible to obtain the global optimum . All other optimizations are optimal up to an unresolved uncertainty. The optimality gap is the maximum percentage the primary objective could be improved by an unknown solution. However, there is no guarantee for a better solution.

Table 2: Numerical results of optimization studies (minima are underlined)

Solution Nr.	**Platform**	**Design Objective**	**Mass [%]**	**Energy need [%]**	**Feed-through cables [%]**	**CoG YZ Offset [m]**	**Run-time [s]**	**Optimality Gap [%]**
0	0		100	100	100	100		
1	1	Mass	86	87	80	48	18000	11,4
2	1	Energy need	87	87	77	46	18000	16,3
3	1	Feed-through Cables	105	98	<u>46</u>	64	564	0,00
4	2	Mass	85	85	67	44	18000	7,54
5	2	Energy need	87	85	72	<u>42</u>	18000	15,5
6	2	Feed-through Cables	105	97	<u>46</u>	72	468	0,00
7	3	Mass	85	85	72	47	18000	5,37
8	3	Energy need	86	<u>85</u>	62	51	18000	7,24
9	3	Feed-through Cables	106	96	<u>46</u>	87	772	0,00
10	4	Mass	<u>82</u>	87	89	52	18000	22,3
11	4	Energy need	84	86	71	51	34521	30,1
12	4	Feed-through Cables	104	102	<u>46</u>	72	2785	0,00

The minimum mass is achieved for DCIOM platform (platform 4). Compared to the current avionics system this is a reduction of 17.5%. Considering energy need, the optimum is achieved for the cabinet platform (platform 3).
The minimal energy need is 24.5% lower than for the current avionics system. The variation in energy need is smaller as for other objectives. Optima for feed-through cables have significantly higher energy need and mass. Mass and energy need are correlated, but not equal. Several solutions with higher mass and lower or equal energy need exist. The gains of the energy need optimizations are, however, low.
The center of gravity (CoG) has a theoretical improvement of 52.4%, which however neglects the contribution of structure and fuel to the CoG. Moreover, the CoG has not directly been optimized. The number of feed-through cables has a minimum with 54% less connectors compared to the reference.

Figure 10 depicts a scatter plot of the reference architecture and the optimization results. Each section compares exactly two objectives. Each marker represents one architecture. The objective values are normed from minimum to maximum. Eight of twelve architectures dominate the reference solution in all objectives. Five solutions have higher mass or energy need as the current avionics system. In almost every comparison a separation in two clusters can be observed formed by solutions optimized for mass and energy need on one side and solutions optimized for feed-through cables on the other side. The current avionics system is the only architecture that is not part of any cluster. The solutions in the mass and energy need cluster perform similar.

