

LARISSA: Layered Architecture Model for Interconnection of Systems in UAS

Emerson A. Marconato, Daniel F. Pigatto and
Kalinka R. L. J. C. Branco
Institute of Mathematics and Computer Science
University of São Paulo
São Carlos, Brazil
Email: {emerson, pigatto,
kalinka}@icmc.usp.br

Luiz Henrique C. Branco
Federal Institute of São Paulo - IFSP
Araraquara, Brazil
Email: luiz.branco@ifsp.edu.br

Abstract-Architectural models have been used to enable the development of more appropriate and structured systems, from the simplest to the most complex. The use of models in embedded systems is important especially when it comes to critical embedded systems. In such systems, as unmanned aerial vehicles, these models are designed to allow compliance standards, reduction in production time, reduce and make easy the development and maintenance process. Critical embedded systems have specific requirements, such as high reliability and real-time response, security and performance. The definition of an architectural model that allows these aspects is taken into account, and that provide compliance with the standards and enables accurate and rapid development. This is innovative and allows the scientific community and the industry to have benefits. In this sense, the LARISSA aims at developing an architectural model for systems interconnection in unmanned aircraft systems (UASs), allowing features reuse and certification of these aircrafts.

Keywords - *Architectural model, UAV, UAS, critical embedded systems, IMA, certification*

I. INTRODUCTION

Technological and scientific advances have been providing various changes in various sectors of industry, commerce and services. Embedded systems are computer systems that are part of a larger system. These systems provide, in most cases, real-time monitoring and control. They are considered safety-critical when possible failure may result in loss of life or high-value assets [1], [2], [3], [4] and [5]. Both hardware and software in embedded systems have become increasingly complex. Multicore and multiprocessor systems have become common, which has further increased the complexity of software [6]. Moreover, they are becoming more common and both can be seen in homes and in business environments where they have been used for the control or management information. The advances in technology allow these systems with greater processing power, memory, and adaptation to different needs, and can even communicate with any other device, embedded or not.

UAS is a typical application of a critical embedded system. The term UAS was adopted by the FAA (Federal Aviation Administration) and by the international academic community to describe a system that includes not only the aircraft, but also all the associated elements such as payload, ground control station and communication links [7]. UASs have been widely used in precision agriculture, national security and environmental monitoring. Several papers have been published in this area, demonstrating the feasibility of using such vehicles as important tools for performing precision agriculture and the environment monitoring [8], [9], [10].

There are different types of UASs that have even different capacities. Some aircraft can fly autonomously, following a pre-programed flight path (based on a grid or a sequence of waypoints) [8], while others can fly receiving commands from ground stations operated by pilots.

The size of the aircraft can range from micro to large, and the ground control station can be implemented on smartphones, tablets, laptops or networks of workstations (distributed control stations).

Thus, the aircraft may vary not only in size, but also in shape and type of propulsion performance. The ground station interface can vary from a joystick to a tangible user interface (for example, a table with tangible augmented reality). The performance of the communication links and the type of payload are also very important to fulfill the mission intended for the system.

Specialized literature mentioning that UASs will become popular in the next 10 years, performing different missions, from agricultural border inspections to the automatic cargo transport [11], [12], [13], [14] and [15].

This growing use of UASs should allow them become common. Therefore, architectures, that enable the organization and more specific definition of the components that make up these embedded systems (UASs), enables the easy development of hardware and software that compose them.

Once the components of a UAS may be divided into aerial segment and ground segment, and each segment may be subdivided, the subdivision segments in these layers

allows the system subdivision into subsystems which can have different implementations and separate the architecture components in levels of criticality. These layers can be represented by models, which are intended to serve as guides for the development of UASs, specifying how it will interconnect the components.

The rest of this paper is organized as follows. In Section II, we present the related work in the field of Reference Model Architecture. Section III describes the requirements for architecture models.

Section IV presents LARISSA a new architecture model for UASs, all the details about LARISSA are also presented. Finally, Section V presents our conclusions regarding this work, as well some future prospects for this research.

II. RELATED WORK

This section presents a review of Reference Model Architecture in embedded systems and critical embedded systems. The Reference Model Architecture found in the literature can be classified in Federated (traditional architectures) and Non-Federated (non-conventional architectures).

A. Traditional architectures (*Federated*)

NIST (National Institute of Standards and Technologies) provides a reference model for UAVs [26]. In this particular pattern, the reference model was proposed to specify military rules, practices and controls in a comprehensible and intuitive way for a human commander. The architecture approach proposed is different from that required in our project, focusing on a lower level of abstraction, which the layers specify what components found in a UAS system should do.

1. *The Project of Pastor, Lopez and Royo*

The work proposed in Pastor, Lopez and Royo [27] is based on software and hardware architecture designed to operate the controller mission and the payload in mini/micro UAVs (Unmanned Aerial Vehicle). According to the authors, the innovation is the use of a distributed hardware architecture which is easily scalable by the use of LAN architecture-based software subscription services, communication abstraction layer and flow of execution based on mission planning. According to the authors, the high level of modularity offered by a LAN provides flexibility for coupling the microprocessor type most appropriate to use the module, given its functional requirements.

2. *MCAP*

In Olson and Burns [28] it is proposed another architecture model. This is Phase III of the project called MCAP (Manned/Unmanned Common Architecture Program), used by the U.S. Army for use in UAVs such as FCS (Future Combat Systems) and C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance).

The MCAP architecture, Phase III, is based on electronics and commercial off-the-shelf (COTS) and open standards interfaces. The development objectives of the model was to define and develop an architecture capable of supporting a number of UAVs in the U.S. Army platforms demonstrating the performance of the resulting system in a laboratory environment.

The development of this model relied on the study of three classes of UAS: Unmanned Combat Armed Rotorcraft (UCAR), a combat helicopter unmanned, Class IV Medium Altitude Long Endurance (MALE) and Extended Range/Multi-Propose (ERMP), with two aircraft: Fire Scout and A-160, and FCS Class III, Shadow 200.

3. *RFCSA*

In Deng, Ma and Zhu [29] the authors proposes a project called Reconfigurable Flight Control System Architecture (RFCSA). The development project was performed according to the following prerequisites: simple, unified interface between modules - the function module should be easy to be added, removed or replaced for specific applications; easy for high level applications design - low level of implementation details needed allowing the user to develop high level applications easily; Hardware and Software Reuse - Hardware and software developed for one application can be ported to another without major modifications; Low cost - the product should be cheap, expanding the market of small UAVs; Reliable - the hardware and software must be easy for development and verification.

The RFCSA consists of a series of modules that meet five of the requirements mentioned above. All modules are connected via logical and physical interface modules. The module interface is unified.

