FORMULATION AND IMPLEMENTATION OF AN AIRCRAFT – SYSTEM – SUBSYSTEM INTERRELATIONSHIP MODEL FOR TECHNOLOGY EVALUATION

Prof. Mavris D. N.*, Phan L. L., Dr. Garcia E.*****
***Director, Aerospace Systems Design Laboratory (ASDL), Georgia Institute of Technology**
****Graduate research assistant, ASDL ***Research engineer, ASDL**

Keywords: *technology integration, subsystem interactions*

Abstract

The integration of a new technology at the subsystem level can have impacts on the system-level parameters that are difficult to assess. The physical architecture of a commercial aircraft and its subsystems is established in order to develop an object-oriented framework for the evaluation of new subsystem technologies.

This framework enables the designers to model the subsystems and their interactions, assemble them and perform design studies in an interactive integrated environment.

1 Introduction

The design of aircraft subsystems is primarily driven by the high-level requirements the aircraft imposes. However, as subsystems grow in complexity, their integration has become a critical issue in aircraft design, as they can in turn impose constraints on other subsystems as well as on the overall system – the aircraft. For example, after a subsystem is designed, the aircraft designer might realize that there is not enough volume to fit the subsystem and that the subsystems layout might have to be redefined. These ‘feedback’ constraints, depicted in Figure 1, are essential to understand as they can impact the overall performance of the system. For instance, pneumatic subsystems require bleed air from the engine high pressure compressor, inducing an increase in engine volume and weight in order to continue providing the required thrust.

Further, a good comprehension of these feedback and cross-subsystem constraints can help prevent costly changes late in the design process and provide guidance in the evaluation of architecture alternatives. The choice of an architecture with electrically-powered actuators, for example, has consequences on the hydraulic subsystem and the level of redundancy required. Also, electric actuators may be heavier than hydraulic ones but may still enable a considerable reduction in aircraft weight as the need for hydraulic lines is suppressed.

This kind of architecture tradeoff requires a model that captures all the relevant interactions between subsystems. This can also be helpful in the case of technology additions. Thus, it is essential, when a change is made on a subsystem through technology insertion, to comprehend the extent to which the system and the other subsystems are affected. While system engineers normally understand well how much improvement a technology may bring to a subsystem, they may find it difficult to foresee how the change will propagate to other subsystems and what further modifications will be required.

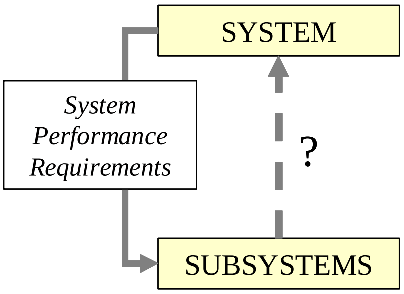


Figure 1: Subsystem feedback

2 Aircraft Physical Architecture

The study of the relationship between the aircraft and its subsystems constitutes the core of the discipline called 'systems engineering' (SE). In the SE process three interconnected activities coexist: requirements analysis, functional analysis and design synthesis [1]. A premise of SE is that the overall system (here the aircraft) should be regarded in a holistic manner. Indeed, the aircraft, an assembly of interdependent subsystems that are designed to perform a particular function [2], should be considered as a whole. Each subsystem only exists to serve the aircraft and hence should not be treated individually, but within its environment.

Thus one of the first activities in architecture design consists of decomposing the aircraft into its main subsystems and establishing the underlying hierarchy. The physical breakdown of the system can be visualized in a block diagram called 'specification tree' [3]. There are numerous ways to decompose a system into smaller subsystems. For this study, the specification tree was created so that it could relate to the index published by the Air Transport Association (ATA), the ATA Specification 100 [4], already widely accepted within the aerospace community. ATA Specification 100 divides the aircraft subsystems into numbered chapters. The index of the ATA chapters included in the aircraft physical breakdown, which will be designated by the term "subsystems", is given in the appendix. In order to take into account the functional breakdown of the aircraft, the ATA Chapters were grouped into intermediate modules called "segments", as can be seen in Figure 2.