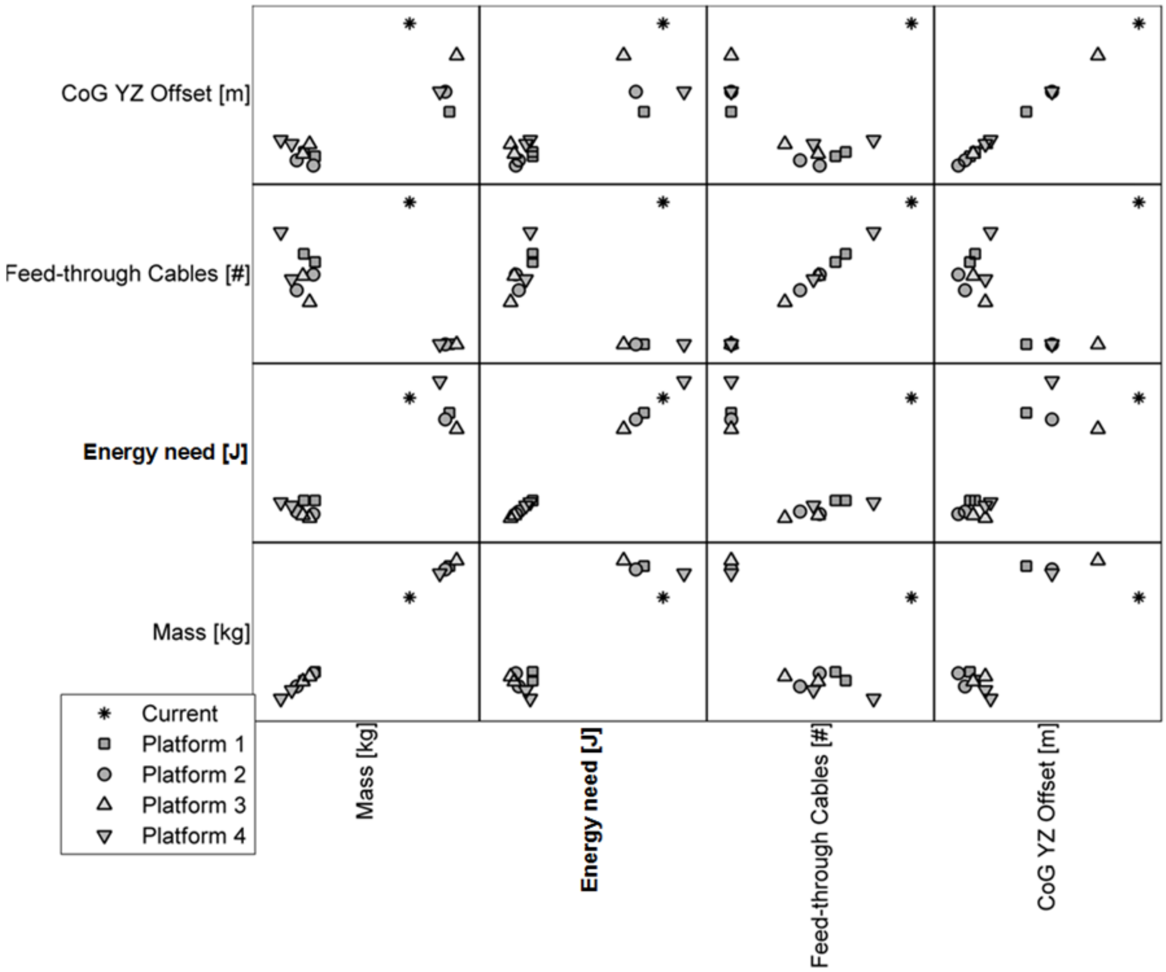

Figure 10: Scatter plot of all architectures retrieved from optimization compared to the current design in the four design objectives

6. Discussion

Optimization studies show possible **improvements** in all four objectives. Up to 17.5 % in mass, 25% in energy need, and 54% in feed-through cables are possible. In comparison to the current avionics system all derived architectures outperform the current avionics system in at least two objectives. Eight of for solutions are significantly better in all objectives.

Looking at the resulting **architectures**, mass is reduced on one hand by reallocating I/Os and wires. Up to 30% wire mass is saved. On the second hand and less obvious, also the device mass is reduced by up to 30% by merging SSCV and SSTM and by moving functions. Devices have, however, the same physical properties as the current avionics. Reducing the mass also reduces partially the energy need. The optimization of feed-through cables results in a separation of the launcher's sections in terms of wires. Consequently, there is a big contradiction between mass and energy need versus the number of feed-through cables. Reducing the number of feed-troughs increases mass significantly.

Noticeable, is that the **optimization** for feed-through cables resulted in a single optimum of 54% less connectors, and showed the fastest convergence. A small but still noticeable difference exists between energy need and mass. Higher mass, does not necessarily result in higher energy need. The results show that architectures with more mass can have reduced energy need. Moreover, more devices can be beneficial for routing and spare. Since energy need - as a kind of a corrected mass - should be of primary interest for launcher avionics. The mass objective might be obsolete in future, while increasing the capacity for avionics. This requires, however, a detailed verification and maybe an adaptation of the simplified energy need equations. Even the not optimized center of gravity seems to be improved. The latter, however, does currently not take any non-avionics masses.