4. *LMA*

The project presented by Neto et al. [30] is a modular embedded architecture, consisting of three levels: embedded systems, communication link and inertial navigation system.

The purpose of the project is to design and build a platform for UAV research and development of autonomous behavior. The proposed architecture consists of modular embedded electronics and communication protocols based on the OSI model. This modular architecture consists of a main circuit with a microcontroller/processor and a number of modules required for the performance of tasks such as: inertial and non-inertial unit, unit aid to navigation, accelerometer, magnetometer, barometer, GPS unit motor control and others.

B. *IMA architecture*

IMA (Integrated Modular Avionics) constitutes an architecture widely used in conventional aircraft type and have extensive use in the recent literature for UASs.

Prisaznuk [31] proposed the IMA (Integrated Modular Avionics), an integrated model of avionics that is used as architecture for conventional aircraft. IMA was initially proposed to be used in commercial and military aviation. It

is set around the concept of high computational processing power and OS modules that allows independent processing of the application processing software. The modules share hardware resources and are allocated in offices, which have well-defined interfaces with the aircraft.

According to Watkins and Walter [32], it is possible differentiate IMA architecture from the federated architecture (conventional). The authors state that IMA architectures provide sharing of processors when processing information, communications and I/O. The resource row is divided for use of multiple avionics functions. The avionics functions served by IMA can be from different companies and their criticality is guaranteed due to the robust partitioning mechanisms that are inherent in the architecture. In contrast, federated avionics architectures implement independent collections of dedicated computing resources (CPU, communications and I/O) for each avionics function normally contained in Line Replaceable Units (LRUs) or Line Replaceable Modules (LRMs). Still constitute characteristics of IMA, according to [33]: the use of open standards and provision of a single data bus to interconnect the major aircraft systems.

Advantage of the IMA compared to federated are the economy of space, weight and power consumption or SWaP (Space, Weight, and Power), due a single unit performs various functions. Another advantage is the consolidation of hardware; it has several applications running on fewer processors [33].

IMA architecture was used entirely in the design of commercial aircraft Boeing 777 and Airbus A380, the Boeing military aircraft KC767A.

Recently, IMA architecture has been applied in the design of UASs.

1. The Project of Ilarslan, Bayrakceken and Arisoy

The project proposed by Ilarslan, Bayrakceken and Arisoy [34] is based on the development of avionics system for a mini VTOL (Vertical Take-Off and Landing) for purposes of academic study of flight dynamics, using a derivative of the IMA architecture.

The construction of a mini UAV is possible, according to the authors, due to the emergence of new technologies such as electric motors brushless (no brush system for electrical power induced) and Li-Po (Lithium Polymer) that are lighter compared to lead-acid or nickel-metal, most common to be found. The project objectives are to review the concepts of architecture avionics systems and develop flight control algorithms. The authors understand that beyond the basics of IMA, the prerequisites of this architecture are also the application of open standards and use of common hardware and software modules, from the data bus to the software language, the use of Commercial off-the-shelf (COTS) at all levels, the use of Real-Time Operating Systems (RTOS) commercially available that support protection (DO-178C), security (MILS - Multiple Independent Levels of Security/Safety) standards and partitioning (ARINC-653) and the development of software that meets the required standards.

2. ANKA

Another project that uses IMA architecture is the ground control station used in the Turkey aerial force known as ANKA [35]. It is related to the ground control station used in conjunction with the category MALE UAV (Medium Altitude Long Endurance) flying at 30,000 feet and a range of 200 km. The ground control station uses two types of software, the real-time, responsible for flight control, telemetry and interface, which are developed in ANSI C language and run on two PowerPC computers, working in a redundancy system Master/Slave; and non-real-time, responsible for recording telemetry and images, preflight testing and other features that are developed with object oriented methodology and C# and .Net language.

The development of both types of software follow some standards such as the use of NATO STANAG 4586 - overall architecture and message interfaces for control stations [36]; RTCA DO-178B - guide to developing critical software for avionics, which defines processes, objectives and activities for each level of criticality [37]; IMA and standard software development partitioned ARINC-653 - the software will benefit from partitioning of time and space. According to the authors, the use of IMA in the design of real-time software employed in the ground control station help modularization, protection and split time in the software components. The use of this model allowed development where each developer was responsible for one module and one of them was responsible for the integration of the parts, which facilitated the development and verification of the final product.

3. Linux-based ARINC 653

In the work proposed by Jin et al. [38] the authors chose IMA as an alternative to federated architecture due to the increasing use of computing resources in avionics are generating problems of SWaP (Size, Weight, and Power).

IMA architecture has been suggested to solve the problems and provide better SWaP consolidation and testability of the software by means of modularization in parts. This division into parts provides an efficient way to integrate various real-time applications in a transparent manner and in the same computing device, while providing isolation in the execution environment (that means no interruptions from a device on another not unless this is desired). ARINC 653 defines basic resources for IMA and standardized APIs [38].

This session presented the concepts and definitions of architectures, model and reference models. Several papers were presented that assist in defining the reasons for building a new architectural model for interconnecting systems for unmanned aerial vehicles.

Even though there are various models and architectures in the open literature, none of the presented models include all the requirements for making a UAS that is not specific. A summary of requirements met by each project is presented in Table 1.

This proposal differs from others, as this is the architecture at a higher level of abstraction. None of them establishes or specifies the architecture level glimpsed here,

with references on models of layered architecture for the interconnection of systems in UAS almost nonexistent.

Table 1
Comparing all the works found in the open literature.

Layer / Project	MCAP	Project of the Pastor, Lopez and Royo	RFCSA	LMA	Project of Ilarslan, Bayrakeeken and Arisoy	ANKA	Linux-based ARINC 653
Physical (Aerial Segment)	-	partial	partial	partial	√	-	-
Distributed RTOS	partial	√	partial	partial	partial	-	partial
System Abstraction	-	-	-	√	-	-	-
Monitoring & Control	partial	-	partial	partial	partial	-	√
Navigation & Services	partial	√	partial	√	√	partial	partial
Mission	√	√	partial	√	√	-	√
Physical (Ground Segment)	-	-	-	partial	-	√	-
Ground Station	-	partial	-	partial	√	√	partial

III. THE ARCHITECTURE MODEL REQUIREMENTS

A model is an abstraction representation, from which it can be evaluated in a rational manner, the instances properties of this model. Any model should enable the description of all possible instances of previously modeled concept.

A reference model, on the other hand, is not directly tied to any standards, technology or other concrete implementation detail. It provides a common semantics that can be used unambiguously across and between different implementations.

In the literature, the terms architectural model, architectures, reference models and reference architectures are treated as synonyms, although they have distinct meanings, even though in some cases the difference is subtle.

Architecture is a structure that identifies, defines and organizes components. The relationship and the principles of design of components, functions and interface established between subsystems can also be defined by architecture [16].