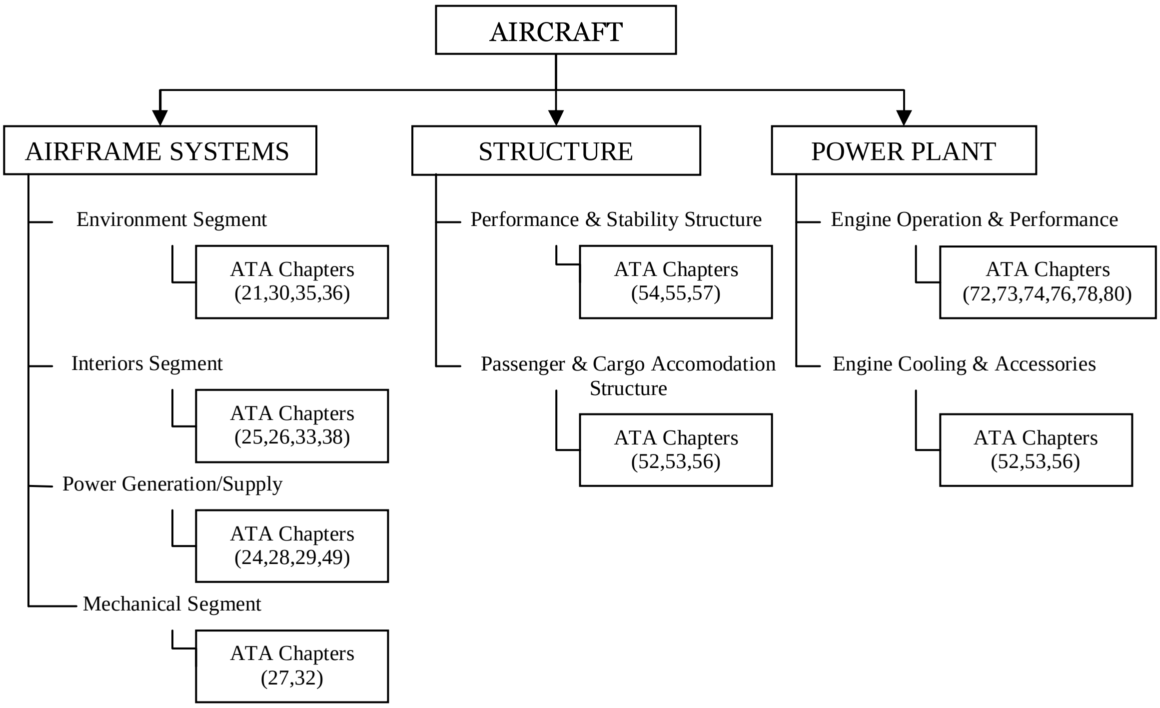


Figure 2: Aircraft physical breakdown: specification tree

3 Aircraft Subsystems and Interfaces

A subsystem (ATA chapter) is described by several properties. Firstly, it performs one or several functions that contribute to the realization of the higher-level function of the segment to which the subsystem belongs. For example, the functions of the pneumatic subsystem (ATA 36) are to provide and manage the distribution of pneumatic power.

Among the attributes that define the state of a subsystem, a distinction can be made between static attributes and the ones that will vary through the mission. The static attributes, or *characteristic parameters*, are the outcome of the design of the subsystem and are sufficient to describe the subsystem taken out of its operating environment. They include quantities such as weight, volume, or maximum electrical power output. The attributes of the subsystem that will vary during operation are the *variable parameters*.

The subsystems communicate with each other by means of interfaces. An interface is a common boundary between two objects. Following the nomenclature used by the Systems Modeling Language (SysML) [5], which is a modeling language tailored for systems engineering, the parameter associated to the information conveyed by an interface is called a *flow*. From the subsystem perspective, a flow can either be an input, in which case the subsystem is a client, or an output, in which case the subsystem is a supplier. This is illustrated in Figure 3, where subsystem A (for example ATA 24 - Electrical power) is the supplier of the flow (AC current) to the client B (ATA 27 - Flight controls). A port is the interaction point of a subsystem through which information flows to/from another subsystem.

