SYSTEM REQUIREMENTS FOR THE SUPERVISORY CONTROL OF A FUSELAGE ALIGNMENT ROBOTIC CELL

Glauber Mosqueira, glauberm@ita.br
Leopoldo Rideki Yoshioka, lryoshioka@gmail.com
Osamu Saotome, osaotome@ita.br
Luís Gonzaga Trabasso, gonzaga@ita.br
Instituto Tecnológico de Aeronáutica (ITA)
Aircraft Structure Assembly Automation Laboratory (ASAA Lab)
Praça Marechal Eduardo Gomes, 50 - Vila das Acácias - São José dos Campos
12.228-900 - São Paulo, Brazil

Abstract. *The ever-growing need to improve manufacturing processes has recently led to an increase in the number of automation solutions employed to assemble aircraft structural elements. One assembly process of particular interest to the Brazilian aircraft industry is the automation of the alignment and fastening of fuselage sections. This procedure requires a manufacture robotic cell comprised of many elements, ranging from industrial manipulators to large volume metrology systems. In order to synchronize the operation of all these components, the cell must have a centralized control system, namely a supervisory system. This work presents a functional analysis for the requirements of such a system. Given the complexity of the supervisory control, this analysis is crucial to plan out the development work before implementation and to assure all requirements are met. The paper presents functions mapped into five categories: integration, configuration, monitoring, control and analysis. The integration functions contemplate network requirements to physically integrate the cell equipments. Configuration functions allow for the customization of the supervisory system by the user. Monitoring and control functions manage the interactions with the equipment, while analysis functions generate the operation history and alike. A function tree is designed and then the tree items are detailed in accordance to the European Cooperation for Space Standardization ECSS-E-ST-40C software specification standard.*

Keywords*: Functional analysis, supervisory control, computer networks, fuselage alignment*

1. INTRODUCTION

The ever-growing need to improve manufacturing processes has led to an increase, in recent years, in the number of automation solutions employed to assemble aircraft structural elements (Sahr *et al.*, 2009). The work described herein is part of the activities being carried at the Aircraft Structure Assembly Automation Laboratory (ASAA Lab). An outcome of a partnership between the Instituto Tecnológico de Aeronáutica (ITA) and the Brazilian aerospace sector, the ASAA Lab is developing an automated assembly process for aircraft fuselages. This work is a first step towards the modernization of the aerospace manufacturing techniques in Brazil, aiming at the better product quality and faster production times brought by automation. Given the current development pace of this area for the aerospace industry in general, initiatives like the ASAA Lab are crucial to keep the Brazilian companies competitive. They represent an investment not only in the short term fabrication capacities, but also in a long term know-how accumulation and human resources training in automation processes.

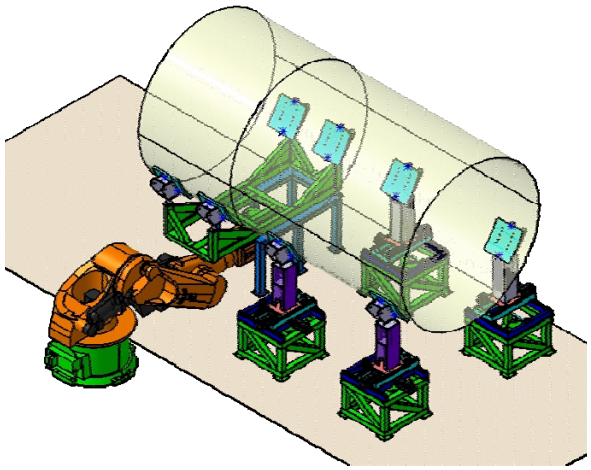

Figure 1. Computer simulation of the ASAA robotic cell

The robotic cell at the ASAA lab is comprised of several modules (as partially shown in Fig.1), including two industrial manipulators (KUKA™'s KR-210 and KR-500), three metrology systems (Nikon™'s iGPS, Laser Radar and K-610), a 12DOF non-conventional robot (NIVA) and two custom made end-effectors (EFIP and EMC). The development of each module has been carried independently, and the subsequent step in the project is the creation of a supervisory controller to integrate all subsystems. This article presents a functional analysis detailed into a requirements specification for the Supevisory Control System of the ASAA cell. Given the complexity of this controller, the analysis is essential to guide the development of the implementation phase, saving coding/debugging time and ensuring all desirable features are included in the system. The paper is organized as follows: section 2 describes the ASAA cell from network and operation standpoints; section 3 presents the functional analysis and finally section 4 details the analysis into a requirements specification.