Moreover, a reference model for architecture is an architecture that the entities, relationships and information units involved in the interactions between and within

subsystems and components are defined and modeled. In short, it is a model that incorporates the basic purpose and the idea of the system and can be considered as a reference for various purposes [17] and [18]. The term architectural model used in this study reflects exactly that last statement: incorporate the basic goal and ideas of the system.

The terms architectural styles and reference architecture are also commonly used interchangeably [19]. However, some authors [20] stated that reference architecture could be seen as a structure that provides a characterization of the functionality of a given application field software systems. There are reference architectures for many application domains, such as: for performance evaluation [21] avionics [22]; to test environments [20]; distributed applications [23]; for systems e-commerce [24]; for Web servers [25].

The increasing use of UAVs should cause them to become increasingly common. In this scenario, the techniques proposed in this work will facilitate the development of automated applications for UAVs, allowing these vehicles be more easily inserted and incorporated into the airspace, contributing to their spread.

In order to propose a broader understanding of the component parts of a UAS system, we propose a layered model, which can be subdivided as needed. Figure 1 illustrates the model.

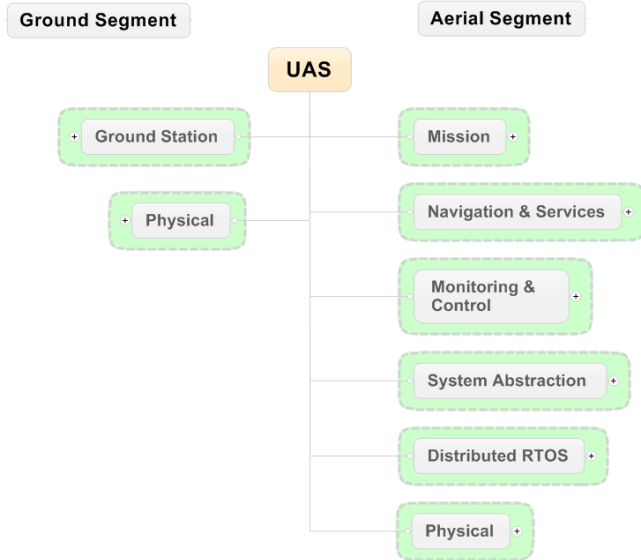


Figure 1: LARISSA – The proposed reference model.

IV LARISSA: LAYERED ARCHITECTURE MODEL INTERCONNECTION SYSTEMS IN UAS

In LARISSA model, the components of a UAS may be divided into an aerial segment and ground segment. The aerial segment is hierarchically composed of: (i) physical layer, (ii) distributed RTOS (Real Time Operating System) layer, (iii) system abstraction layer, (iv) monitoring and control layer, (v) navigation & services layer, and (vi) mission layer. Ground segment is divided into (i) physical layer and (ii) ground control station layer.

The separation into layers allows the system to be divided into subsystems that can have different implementations and assists the separation of the parts that compose a complex critical embedded system in different levels of criticality. Thus, the advantages offered by the service-oriented architecture can be applied in sections of low criticality, making the development of these sections

simpler and more flexible. Among the possible advantages, we can mention: separate the environment, improving the reusability and maintenance, higher level of abstraction and interoperability, more interactive interfaces between devices, certification of components and configuration information systems, and the ease in use of services supplied to the system by external servers.

These layers can be represented by models, which are intended to serve as guides for development of UAS systems, specifying how it will interconnect the various components such as sensors, control circuits, GPS, payload, communication with the ground control station, and others.

In Information Technology, a layered architecture is used to define the specific responsibilities of each layer and the interconnection among them. Based on an architectural model, the hardware manufacturer or software designer can develop their products knowing exactly which layer will interact in UASs, which are the input and output parameters, and what type of connection must be used.

According to [39], certain principles must be applied to define the layers: a layer must be created where there is need for other level of abstraction; each layer must perform a well-defined function, which should be chosen aiming the definition of standard protocols; layer limits must be chosen to reduce the flow of information transported among the interfaces; the number of layers should be large enough, so that different functions do not need be placed unnecessarily in the same layer, and small enough, so that the architecture does not become difficult to control. Architecture to be considered complete should define what each layer could perform, specifying services and protocols that are used in each one.

Figure 2 illustrates the architecture model proposed in hierarchical way. Moreover, the papers related to UASs in the literature have UASs implemented using traditional approaches [40], [12], [13], [14] and [15]. On the other hand, there are roadmaps illustrating the progress expected for UASs, published periodically by military organizations such as the United States Air Force. These roadmaps mention that for the future it will be necessary adopt an architecture open, standardized and scalable enabling the rapid addition of modular functionality.