Looking at the four tested **platforms**, the solutions of all platforms are close by. There is no single winning platform. Dual-lane modules are the winner in mass, cabinets have the lowest energy need, and CMs+IOMs are superior on average. CIOMs are not superior, but it uses the lowest number of modules and module types on average. The overall difference in objective performance between optimized architectures is small, since all platforms have relatively equal properties. Nevertheless, differences are observed. CIOMs are the traditional solution. With them only two device types are necessary, and the lowest number of devices is achieved. However, the overall demand for computing resources in the current model is low, such that the highest CPU spare exists. The use of pure computing CMs in platform 2 results in perfect CPU utilization. Overall it seems to have the best usage rate of devices. Cabinets are beneficial in wiring mass, although they have in total the highest number of devices with larges total board area. This points out that the chosen form factor might be too small to efficiently place the relatively large I/Os. Dual-lane modules have the highest degree of freedom in the wiring, and this indeed results in a minimum wire mass. Moreover, it has the finest distribution of CPU resources, which is beneficial for the CPU usage and might also be beneficial for local control loops. Dual-lanes architectures have, however, deficiencies in energy need and feed-through cables.

Some **general effects** are visible for all platforms. If mass is important, architectures tend to focus on the upper stage. There is the largest number of devices, and many I/Os from the main stage and boosters are routed to the VEB. Energy need optimization tends to remove unnecessary wires from the upper stage and results, therefore, in more devices of the lower stage. The optima in feed-through cables is achieved if all I/Os are hosted locally.

Comparing the results to **aircraft IMA** systems, it must be stated that the number of devices and blades is very high. The reason is that the used board areas of I/O resources are approximately ten times larger that typically for aircraft IMA. It could not be clarified if this difference is caused due to miniaturization, higher environmental robustness (e.g. radiation hardening) or different I/O characteristics. It is assumed that smaller I/Os are possible, which would increase the benefits dramatically.

In **summary**, all platforms might bear some potential further for improvement. However, even now all architectures are clearly beneficial to the current avionics system. Therefore, the search for a new avionics system for the launcher and also the use of IMA technology seems unquestionable from the technical point of view. Moreover, the methods and tools used here for modeling and optimization uncovered architectures and effects that would hardly be known otherwise. Moreover, they shortened significantly the design time and increased architecture maturity. It is, therefore, proposed to continue the use of methods and tools in further architecture design.

Three **variants** seem **favorable**. Platform 2 should be chosen if high similarity to current architectures and simple and independent integration is required. Platform 3 cabinets are beneficial in all objectives and bear a high degree of freedom for configuration and upgrade. Platform 4 should be chosen if local control and safety are most relevant. In any case, further refinements of models, objectives, and platforms are suggested before making a decision. Moreover, some differences between aircraft and launcher IMA have to be kept in mind.

- The Ariane 5 avionics system degrades over its mission. This means parts of the avionics architecture must be safely separated. The remaining part of architecture must continue to work.
- Pyrotechnic interfaces do not exist in civil aircrafts. There is, therefore, no IMA hardware available with pyrotechnical I/Os
- Radiation resistance is much more crucial for the launcher avionics. It is likely that no current IMA hardware is suitable for space flights.

7. Conclusion

This study investigated the potential benefits and architectures for the Ariane 5 launcher using Integrated Modular Avionics (IMA) known from civil aircrafts. A model-based architecture optimization framework named Avionics Architect was used to find optimal architectures for four different IMA platforms. Platforms range from combined calculation and I/O devices to separated calculation, cabinets, and dual-lane devices. A detailed model of the current launchers sensors, actuators, and anatomy was created. All major cable length, I/O types, and wire types have been taken into account. An automatic calculation of optimal device sets for the for mass, energy need, and feed-through cables revealed improvements of 17% to 54% in single objectives compared to the current avionics system. Moreover, the strengths and weaknesses of the platform could be identified. There are three architectures, which are favorable candidates for the next launcher avionics system. These should be considered in further avionics architecture design. Whereby each architecture still bears additional potential to be leveraged. In addition, results show that the Avionics Architect is suitable for the identification of a launcher reference architecture and has the capability to assess potentials for improvement of existing architectures.