2. THE ASAA CELL

2.1. Network considerations

Modern manufacture cells are comprised of at least a handful of modules likely supplied by diverse manufacturers. In this case, integration becomes more than a plug & play issue, entering the realm of complex computer networks where different interfaces and protocols must be brought together. At the ASAA cell, all modules are physically connected through an IEEE 802.3 (Ethernet) network and although some equipment can intercommunicate, an overall communication controller is yet to be conceived. As shown in the following sections, this controller constitutes the foundation of the supervisory system. At the logical level, different communication protocols are used according to manufacturer and/or system constrains. In order to help the reader gain some quick understanding about the problem discussed in this paper, the block diagram shown in Fig. 2 illustrates the logical network topology of the ASAA cell.

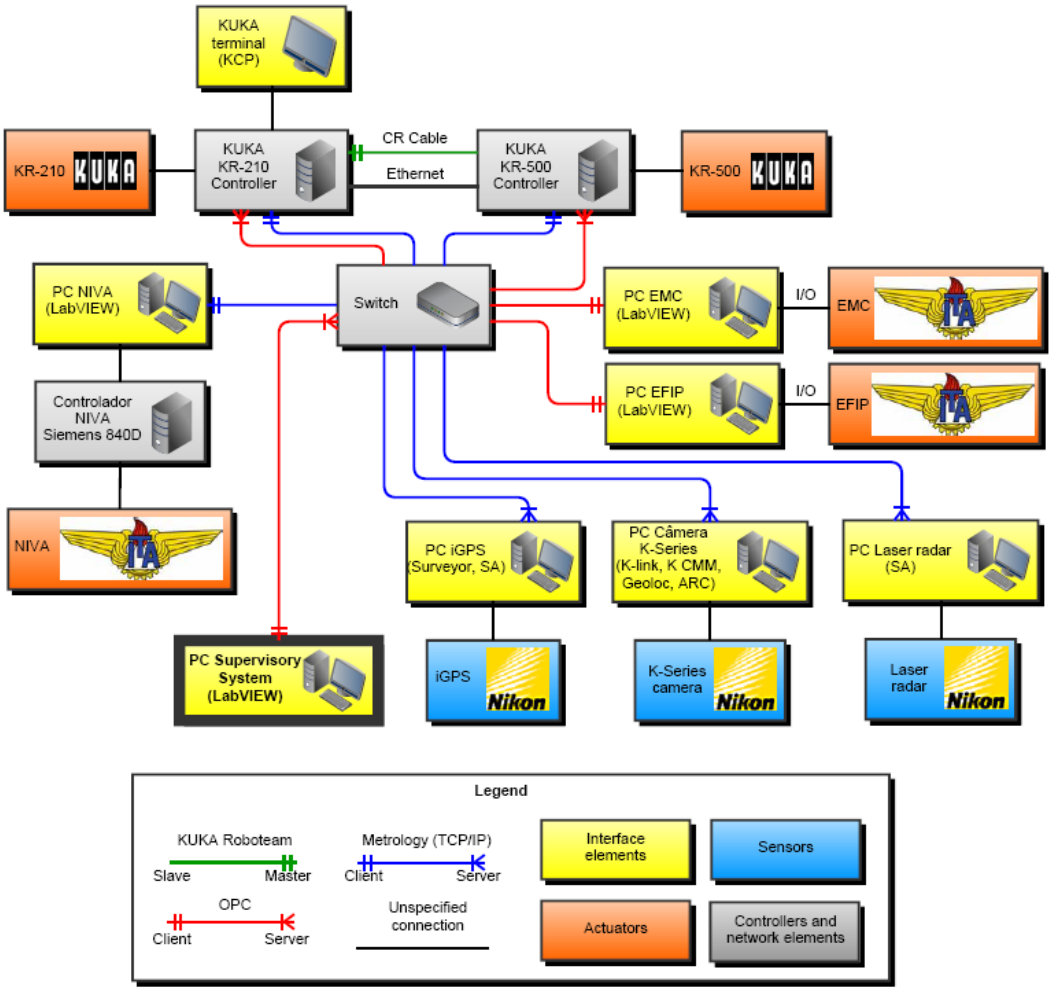

Figure 2. Logical network topology of the ASAA robotic cell

The network contains several communication protocols and software suites. For the sake of convenience, LabVIEW™ was predefined as the standard platform for the ASAA cell supervisory system. As such, all custom-made interfaces are being developed in this environment, as it is the case for the EFIP, EMC, NIVA and supervisory system terminals. The KUKA terminal communicates factory-standard through OPC (OLE for Process Control), which is supported by LabVIEW™. OPC is a series of standards specifications conceived to fill industry-specific needs (OPC Foundation, 2011), bridging the gaps between different manufacturers. At last, all metrology terminals and programs communicate through TCP, which is also supported by LabVIEW™. Other intermediate protocols are employed to connect actuators and sensors to the respective terminals, but from a supervisory perspective anything behind a terminal is not relevant.