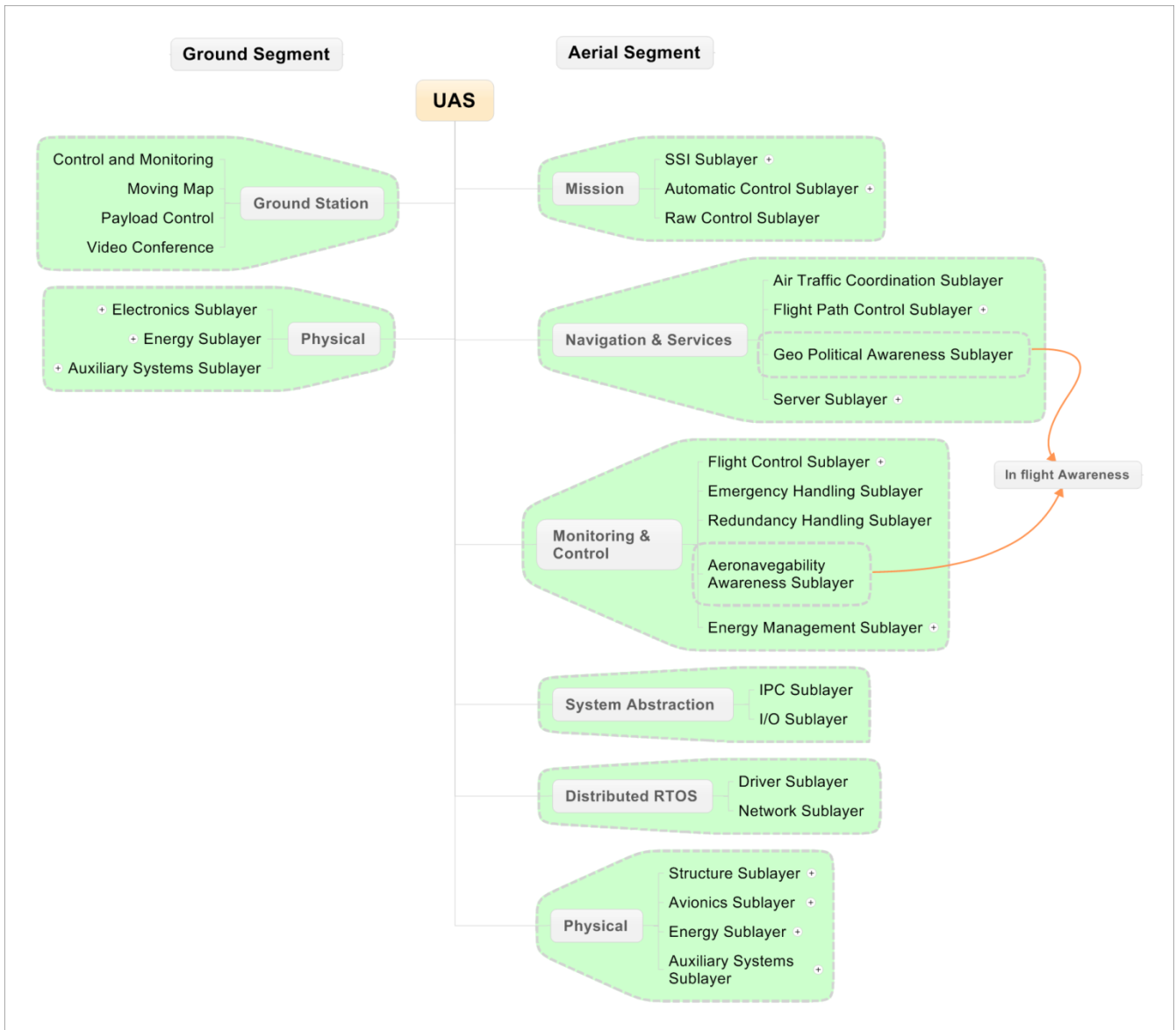


Figure 2: LARRISA: Proposed model and its first layer.

Each layer is composed of sub-layers, which are described in the next subsections.

A. Physical Layer

The aerial segment's physical layer is the aircraft's hardware layer, which is decomposed in the structure, avionics, energy and auxiliary system sub-layer. Each sub-layer may be subdivided into more specific sub-layers.

1. Structure Sub-layer

The structure sub-layer represents the aircraft's categories used as UAV according to their aerodynamic characteristics. These sub-layers are shown in Figure 3.

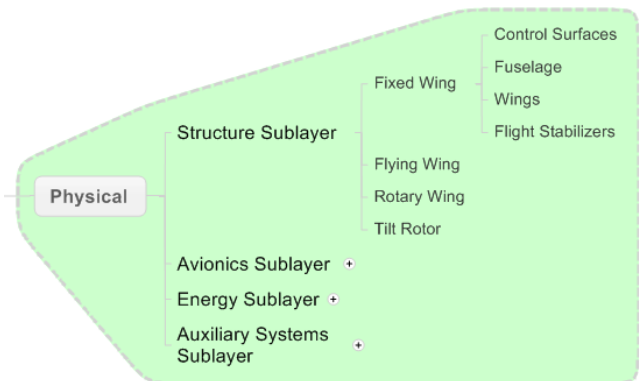


Figure 3: Structure Sub-layer.

The Fixed Wing sub-layer represents the aircrafts that have the same components of these aircraft in conventional models, such as Control Surfaces (ailerons, flaps and moving parts of the rudder and stabilizer flight). The fuselage can be understood as the body of the aircraft, which are fixed wings

and flight stabilizers. The Wings are the components of the aircraft that provide support, one of the forces involved in flight. The wings may vary according to profile and scale, and these variations influence the aerodynamics of the aircraft. Flight Stabilizers are responsible for maintaining the stability of the aircraft, which is basically formed by the rudder and horizontal stabilizer or rudder depth.

In Flying Wing sub-layer there is an aircraft that is shaped like a wing, not having a fuselage.

The Rotary Wing sub-layer includes the aircraft with the same characteristics of a helicopter, and the wings are actually the blades of the propellers. The quad-rotor and its variants fall into this category

The Tilt Rotor sub-layer specifies the aircraft that has the unique characteristic: helicopter/airplane combination aircraft.

2. Avionics Sub-layer

The aircraft's electronic systems shown in Figure 4 are described in Avionics sub-layer, which is composed of sub-layers number.

The sub-layer Avionics Controller presents electronic components responsible for: Flight Control - keeping the stabilized flight; Navigation Control - which operates in conjunction with the Flight Control and responsible for following the flight plan; Control subsystems - operate in other avionics control, such as the payload.

The sub-layer Avionics Communications is subdivided into Aerial/Ground, which describes the subsystems responsible for the data transmission on the Aerial Traffic Control, Telemetry, Data Sensors installed in the aircraft and Data Relay, like a relay station. The other subdivision is Aerial/Aerial, which describes the components of communication between aircraft for the Coordination of Aerial Traffic Coordination Squadron or, when multiple UAVs are operating in the same airspace under the coordination of one of these.

The antenna sub-layer describes the bands in which the aircraft can perform data transmission such as ISM (Industrial, Scientific and Medical), LEO (Low Earth Orbit), GEO (Geostationary Orbit), VHF and GPS. Also, this layer is composed of Stationary or Positional antennas, the latter may have their positions changed mechanically or through electronic.

The sub-layer Infrastructure defines Electrical Wiring and Electrical Connectors to be used in Avionics sub-layer.

A Servo Motors & Actuators sub-layer specifies electromechanical devices that operate primarily on the control surfaces of the aircraft.