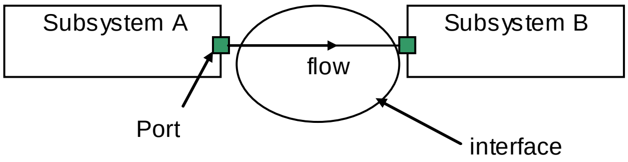


Figure 3: Subsystem interface

It is worth noting that the labels ‘supplier’ and ‘client’ relate to the considered flow. Thus, a subsystem may be a supplier of a flow as well as a client of another flow. Because a flow is by definition a quantity related to the communication between two subsystems during the operation of the aircraft, it is a type of variable parameter. The variable parameters that do not fall into the flow category are called *operational parameters*, as they describe the state of the subsystem during operation.

Identifying and documenting all the interfaces within the overall system is essential for the success of a design project. However, for complex systems like aircraft, it may become necessary to simplify the model used in the first phases of the design process by considering only the most significant interfaces [6]. These can be divided into six main categories:

- Electrical power
- Hydraulic power
- Pneumatic power
- Mechanical
- Heat
- Data and command

4 Requirements and Constraints

As stated in section 2, one of the fundamental activities of systems engineering is the analysis of requirements. The whole set of requirements that apply to a system is sometimes considered as one type of system architecture. At the top of this architecture lie the customer requirements, which define important aspects of the design such as the mission, the payload or range of operation. Starting with these top-level requirements, the design process goes through a series of design phases from conceptual to detailed design (cf. Figure 4), each of which results in a set of more specific requirements and specifications that the next design phase should fulfill [7]. Eventually, the top-level requirements are broken down into requirements that the subsystems and the interfaces must meet.

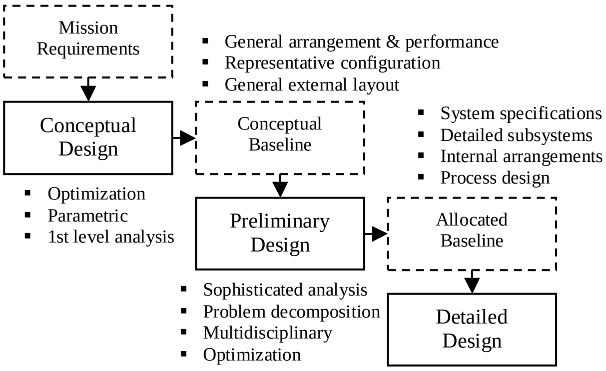


Figure 4: Design process

Subsystem requirements generally concern the subsystem characteristic parameters, while the interface requirements are associated to the operating conditions of the system. For example, the flight control subsystem may impose an interface requirement on the amount of electrical power that the electrical power subsystem needs to supply. The subsystem imposing the requirement is said to be the *master* whether that subsystem is the supplier or the client for the flow. Conversely, the other subsystem in that flow, meeting the requirement imposed, is called the *servant*.

5 Object-oriented Modeling and Simulation Framework

With the increasing computing capabilities offered to the designer, it is now possible to model the design with better fidelity. By nature, object-oriented languages enable the designer to create a virtual model of the system that closely matches the reality of the physical architecture of the vehicle [8]. Further, the modular aspect of object-oriented programming suits the growing need for a distributed modeling and simulation environment where parts can be defined offline by third-tier designers and then integrated in a common platform for analysis, as illustrated in Figure 5.

5.1 Class Structure

The transposition of the physical architecture of the aircraft into an object-oriented model is straight forward: each block of the physical breakdown is modeled by a class, and the aircraft class is the aggregation of the subsystem classes. Because the latter share some structural properties, they derive from a common parent class, the class “Subsystem”, as represented in Figure 6.