2.2. Operation sequence

Definition of the cell operation sequence must also be considered as a starting point for the development of the supervisory system, to ensure its software dataflow mimics reality. This allows for adequate (realistic) monitoring and control. That means the software should be modeled with the time progression of the operation taken into account, although in this case strict real-time features will not be of concern. Fig. 3 presents the operation sequence of the ASAA cell, where the equipments involved in each operation are represented in the dashed boxes.

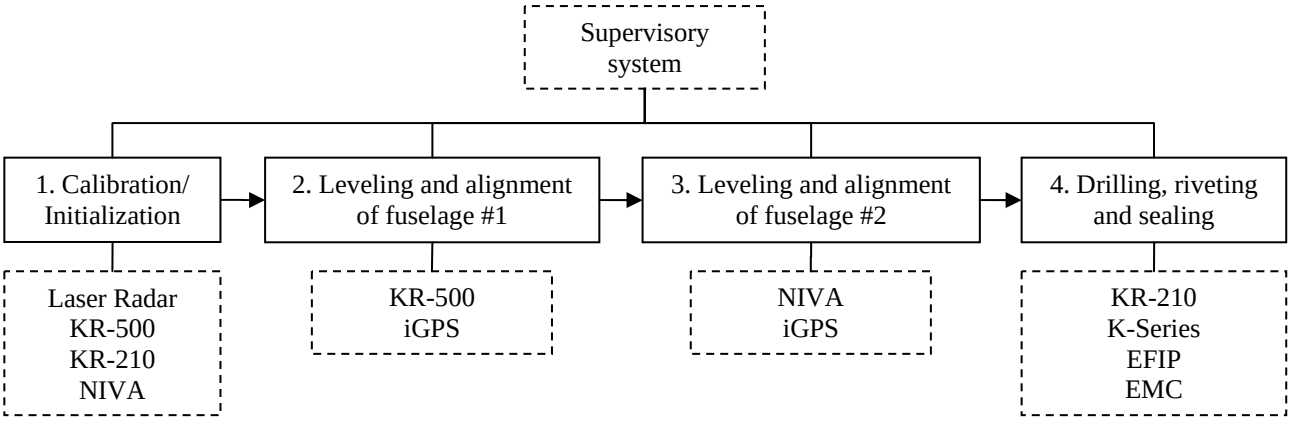

Figure 3. Operation stages of the ASAA cell and related equipment

Stages were divided according to a set of operations involving a fixed subgroup of devices. A fifth stage, inspection, has also been considered but at this point development work has yet to be started towards this end. Inspection functionality will be specified as an optional requirement for the supervisory system.

2.3. Service requests

The functionality of the supervisory system should not overlap that of the individual equipment, complementing rather than replacing each device's low-level commands. This architecture preserves the safety and performance features originally implemented by the manufacturer and avoids unnecessary duplicated calculations in the cell. The concept of service requests (as illustrated in Fig. 4) was developed to address this principle. A service consists of a valued action provided by an equipment, such as a measurement (for metrologic sensors) or motion (for robots). The supervisory system can request a device to start, interrupt or finish a service. The service itself is handled entirely by the dedicated controller of the equipment as proposed in its intended use. Whenever the equipment receives a request, it returns a status message to keep the supervisory control in the loop.

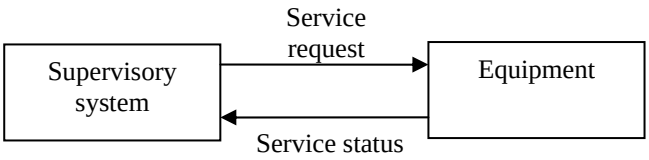

Figure 4. Service request concept

3. FUNCTIONAL ANALYSIS

Information gathering is a crucial phase in systems development. In the Value Analysis and Systems Engineering context, this information focuses on the function required by a design, process or service to accomplish its objective or

mission, striving to develop alternatives designed to achieve only the required functions at the lowest cost while meeting the fundamental requirements (Wixson, 1999). Functions are the intended effect of a system, and functional analysis is a technique used to map all functions of a system (ECSS, 1999). This analysis outlines the required functions in a systematic and useful manner, increasing the success rates of complex designs. The functional analysis considers a system in terms of functions rather than solutions, speeding up the mapping process and leaving an eventual solution discussion to the architecture design phase. Moreover, considerations about the reliability or safety of functions are not in its scope. Functions are enlisted top-down in increasing levels of specificity, with the top level derived directly from the system mission statement. Functions should not be passed from one system to another in the same level, but should be received from a higher level system and either serviced or further distributed to lower level systems.