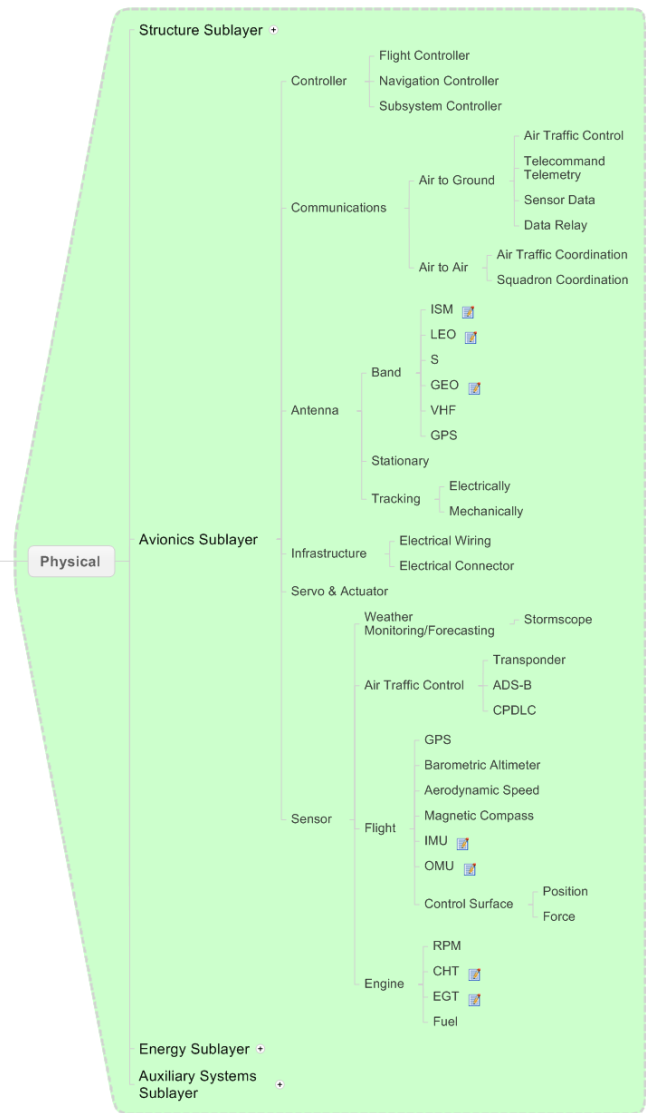


Figure 4: Avionics Sub-layer.

The Sensor sub-layer is the sub-layer of Avionics responsible for interacting with the sensors and to translate this information to monitoring, navigation and other systems. Compose this sub-layer: Weather Monitoring Forecasting; aerial traffic control, which can be done by transponder, ADS-B and CPDLC; Flight, with its subsystems: GPS, Barometric Altimeter, Speed Aerodynamics, Magnetometer, IMU (Inertial Measurement Unit), SMO (Optical Measurement Unit) and Control Surfaces observing the Position and Force; Engine, with sensors for reading RPM, CHT (Cylinder Head Temperature), EGT (Exhaust Gas Temperature) and Fuel (reading level).

3. Energy Sub-layer

Systems for supplying power to the UAV, as well as propulsion systems are treated by energy sub-layer, as illustrated by Figure 5.

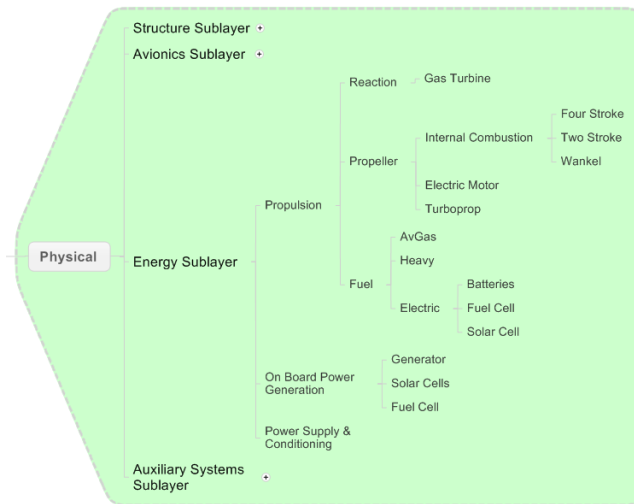


Figure 5: Energy Sub-layer.

This sub-layer is divided into three sub-layer namely: Propulsion, this sub-layer is listed with its categories Reaction (turbines) and Helix, which may be for Internal Combustion engines with four time cycles, two times cycles or a Wankel motor. In the Helix sub-layer we still have the Electric and Turboprop engines. As for the fuel we have: Aviation Gasoline, Kerosene and Electric divided into Batteries, Fuel Cells and Solar Cells; the Embedded Energy Generation, larger UAVs may contain proprietary systems for power generation for electronic components or even their engines, these being: Generator, Solar Cells and Energy Cells; a Power Supply and Conditioning treats the most appropriate place and way to install the energy systems in UAVs.

4. Auxiliary Systems Sub-layer

Finally, the Physical layer has Auxiliary Systems sub-, illustrated in Figure 6.

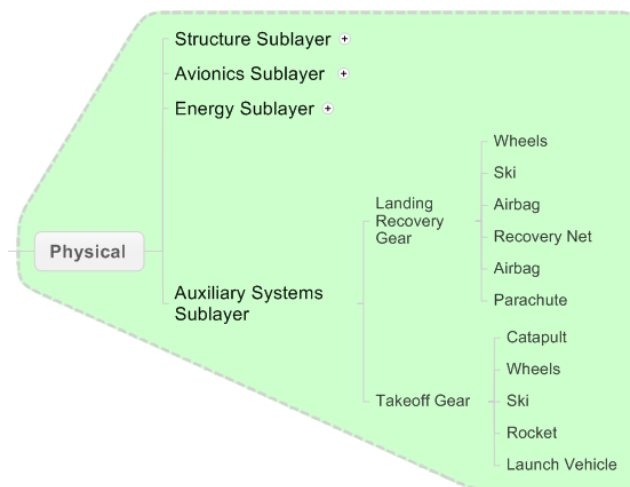


Figure 6: Auxiliary Systems Sub-layer.

This sub-layer is divided into Landing Mechanisms and Takeoff Mechanisms: the Landing Mechanism is divided

into wheels for landing fixed-wing aircraft on a ramp, landing skids, used by rotorcraft. For fixed-wing aircraft, when there is not the possibility of using a runway, it is used an Airbag System, Network Recovery or Parachute, depending on the characteristics of each aircraft; Takeoff Mechanisms ranging from conventional Wheels for fixed and rotary wing and Ski wings, even the unconventional when using fixed-wing aircraft and the inability to use a runway for takeoff, with a catapult, use of aids and Rockets Launch Vehicles.

B. Distributed RTOS

The distributed RTOS layer, illustrated in Figure 7, describes a set of API used by the real-time operating system embedded in the aircraft, used as input to or an output of the RTOS. In the Driver sub-layer are the hardware drivers APIs, and in the network sub-layer are the network APIs.



Figure 7: Distributed RTOS Layer.

C. System Abstraction

The system abstraction layer's function, illustrated in Figure 8, is defining a set of hardware for use in the upper layers. The IPC (Inter-Process Communication) sub-layer is responsible for the abstraction of communication among processes, and the I/O sub-layer controls the operation of input and output devices.

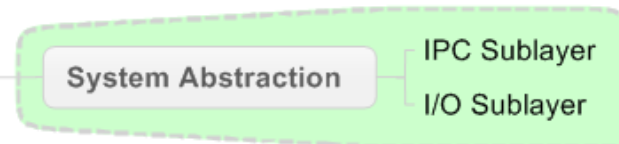


Figure 8: System Abstraction Layer.

D. Monitoring & Control

The Monitoring & Control Layer, illustrated in Figure 9, is responsible for monitoring the aircraft actions as well its control.