7.1. Outlook

This report was a first study on optimizing the launcher avionics and IMA technology. Although significant benefits have been shown, there is room for improvement. Some imperfections and open points have to be addressed on the way of finding really the optimal avionics system for the launcher.
Considering the architecture model, the current avionics, peripheral, and anatomy model is most precise in the upper stage. More precise modeling of lower stage, boosters, and payload seem necessary. Considering the launcher avionics, higher benefits could be achieved, if

- including the SSPE and SSSA in the optimization,
- optimizing the LDCN,
- considering current level of interface miniaturization (very important!),
- try new installation locations for avionics,
- and placing cabinets more optimally or combine cabinets with smaller remote I/O gateways

Moreover, the used method could be extended in the following points to extend the scope of the study:

- Implement objectives for power management optimization, reliability assessment, thermal considerations.
- Realistic modelling of staging principle considering propellant types and delta-v to be reached.
- Develop an optimization method for reliability.
- Develop an optimization method for power distribution, and other spliced cables.

References

[1] Butz, H. 2007. Open integrated modular avionic (IMA): State of the art and future development road map at airbus deutschland. In: *Proceedings of the 1st International Workshop on Aircraft System Technologies*. 211–222

[2] Monchaux, D. and P. Sangare. 2012. Avionic-x: A demonstrator for the next generation launcher avionics. In: *ERTS2012 - Embedded Real Time Software and Systems*.

[3] Jeremy, R., J. Georges, and T. Divoux. 2013. Toward self-reconfiguration of switched Ethernet architectures in the next generation of space launchers. In: *SPACOMM 2013, The Fifth International Conference on Advances in Satellite and Space Communications*. 17–22.

[4] Forster, S., M. Fischer, A. Windisch, B. Balser, and D. Monjau. 2003. A new specification methodology for embedded systems based on the pi;-calculus process algebra. In: *Proceedings. 14th IEEE International Workshop on Rapid Systems Prototyping*. 26 – 32.

[5] Gamatie, A., C. Brunette, R. Delamare, T. Gautier, and J.-P. Talpin. 2006. A modeling paradigm for integrated modular avionics design. In: *SEAA '06. 32nd EUROMICRO Conference on Software Engineering and Advanced Applications*. 134–143.

[6] Delange, J., L. Pautet, A. Plantec, M. Kerboeuf, F. Singhoff, and F. Kordon. 2009. Validate, simulate, and implement arinc653 systems using the AADL. In: *Proceedings of the ACM SIGAda annual international conference on Ada and related technologies*, SIGAda '09. 31–44.

[7] Lafaye, M., M. Gatti, D. Faura, and L. Pautet. 2011. Model driven early exploration of IMA execution platform. In: *IEEE/AIAA 30th Digital Avionics Systems Conference (DASC),* 7A2–1 –7A2–11.

[8] Van Baelen, S., T. Weigert, I. Ober, H. Espinoza, and I. Ober. 2010 Model based architecting and construction of embedded systems. In: *Models in Software Engineering*.

[9] Sagaspe, L. and P. Bieber. 2007. Constraint-based design and allocation of shared avionics resources. In: *26th AIAA-IEEE Digital Avionics Systems Conference*.

[10] Al Sheikh, A., O. Brun, , and P. Hladik. 2009. Decision support for task mapping on IMA architecture. In: *Junior Researcher Workshop on Real-Time Computing (JR-WRTC2009)*. 31–34.