Bartolomei *et al.* (2001) describes the Functional Analysis Systems Technique (FAST), an implementation of the functional analysis concept, as a rigorous method for understanding complex systems. The functions are mapped in a how-why arrangement (Fig. 5), in which lower functions must answer how higher functions are carried out whereas higher functions must answer why lower functions are there. Each function is described in a "verb + noun" structure.

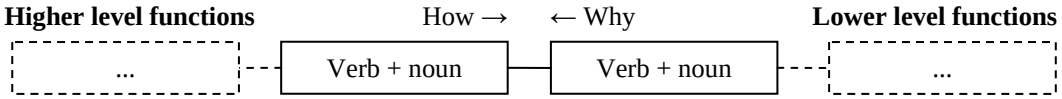

Figure 5. The FAST how-why arrangement

The following diagrams present a functional analysis of the ASAA supervisory system. Each of the five branches derives from the "Supervise ASAA cell" topmost function. The dashed boxes represent a solution rather than a function and are a transition to an architecture design, occupying the lowest levels of a functional analysis. Fig. 6 (top) presents the mapped integration functions of the ASAA cell, constituting the network foundation for the implementation of the supervisory system. Fig. 6 (bottom) presents the configuration branch. These functions allow the user to customize the software, and support industry-relevant safety features such as varying levels of access profiles.