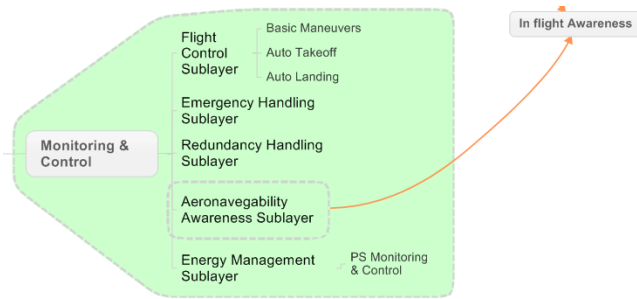


Figure 9: Monitoring & Control Layer.

It is divided into the Flight Control sub-layer, Emergency Handling sub-layer, Redundancy Handling sub-layer, Airworthiness Awareness sub-layer and Energy Management sub-layer. The Flight Control sub-layer responds to Basic Commands being executed by the aircraft, by the Automatic Take-off and also by the Automatic Landing. On the other hand, the Emergency Handling sub-layer is responsible for events that are not planned, such as battery consumption, making it impossible to accomplish the mission. The Redundancy Handling sub-layer manages duplicated subsystems in the aircraft that were installed to increase the reliance. The Airworthiness Awareness sub-layer is responsible for sensors and embedded detectors in aircraft, which purpose is to obtain information like those a human being has the capability to identify, such as smoke on board. The Energy Management sub-layer is responsible for the monitoring energy levels consumed by the aircraft.

E. Navigation & Service

Navigation & Services layer, illustrated in Figure 10, consists of Air Traffic Coordination, Flight Path Control, Geo Political Awareness, and server sub-layers. This layer is responsible for the aircraft’s navigation, sending signals that perform the required path to accomplish the mission. The Air Traffic Coordination sub-layer responds to traffic in the airspace in which the aircraft is operating. The Flight Trajectory Control sub-layer guides the aircraft navigation to achieve the Waypoints or the Grid coordinates defined by the mission. The Geo Political Awareness sub-layer is responsible for the virtual threshold that the aircraft must operate. The Server sub-layer contains non-priority services that help navigation and mission accomplishment. These services can be based on WWW (World Wide Web) or DTM (Data Transfer Mechanism).

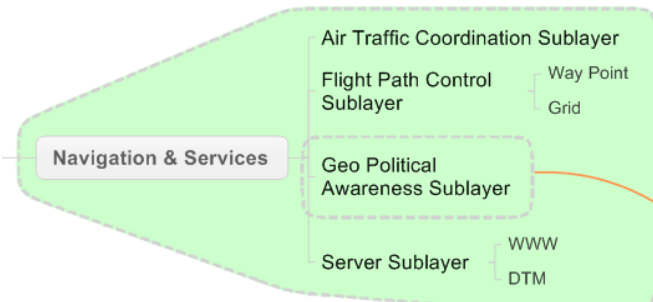


Figure 10: Navigation & Services Layer.

F. Mission

The Mission layer, illustrated in Figure 11, is divided into SSI (Smart Sensor Interface), Automatic Control, and Raw Control sub-layer.

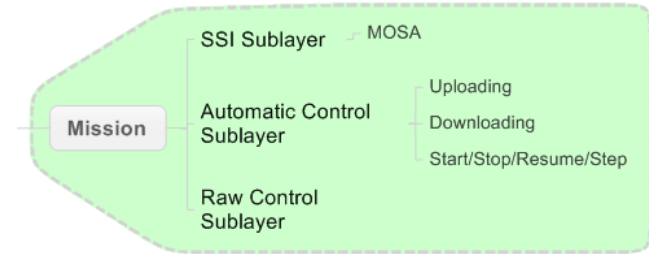


Figure 11: Mission Layer.

The SSI sub-layer [41] is responsible for accessing the MOSA (Mission Oriented Sensor Array) [41] and perform the entire checking, allowing discovering whether the aircraft met all the attributes to accomplish the defined mission. The Automatic Control sub-layer is responsible for accepting the mission data (Uploading), send the collected data (Downloading) and starting, stopping, resuming or performing part of the mission (Start/Stop/Resume/Step). The Raw Control sub-layer is simply responsible for sending data that does not need a proper treatment. With this layer, we ensure that data reaches pre-set destination.

This layer terminates the sequence of layers from the Aerial Segment. The Ground Segment’s layers are described in the next subsections.

G. Physical – Ground Segment

The ground segment’s Physical layer resembles, in some respects, the aerial segment’s Physical layer. The division into sub-layers is presented as follows: Electronics, Energy and Auxiliary Systems.

1. Electronics Sub-layer

The Electronics sub-layer, illustrated in Figure 12, consists of the Station Hardware basically computers with CPUs, Video Monitors, Keyboards, Mouse and Joystick. Communications sub-system is identical to this in aerial segment. The same concept is valid for the antenna sub-system. The sensor sub-system has only Weather Monitoring.

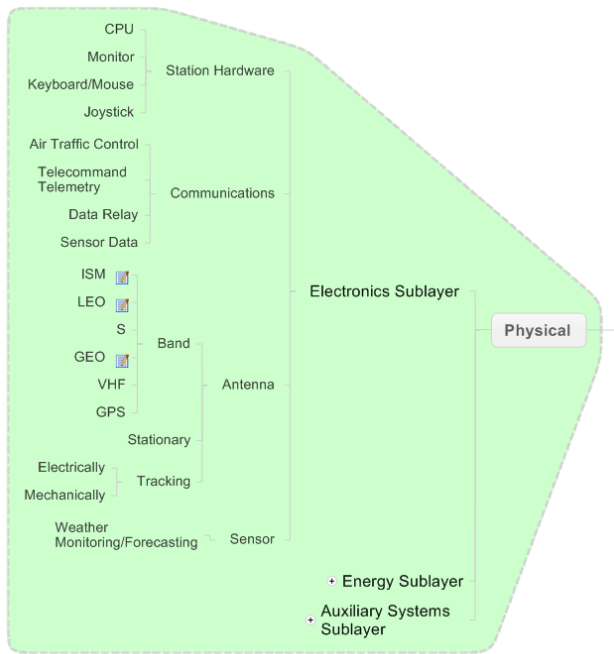


Figure 12: Electronics Sub-layer.

2. Energy Sub-layer

Energy sub-layer, illustrated in Figure 13, is responsible for providing energy for the ground segment, and has certain similarities with the same sub-layer in aerial segment, excluding the propulsion system and adding the batteries.