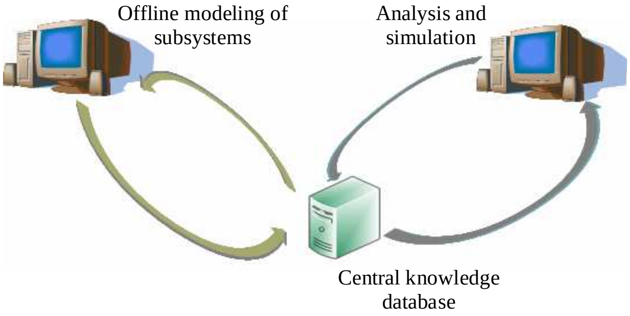


Figure 5: Distributed design environment

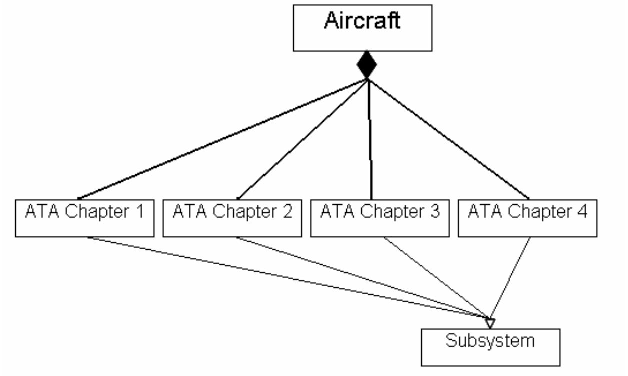


Figure 6: Top-level class diagram, simplified

Each subsystem class defines string attributes for the functions that the corresponding ATA chapter performs, real number attributes for the characteristic parameters, and attributes describing input and output interface flows, of the type “Interface”, which is a custom-defined class. Because an interface is a common boundary between two subsystems, two interacting subsystems will have interface attributes that point to the same interface objects, as described in Figure 7. In this figure, the connection between ATA1 and ATA2 is done through the flow parameters ATA1.outFlow_12_1 and ATA2.inFlow_12_1, which in fact point to the same object flow_12_1 of the class Interface. Thus, any change in the output flow from ATA1 is automatically reflected on the input flow of ATA2.

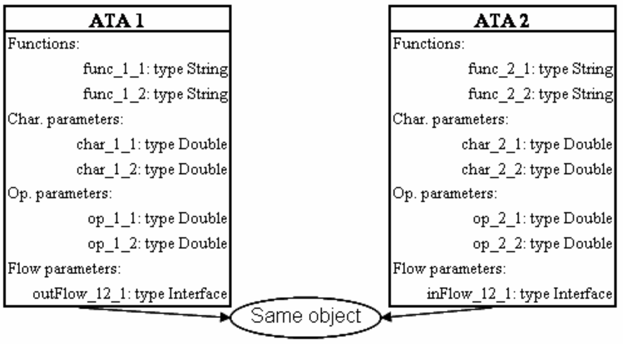


Figure 7: "ATA chapter" class diagrams

The class Interface defines a certain number of important attributes present in any interface object. First and foremost, it contains the value of the flow. The other attributes include parameters pointing to the supplier, the client, and the master. If the client subsystem defines an interface requirement (a demand on the value of the flow), the interface object needs to be provided with the name and location of the function defining that requirement. Also, the Interface class defines a string attribute that describes the type of interface (electrical, hydraulic, etc.).

5.2 Framework and Methodology

As hinted above, the object-oriented integrated framework for technology impact evaluation consists of three main components, borrowing from the domain of knowledge engineering [9]. The first component of the framework is the "Subsystem Modeler", used by the designer to define and model the various subsystems as well as the system parameters of the overall aircraft. This is done "offline", prior to the technology evaluation study, and can be distributed across the different stakeholders of the subsystem design process. Hence the need for the second component of the framework, the "Subsystem Library", which stores the various subsystems in a central database. The third component, the "Virtual Simulator", allows the designer to load the subsystems from the Subsystem Library, instantiate and assemble them, and connect them to the aircraft.

The Virtual Simulator component offers an integrated environment where the designer may call external "expert tools" for purposes such as aircraft sizing or subsystem detail analysis and design. It allows for interactivity with the designer, who can change the values of the subsystem characteristics or resource requirements and instantly visualize the impact of these changes on the other parameters, without having to manage multiple platforms.

Once the aircraft is fully assembled in the Virtual Simulator, the designer can evaluate candidate technologies for insertion in the reference aircraft. The methodology for technology evaluation in the integrated environment described above is represented in Figure 8.