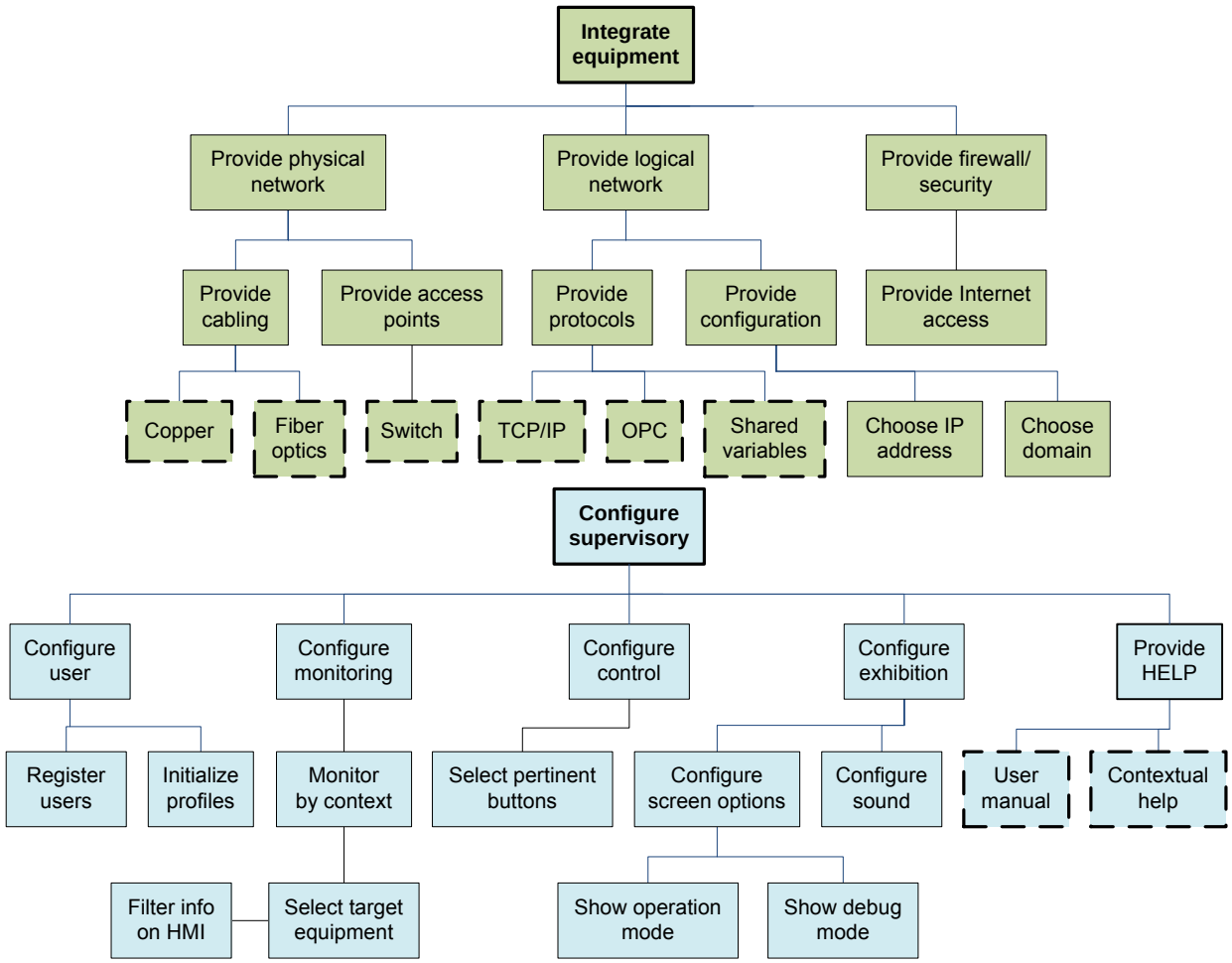

Figure 6. Function tree, integration (top) and configuration (bottom) branches

The monitoring branch (Fig. 7) contains the actual supervisory functions. The alarm monitoring section distinguishes between alarms generated at individual equipment and supervisory alarms. This is important to ensure the supervisory system is able to replicate external alarms besides generating its own. It is worth pointing that the operation monitoring section inherently resembles the operation sequence shown in section 2.2, as one would expect. The equipment monitoring section also follows this pattern, but with more specific functions. Finally, the environment monitoring section support industry-relevant functions that account for the presence of workers around the cell and provide visual capabilities for added safety, aiding in debug, accident and failure investigation.