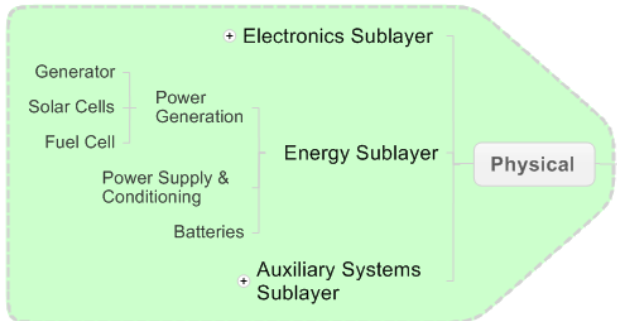


Figure 13: Energy Sub-layer.

3. Auxiliary Systems Sub-layer

The Auxiliary Systems sub-layer, illustrated in Figure 14, also similar to its analogue in the aerial segment, but it is simpler divided in Landing Mechanisms sub-systems where there is only the Recovery Network, and Takeoff Gear sub-system where there is only Catapult and Launch Vehicle. The Shelter sub-system is a place where it is mounted the ground control station's structure. Living Support responds by water supply and air conditioner, enabling working conditions for operators of UASs.

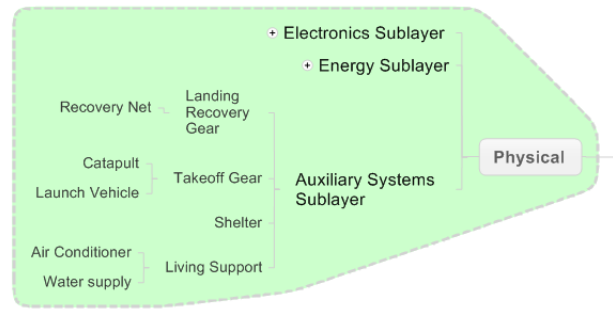


Figure 14: Auxiliary Systems Sub-layer.

H. Ground Station

The Ground Station Layer, illustrated in Figure 15, has Control & Monitoring, Moving Map, Payload Control and Video Conference sub-layers.

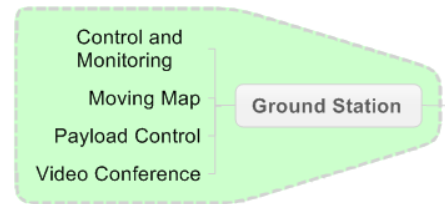


Figure 15: Ground Station Layer.

The Monitoring & Control sub-layer receives information in the form of aircraft telemetry and can also issue commands to guide the aircraft. The Moving Map sub-layer is responsible for the exchange of maps ability, submitting new maps to the aircraft when the initial mission is changed. The Payload Control sub-layer sends signals to aircraft in order to control the movement and operation of sensors, cameras and radars. The Video Conference sub-layer is responsible for exchange sound and image with other control stations.

In general this is a description of the proposed model. It is inherent in the completion of work in progress to define and describe in detail all the layers and sub-layers being proposed, as well as others that may emerge by the end of it.

IV CONCLUSIONS

UASs are complex systems that perform complex missions. Large UAVs are distributed in dozens of different processor systems. In these systems, there is the interconnection for the exchange of information among the different layers that compose a layered model based strictly delimits what is the function of each layer and what parameters it receives from the layer below and which parameters should yield to the layer above, thus facilitating the development of software. The Reference model architecture aims to standardize the various parts that make up a system. This kind of architecture brings benefits to systems like UAVs, primarily for being safety critical and complex systems. In this sense, LARISSA meets the existing needs.

This benefit may be realized when the development and use of SSI and MOSA, for example, because once developed they will use LARISSA following the reference model.

LARISSA can still become a reference for the creation of a UAS because the model will have all the specification of systems that make up a UAS, which enables a manufacturer X, for example, design and build the structure of the UAS, engage in this one mission control system developed by the manufacturer Y, developed by a sensor manufacturer Z, and so on. Another expected benefit is that the use of an architectural model can reduce the project cost because it enables already developed technology reuse, further expanding the use of UAS, either for military or civilian purposes.

The use of LARISSA architecture allows the definition of patterns and shapes, especially taking the compliance of these standards (in this specific case of organs such as the FAA). Layering allows reuse and meet requirements including certification and inclusion of these aircraft in a non-segregated airspace.

This work is ongoing and we intend to validate it using two techniques: Using questionnaires, which will be sent to researchers and manufacturers of UAS; and Make the implementation of one or more layers, i.e. specification, implementation and use in conjunction with the protocols and services; and Use case: applying the standard layer implemented in two or more low cost autopilots, such as Ardupilot, Slugs, OpenPilot and Paparazzi.

The authors acknowledge the support granted by CAPES, CNPq and FAPESP (process number 2012/08498-5).

REFERENCES

- [1] L. Lazic' and D. Velasevic', "Applying simulation and design of experiments to the embedded," *Software Testing, Verification & Reliability*, vol. 14, no. 4, pp. 257–282, 2004.
- [2] W.R. Dunn, "Designing safety-critical computer systems," *Computer*, vol. 36, no. 11, pp. 40–46, 2003.
- [3] A. Armoush, E. Beckschulze, and S. Kowalewski, "Safety assessment of design patterns for safety-critical embedded systems," in *SEAA '09: Proceedings of the 2009 35th Euromicro Conference on Software Engineering and Advanced Applications*, Washington, 2009, pp. 523-527.
- [4] S.P. Kumar, P. S. Ramaiah, and V. Khanaa, "Architectural patterns to design software safety based safety-critical systems," in *ICCCS '11: Proceedings of the 2011 International Conference on Communication, Computing & Security*, New York, 2011, pp. 620-623.
- [5] Z. Yi, W. Cai, and W. Yue, "Adaptive safety critical middleware for distributed and embedded safety critical system," in *NCM '08: Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management*, vol. 1, Washington, DC, 2008, pp. 162-166.
- [6] J. Kouloheris, "Future of System Level Design," in *panel discussion of First IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, 2003.
- [7] GAO, "Unmanned aircraft systems - federal actions needed to ensure safety and expand their potential uses within the national airspace system," GAO, GAO-08-511, 2008.
- [8] O. Trindade, L.O. Neris, L. Barbosa, and K.R.L.J.C. Branco, "A Layered Approach to Design Autopilots," in *IEEE-ICIT 2010 International Conference on Industrial Technology*, Viña del Mar, 2010, pp. 1395-1400.
- [9] K.R.L.J.C. Branco, J.M. Pelizzoni, L.O. Neris, O. Trindade Jr, F.S. Osório, and D.F. Wolf, "Tiriba a new approach of uav based on model driven development and multiprocessors," in *IEEE International Conference on Robotics and Automation - ICRA Communications*, Shanghai, 2011, pp. 6584–6587.
- [10] O. Trindade, K.R.L.J.C. Branco, L.O. Neris, and L.F.C. Chavier, "Robôs aéreos," in *Robótica Móvel*, 2012, pp. 461–478.
- [11] D. Rodrigues, R.M. Pires, J.C. Estrella, M. Vieira, M. Corrêa, J.B. Camargo, K.R.L.J.C. Branco and O. Trindade, "Application of soa in safety-critical embedded systems," *Communications in Computer and Information Science*, vol. 206, pp. 345–354, 2011a.
- [12] DOD, "OSD UAV Roadmap 2002-2027," U.S. Department of Defense. Office of the Secretary of Defense, Washington,DC, 2002.
- [13] DOD, "Unmanned Aircraft Systems Roadmap 2005-2030," U.S. Department of Defense. Office of the Secretary of Defense, Washington,DC, 2005.
- [14] DOD, "Unmanned Systems Roadmap 2007-2032," U.S. Department of Defense. Office of the Secretary of Defense, Washington,DC, 2007.
- [15] DOD, "Unmanned Systems Integrated Roadmap FY2009-2034," U.S. Department of Defense. Office of the Secretary of Defense, Washington,DC, 2009.
- [16] T. Grant, "Unifying planning and control using an ooda-based architecture," in *Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, Republic of South Africa: South African Institute for Computer Scientists and Information Technologists*, 2005, pp. 159–170.
- [17] L. Jun, Y. Zhiqiang, Q. Jianzhong, and L. Shukuan, "A role-based reference model for the service properties of service oriented architecture," in *IFITA '09: International Forum on Information Technology and Applications*, 2009, pp. 341-349.