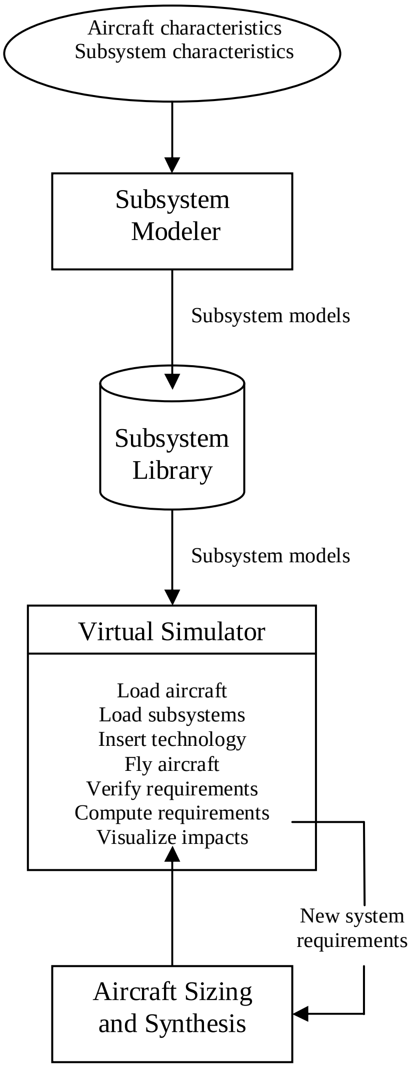


Figure 8: Integrated technology evaluation framework and methodology

To evaluate the impact of a subsystem technology on the overall system and on the interacting subsystems, the designer first inserts the technology by loading a set of values for the characteristics of the newly changed subsystem. This assumes that the impact of the technology on all the characteristics of the subsystem has been previously well evaluated and that it does not vary once the upgraded subsystem is connected to the other subsystems. The next step is then to simulate the operation of the aircraft through the different phases of a mission previously defined. During this step, the behavior of the interfaces is monitored and the Virtual Simulator verifies that the different requirements and constraints are met. Thus, one outcome is the amount of power resources that the supplier subsystems need to provide to their clients. More generally, the output of this simulation of flight operation is a new set of requirements that the system-level parameters need to fulfill. In order to determine the impact on all the considered system parameters, these requirements are then passed on to an aircraft sizing and synthesis tool, which, given a design mission and a geometry, sizes the aircraft and returns the new values of the system parameters.