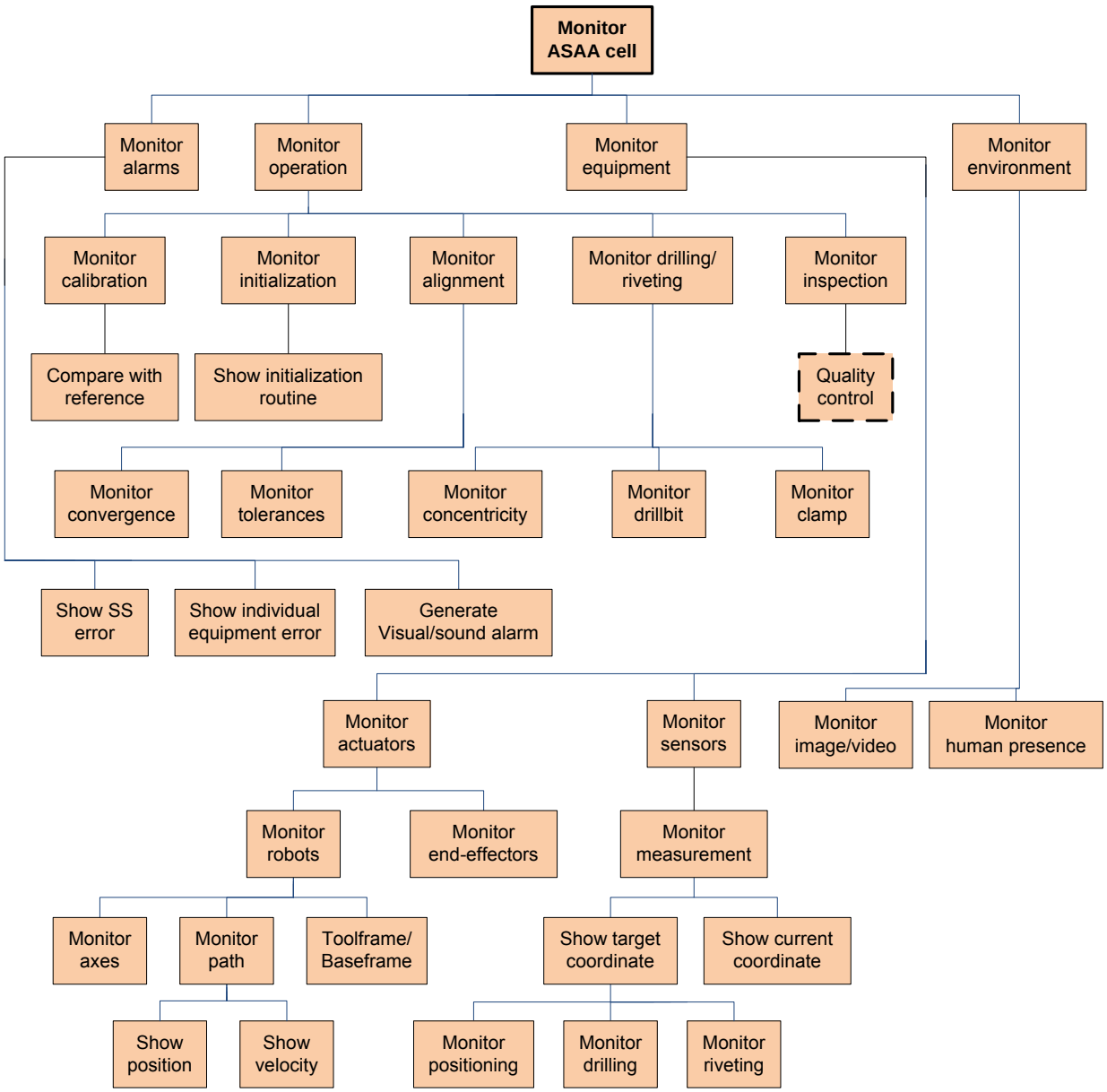

Figure 7. Function tree, monitoring branch

The control branch (Fig. 8, top) is coupled with its monitoring counterpart and together they concentrate the main functions of the supervisory system. Its service sequencing section also features functions based on the operation sequence of the ASAA cell. The service commanding functions establish a reliable communication (such as TCP) with the peripherals to ensure the supervisory system receives an ACK or NACK for every order it sends and therefore stays aware of the status of each single order – whether it was received, not received or denied.

Finally, the analysis branch (Fig. 8, bottom) contains yet another group of support functions that enable the user to gather data about the system. With this info, the operator will be able to calculate routine operation statistics and generate regular reports as well as debug and document eventual failures.

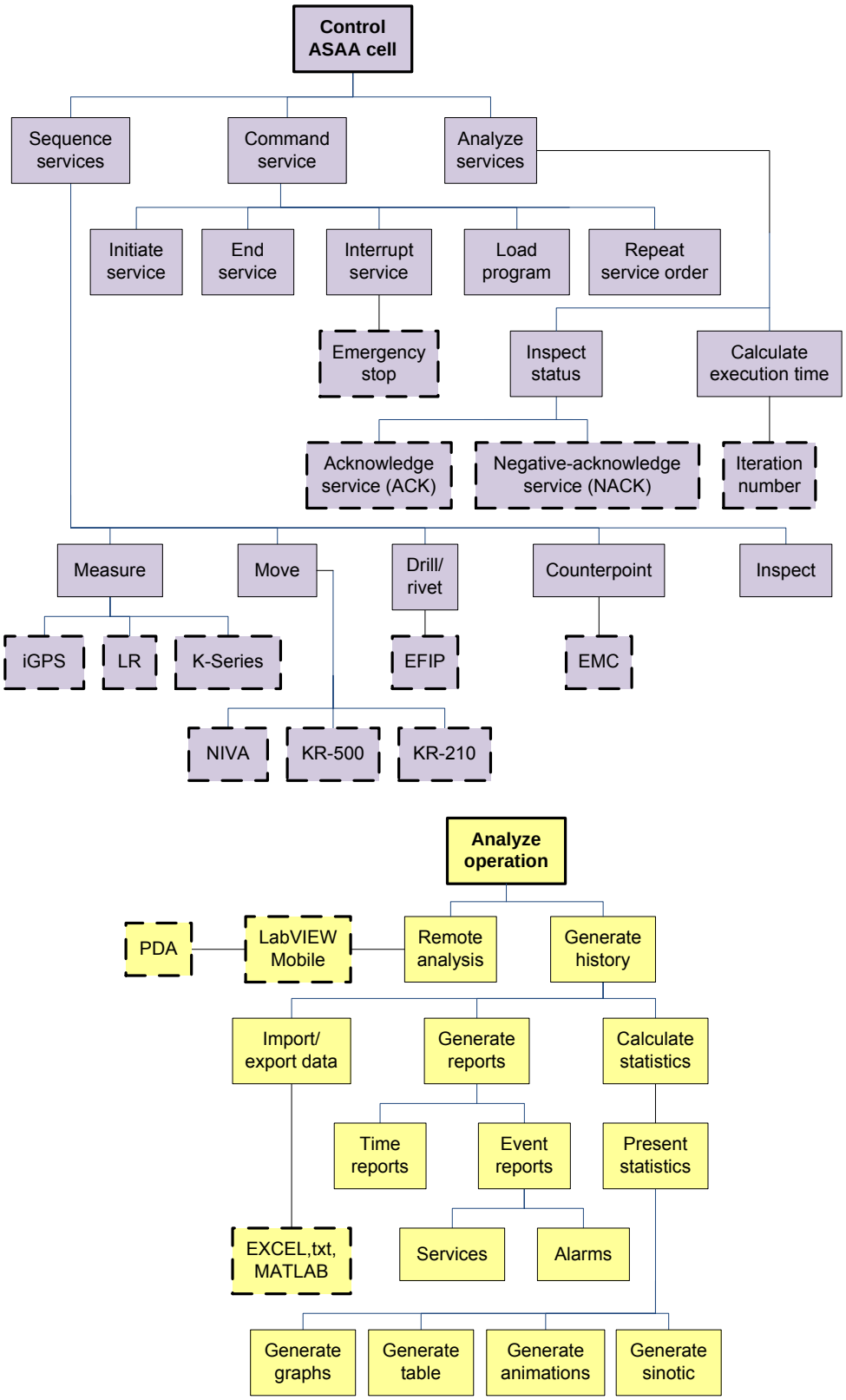

Figure 8. Function tree, control (top) and analysis/post-processing (bottom) branches