- [18] W.C. Regli, I. Mayk, C.J. Dugna, J.B. Kopena, R.N. Lass, P.J. Modi, W.M. Mongan, J.K. Salvage and E.A.Sultani, "Development and specification of a reference model for agent-based systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 39, no. 5, pp. 572–596, 2009.
- [19] A. Mendes, *Arquitetura de Software: Desenvolvimento orientado para arquitetura.*: Campus, 2002.
- [20] N.S. Eickelmann and D.J. Richardson, "An evaluation of software test environment architectures," in *Proc. of the 18th International Conference on Software Engineering*, Berlin, Germany, 1996, pp. 353–364.
- [21] K.S. Barber, J. Holt, and G. Baker, "Performance evaluation of domain reference architectures," in *Proc. of the 14th International Conference on Software Engineering and Knowledge Engineering*, 2002, pp. 225–232.
- [22] D. Batory, L. Coglianese, M. Goodwin, and S. Shafer, "Creating reference architectures: an example from avionics," in *SSR '95: Proc. of the 1995 Symposium on Software reusability*, New York, NY, USA, 1995, pp. 27-37.
- [23] K. Sandkuhl and B. Messer, "Towards reference architectures for distributed groupware applications," in *8th Euromicro Workshop on Parallel and Distributed Processing*, Rhodes, Greece, 2000.
- [24] L. Bass, P. Clements, and R. Kazman, *Software architecture in practice. The SEI Series in Software Engineering*, 2nd ed.: Addison-Wesley Publishing Company, 2003.
- [25] L. Perrochon, "A reference architecture for multi-author world-wide web servers," in *COCS '95: Proc. of conference on Organizational computing systems*, New York, USA, 1995, pp. 197–205.
- [26] NIST, "4D/RCS: Reference Model Architecture for Unmanned Vehicle Systems Version 2.0," National Institute of Standards and Technologies, 2002.
- [27] E. Pastor, J. Lopez, and P. Royo, "UAV Payload and Mission Control Hardware / Software Architecture," *IEEE A&E Systems Magazine*, vol. 22, pp. 3-8, Junho 2007.
- [28] L. Olson and L. Burns, "A Common Architecture Prototype for ARMY Tactical and FCS UAVS," in *The 24th Digital Avionics Systems Conference. DASC 2005.*, Washington, 2005, pp. 8.B.5-1 - 8.B.5-11.
- [29] Z. Deng, C. Ma, and M. Zhu, "A Reconfigurable Flight Control System Architecture for Small Unmanned Aerial Vehicles," in *2012 IEEE International Systems Conference (SysCon)*, Vancouver, 2012, pp. 1-4.
- [30] J.M.M. Neto, R.A. da Paixao, L.R.L. Rodrigues, E.M. Moreira, J.C.J. dos Santos and P.F.F. Rosa, "A Surveillance Task for a UAV in a Natural Disaster Scenario," in *2012 IEEE International Symposium on Industrial Electronics (ISIE)*, Hangzhou, 2012, pp. 1516 - 1522.
- [31] P.J. Prisaznuk, "Integrated modular avionics," in *National Aerospace and Electronics Conference. NAECON 1992*, 1992, pp. 39-45.
- [32] C.B. Watkins and R. Walter, "Transitioning from federated avionics architectures to Integrated Modular Avionics," in *26th Digital Avionics Systems Conference. DASC '07*, Dallas, 2007, pp. 2.A.1-1 - 2.A.1-10.
- [33] Wind River Inc, "ARINC 653 An Avionics Standard for Safe, Partitioned Systems," in *IEEE-CS Seminar*, 2008.
- [34] M. Ilarslan, M.K. Bayrakceken, and A. Arisoy, "Avionics System Design of a Mini VTOL UAV," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 26, no. 10, pp. 35-40, 2011.
- [35] B. Kayayurt, I. Yayla, A. Yapici, and C. Kucukoguz, "Ground control station avionics software development in ANKA UAV," in *30th Digital Avionics Systems Conference (DASC)*, Seattle, 2011, pp. 5B6-1 - 5B6-7.
- [36] NATO/OTAN. (2007) NATO/OTAN - NORTH ATLANTIC TREATY ORGANIZATION. [Online]. <http://nsa.nato.int/nsa/nsdd/listpromulg.html>
- [37] S. Panicker. (2001) Scribd. [Online]. <http://pt.scribd.com/doc/46780104/Applying-DO178B>
- [38] H. Jin, S. Lee, S. Han, H. Jo, and D. Kim, "WiP Abstract: Challenges and Strategies for Exploiting Integrated Modular Avionics on Unmanned Aerial Vehicles," in *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, Beijing, 2012, p. 211.
- [39] A.S. Tanenbaum, *Redes de Computadores*, 5th ed. Rio de Janeiro, RJ, Brasil: Campus, 1997.
- [40] K. P. Valavanis, "Advances In Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy," *International Series on Intelligent Systems, Control, And Automation: Science And Engineering*, vol. 33, 2007.
- [41] R.M. Pires, D. Rodrigues, K.R.L.J.C. Branco, and O. Trindade, "Mosa - mission oriented sensor array: A proposal," in *CLEI '11: Proceedings of the XXXVII Conferencia Latinoamericana de Informática*, 2011, pp. 1309-1318.