6 Implementation and Example of Technology Evaluation

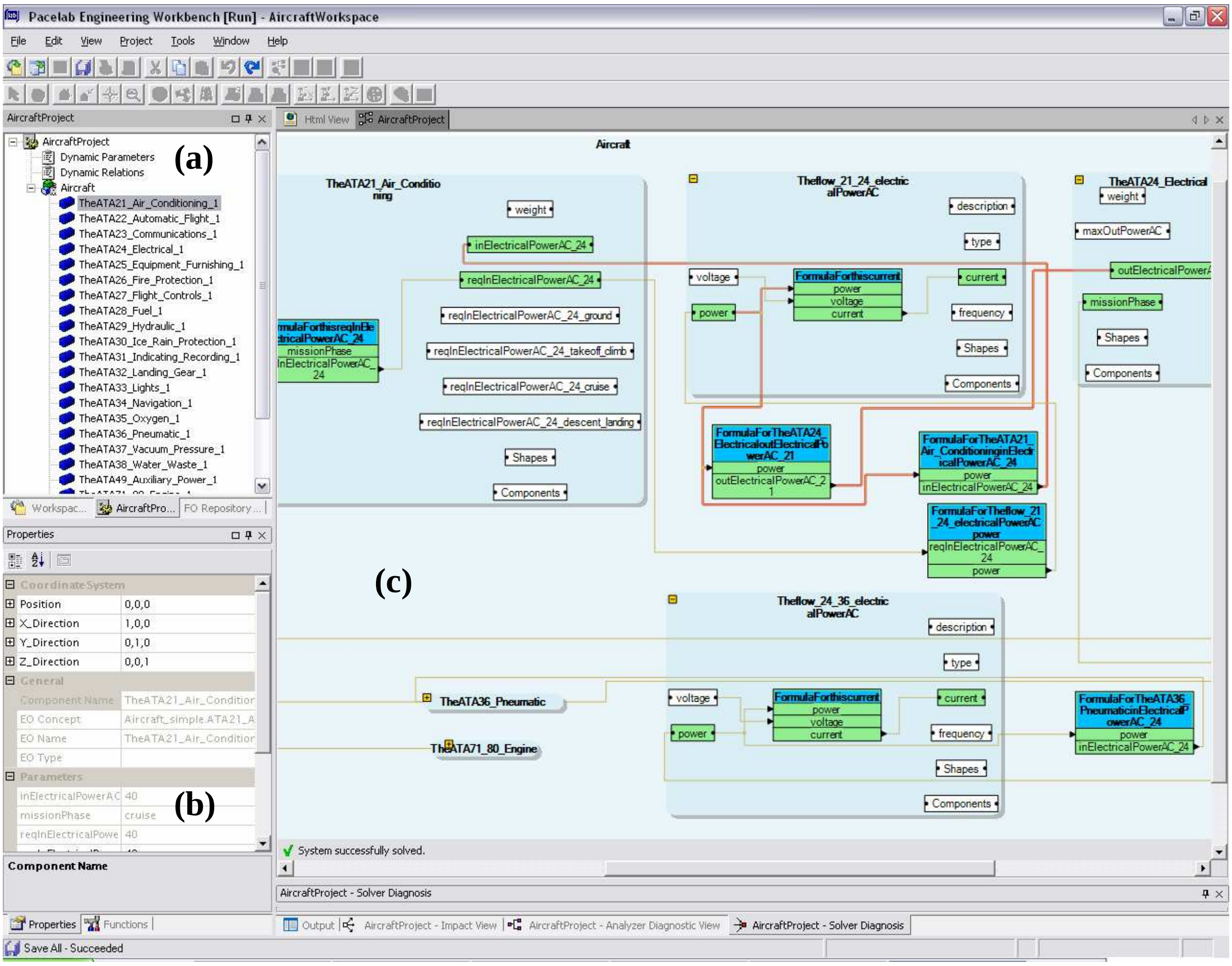
Instead of implementing the proposed framework by creating a tool coded in an object-oriented language such as JAVA or C#, a different alternative was chosen. Indeed, the authors took advantage of the Pacelab Suite [10], a knowledge-based engineering platform coded in C# that already had the built-in capabilities to support the implementation of the three component object-oriented framework mentioned in the previous section. Indeed, the Pacelab Suite, developed by Pace, consists of three interacting modules: the Knowledge Designer, the Knowledge Server and the Engineering Workbench.

The Knowledge Designer, which served as the basis for the Subsystem Modeler, is a visual environment that allows the design engineer to model complex systems and their subsystems offline. The modeling phase can be distributed across several different terminals. Among the many predefined object constructs used in Pacelab, the Engineering Object (EO) is the most fundamental one and constitutes the elementary building block of any Pacelab model. An EO is a class defined by parameters, the formulas that describe the mathematical relationships between them, and methods that define operations on the parameters. Moreover, an EO can be comprised of other EOs, thus yielding larger assemblies of EOs. When references are properly added, an EO can have access to the parameters of other EOs. The object-oriented aircraft modeling framework, developed for this paper, defines the Aircraft EO as an assembly of ATA chapter EOs. The interfaces are also modeled as EOs included as components of the Aircraft system. The flow value of an interface is defined as a parameter of that interface, and is then transmitted to the corresponding output parameter of the supplier and to the corresponding input parameter of the client. The requirement for a flow is defined via a formula inside the master subsystem. The formula takes as arguments the mission phase and operating conditions (normal or emergency), thus taking into account the criticality of a subsystem.