4. FUNCTIONAL REQUIREMENTS SPECIFICATION

ECSS (2009) proposes a standard that defines the principles and requirements applicable to space software engineering. It includes all aspects of space software engineering including requirements definition, design, production, verification and validation, transfer, operations and maintenance. It is suitable for both the airborne and ground

segments. As such, it was used as a guide for the requisites specification of the ASAA cell supervisory system. This section presents a summary of such requirements.

4.1. Integration requirements

Physical network – all equipment contained in the ASAA cell should comply with the IEEE 802.3ab Gigabit Ethernet standard (IEEE, 1999). This is so because all current devices are compatible with this standard and Gigabit Ethernet has been the most commonly found network interface for computers since the past decade. The standard specifies copper cabling Cat-5 or Cat-6, of up to 100m in length, and therefore adequate to the ASAA cell. The network must also have an IEEE 802.3ab compatible switch with at least 32 ports to provide access points for all the cell devices.

Logical network – the supervisory system should support the following communication protocols: OPC, TCP and LabVIEW™ Shared Variables. These cover the full communication spectrum of the ASAA cell. It must also provide standard network configuration functions such as IP address assignment.

External access – Internet access has been discussed as an optional requirement as a means to provide the cell with software update and data export functionality. However, external access might cause security issues since the ASAA cell will operate in an industrial environment. The feasibility of this item must still be assessed.

4.2. Configuration requirements

User configuration – this function should allow user registering in different access levels. As an example, in the <Operator> level users can only run a routine without modifying it. In the <Expert> level, the user can run and edit routines, but has no access to the supervisory system source code. Finally, in the <Administrator> level the user has access to the source code, being able to modify the supervisory system software. Registration must include information such as user name and company position, and must be approved by an Administrator. User preferences (where applicable) are saved with the registered profile. Whenever the user logs into the system, the system stores the login/logout timestamp, and a timeout function must automatically logout the user after a predefined inactivity period.

Monitoring and control configuration – this function should feature contextual monitoring and control capabilities. It means the user will be able to select the pertinent monitoring and control screen buttons/graphs according to context. For added safety, these buttons/graphs will be divided into permanent and transient, being the transient the only ones subject to user selection. Furthermore, the user will be able to adjust the scale and increment associated to each monitoring and control button/graph.

Exhibition and Help – the user should be able to select the most adequate exhibition mode (as allowed by its access level). Possible modes are “Debug” and “Normal operation”. The help function must provide complete system documentation and a LabVIEW™ supported tool, contextual help, should aid users with a simple and focused, in-screen follow-the-cursor help window.

4.3. Monitoring requirements

Operation monitoring – this function should support the monitoring of the ASAA cell in terms of its operation stages (section 2.2). During the calibration stage, it should monitor the creation of reference data and equipment setup. During initialization, it must monitor a self-test initialization routine that shows the user whether the cell is working properly. According to the self-test results, the user/system may or may not allow the cell to start the next operation stage. During alignment and leveling, it must monitor the number of iterations and convergence time taken by the closed-loop process and the tolerances achieved. During drilling and riveting, it must monitor parameters such as hole concentricity and clamp force. Because the control software of both EFIP and EMC was written in LabVIEW™, the target parameters for the supervisory system will likely be a simplification of the EFIP/EMC control software parameters. Finally, during the optional inspection phase this function must check whether the product fabrication parameters, such as hole quality or alignment tolerance, were met.

Equipment monitoring – this function should monitor basic actuator data such as end-effector coordinate, joint displacement, and active base and tool frames. It must also monitor basic sensor data such as current coordinates and target coordinates.

Alarm monitoring – this function should show pertinent warnings/alarms to the user when failure and/or exception states are encountered by the supervisory system. It must establish which actions should be taken for each error. At this point, it is necessary to distinguish between automated actions and actions that should wait for user input. This function should also allow that external alarms (those generated by equipment and/or subsystems other than the supervisory system) are transmitted to the supervisory screen. For this function, it is crucial to analyze the time taken by the supervisory system to reproduce an external alarm and respond to it.