[11] Bieber, P., J.P Bodeveix, C. Castel, D. Doose, M.Filali, F. Minot, and C. Pralet. 2008. Constraint-based design of avionics platform - preliminary design exploration. In: *4th European Congress ERTS Embedded Real Time Software*.

[12] Salomon, U. 2014. *Automatic Design of IMA-based Systems*. PhD thesis, Faculty of Aerospace Engineering and Geodesy of the University of Stuttgart

[13] Zhang, S. 2008. *Communication Infrastructure Supporting Real-Time Applications*. PhD thesis, Technische Universität Hamburg-Harburg.

[14] Al Sheikh, A., O. Brun, M. Chéramy, and P. Hladik. 2012. Optimal design of virtual links in AFDX networks. In: *Real-Time Systems*. 1–29.

[15] Carta, D.C., J.M.P. de Oliveira, and R.R. Starr. 2012. Allocation of avionics communication using boolean satisfiability. In: *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DAS)*. 6C1–1 –6C1–12.

[16] Charara, H., J.-L. Scharbarg, J. Ermont, and C. Fraboul. 2006. Methods for bounding end-to-end delays on an AFDX network. In: *18th Euromicro Conference on Real-Time Systems.*

[17] Annighöfer, B. and F. Thielecke. 2012. Multi-objective mapping optimization for distributed modular integrated avionics. In: *31st Digital Avionics System Conference*.

[18] Annighöfer, B., E. Kleemann, and F. Thielecke. 2011. Model-based development of integrated modular avionics architectures on aircraft-level. In: *Deutscher Luft- und Raumfahrtkongress*.

[19] Annighöfer, B. and F. Thielecke. 2015. A systems architecting framework for optimal distributed integrated modular avionics architectures. In: *CEAS Aeronautical Journal*. 1–12.

[20] Carrier, T. 2005. Radiation hardening - study of a special case: Ariane 5. http://sirad.pd.infn.it/-scuola_legnaro_2005/Presentazioni/C06_Carriere_Thierry.pdf.

[21] Annighöfer, B. 2015. *Model-based Architecting and Optimization of Distributed Integrated Modular Avionics*. Dissertation, Hamburg University of Technology.

[22] Annighöfer, B., E. Kleemann, and F. Thielecke. 2013. Automated selection, sizing, and mapping of integrated modular avionics modules. In: *32st Digital Avionics System Conference*.

[23] Ozlen, M. and B.A. Burton. 2011. Multi-objective integer programming: An improved recursive algorithm. In: *arXiv*, (arXiv:1104.5324v1).

[24] Annighoefer, B.. TuTech Innovation GmbH - competece center systar. http://www.systar-innovation.de/en/, 2014.

[25] Fuchsen, R. 2009.Preparing the next generation of IMA: A new technology for the SCARLETT program. In: *IEEE/AIAA 28th Digital Avionics Systems Conference*. 7.B.5–1 –7.B.5–8.

[26] ANSI/VITA 46.0-2007, 2013.

[27] ANSI/VITA 48.0-2010, 2010.

[28] Arianespace. 2011. *Ariane 5 Users Manual*, 5 edition.

[29] Annighöfer, B., C. Nil, J. Sebald, and F. Thielecke. 2015 (to appear). Scatter & symmetry aware IMA architecture optimization: Use case Ariane launcher. In: *34th Digital Avionics System Conference*.

Acknowledgments

The results presented originate from an avionics modernization study of the ARIANE 5 and future space launchers Avionics within the Airbus Defense and Space *Programme to Enhance Upper Stage Performance and Reliability for Future Expendable Launchers* (PREPARE), co-funded by the *German Federal Ministry of Economic Affairs and Energy* (BMWi, contract code 50 RL 1220) and administrated by the Space Administration of the *German Aerospace Center* (DLR). Their support is greatly appreciated.
launch

Supported by:

Federal Ministry
for Economic Affairs
and Energy

DLR

on the basis of a decision
by the German Bundestag

The author of this publication is responsible for its content.