Ambient monitoring – this function should monitor the environment around the ASAA cell. It must contain at least two sensors: firstly, an optical barrier that detects the movement of workers around the cell; secondly, a video camera

system that records the cell operation fulltime. The visual info might be used together with other logs as a source to detect problems, improve performance and investigate accidents and/or failures.

4.4. Control requisites

Sequencing and service control – this function should be able to sequence the execution of the service requests. For example, the drilling service must be done prior to the riveting one. It is worth pointing that these are high-level services that may include subservices. For example, the drilling service includes a hole concentricity check by a camera, all of this handled by the EFIP software. Given its scope, the supervisory system must operate at the highest control level possible, leaving calculations and low-level commands for the dedicated equipment controllers. There are five service types: measurement (for sensors), movement (for actuators), drilling/riveting (for end-effectors) and the optional inspecting service.

Service ordering – this function should include the main service requests: initiate service, terminate service, interrupt service, load program and repeat service request.

Service analysis – for each service request made by the supervisory system, the equipment to which the order was given should return an ACK or a NACK. In case a NACK is returned, the failure must be handled accordingly. In case an ACK is returned, the equipment must start to transmit the service request until its completion. For example, when a movement request is sent to a robot, it must ACK the request and then continuously report the "moving" status until it has reached its final destination. However, in case there is an error during the execution of the service request, the error messages and pertinent alarms must be shown instead of the "moving" status.

4.5. Analysis requisites

Operation history – all pertinent operation data should be saved in a log file, together with a timestamp of the event occurrence. Examples of pertinent data are user login/logout, alarms and begin/end of services. The log file might be presented in two versions: a complete one, presenting all saved data for investigation in case of a failure, and a summarized one with normal operation history. The log file might also be used for the automated creation of reports concerning the system's health, cell performance or final product quality.

Operation statistics – this function should be able to show operation statistics to the user. Its goal is to present data in a simple and intuitive way so that the user can rapidly grasp the current situation of the supervisory system. The statistics might be presented in graphs, tables, animations or synoptic. Statistics and log data must also be available to export in common formats.

Remote analysis – this function should support remote analysis in order to provide for situations where remote monitoring and control prove useful, such as PDA based messages for assembly line supervisors.

5. CONCLUSION

The design of a supervisory system for a manufacture cell is a complex task. Its development must involve careful planning in order to facilitate implementation avoiding a wasteful employment of time and resources. This paper presented the FAST diagram technique and ECSS-E-ST-40C software requirement specification standard as systematic methods to guide the information gathering phase of complex systems design. The techniques were applied to the development of a supervisory system for the Automated Assembly of Aircraft Structures, a project from the Brazilian aircraft industry aimed at the automation of aircraft fuselage assembly.

The analysis results were presented in a function tree with five branches: integration, configuration, monitoring, control and analysis. The tree was decomposed in several sub-levels and maps all desirable functions the supervisory system must present. These functions were then converted into a requirements specification that, in the following steps of the development, will guide the architecture design, detailed design and software codification and tests

Besides its technical application value, the case study presented may be used for didactic purposes covering the techniques used and as an example to other projects with similar complexity. The step-by-step description detailed in this paper might prove useful in the reconstruction of the creative process reported.

6. ACKNOWLEDGEMENTS

The authors would like to thank Financiadora de Estudos e Projetos (FINEP) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for their financial support, and Empresa Brasileira de Aeronáutica S/A (Embraer) for the support received for this project on visits to their premises.

7. REFERENCES

Bartolomei, J. E., Miller, 2001. "Functional Analysis Systems Technique (F.A.S.T.) as a Group Knowledge Elicitation Method for Model Building", 19th International Conference of the System Dynamics Society, Atlanta, USA.

ECSS – European Cooperation for Space Standardization, 1999. "Space engineering – Functional analysis", ECSS-E-10-05A Standard.

ECSS – European Cooperation for Space Standardization, 2009. "Space engineering – Software", ECSS-E-ST-40C Standard. pp. 95-101.

IEEE – Institute of Electrical and Electronics Engineers, 1999. "Gigabit Ethernet", IEEE 802.3ab Standard.

OPC Foundation, 2011. "What is OPC?". 28 Jan. 2011, <http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp>.

Sarh, B., Buttrick, J., Munk, C., Bossi, R., 2009. "Aircraft Manufacturing and Assembly". Springer Handbook of Automation, pp. 893-910. Springer, Heidelberg.

Wixson, J. R., 1999. "Function Analysis and Decomposition using Function Analysis Systems Technique", International Council on Systems Engineering Annual Conference (INCOSE '99), Brighton, England.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.