The Subsystem Library is handled by Pacelab’s Knowledge Server, a tool that manages Structured Query Language (SQL) databases and libraries of EOs. Finally, the Virtual Simulator builds on Pacelab’s Engineering Workbench, which provides the design engineer with a visual interactive environment where EOs can be loaded from the Knowledge Server. The Engineering Workbench possesses a mathematical engine that automatically updates the values of all the parameters of the model when a change is made. For example, when the requirement on an electrical power input for a subsystem increases, the value of the corresponding flow is automatically increased so that the value of the output flow of the supplier (ATA 24 – Electrical Power) matches the demand.

The integrated environment implemented in the Engineering Workbench can be visualized in Figure 9. As one can see, it is divided into three main panels (a, b, c in the figure) and each contains several tabs. In the first panel (a), the user can load and visualize the aircraft system, its subsystems, and the interfaces. The panel also contains a tab where the content of the Subsystem Library can be visualized. From this tab, all the available subsystems along with the technologies that they implement can be loaded and inserted into the instance of the aircraft currently under study.

The second panel (b) includes a tab named “Properties”, which lists the properties of the system or subsystem selected in panel (a), as well as its parameters. There the design engineer can manually input new values of the subsystem parameters that are not governed by a formula. Another tab contained in panel (b) is the “Functions” tab, which lists all the functions applicable to the selected system or subsystem. For example, if the Aircraft system is selected in panel (a), the function “Fly_Aircraft”, previously defined in the Subsystem Modeler, becomes visible. When launched, this function performs the different mission segments and records the resulting flow values and requirements in a text output file. It also checks that all design requirements are verified and that no constraints are violated. In the case where this fails, a warning is returned and written into



Coordinate System	
Position	0,0,0
X_Direction	1,0,0
Y_Direction	0,1,0
Z_Direction	0,0,1
General	
Component Name	TheATA21_Air_Condition
EO Concept	Aircraft_simple.ATA21_A
EO Name	TheATA21_Air_Condition
EO Type	
Parameters	
inElectricalPowerAC	40
missionPhase	cruise
reqInElectricalPowe	40

Component Name

✓ System successfully solved.

Figure 9: Virtual Simulator integrated environment in the Pacelab Engineering Workbench

the output file. For instance, if the candidate technology for insertion into a reference aircraft is an electrically powered Air Conditioning subsystem (ATA 21) eliminating the need for bleed air extraction from the engine (ATA 71 to ATA 80), the result of the Fly_Aircraft function may show that the Electrical Power subsystem (ATA 24) has too little power available to match its clients' increased needs. The output file will then contain new electrical and pneumatic power requirements that may be used by analysis tools in order to size the electric generators. After the engine is resized, the updated engine characteristics may be fed to an aircraft sizing tool such as the NASA developed FLight OPtimization System (FLOPS), which will compute the new values of the aircraft range and other aircraft level metrics, thus quantifying the impact of the insertion of a "more electric" air conditioning subsystem.

Finally, the third panel (c) of the Virtual Simulator integrated environment provides the design engineer with an interactive graphical view of the model. Each subsystem is represented by an expandable box contained in the larger box representing the overall Aircraft system. When expanded, a subsystem box shows a collection of rectangles, each of which refers to a subsystem parameter. For example, in Figure 9, the box corresponding to the Electrical Power subsystem (labeled theATA24_Electrical) contains a rectangle for the weight of the subsystem and another one representing the power of the alternative current supplied to the ATA 21 - Air Conditioning subsystem (outElectricalPowerAC_21). When a parameter value is changed, the underlying mathematical engine updates the depending parameters. A parameter with a value that is up to date will see its corresponding box colored in green in panel (c), and colored in pink in the opposite case. Thus, during the study, it is easy to determine which parameters fail to have their values updated.

When a parameter refers to another parameter of the model, their respective rectangles are graphically connected to each other via a thin line that the designer can choose to highlight during the study in order to comprehend the interrelationship between two subsystems, or to visualize the parameters directly depending on a particular one. In Figure 9, the relationship between the electrical power received by the Air Conditioning subsystem (inElectricalPowerAC_24) and the electrical power supplied by the Electrical Power subsystem (outElectricalPowerAC_21) is highlighted. One can see that both parameters are set to the value of the parameter "power" defined in the interface flow_21_24_ElectricalPowerAC. The latter parameter is given the value of the power requirement defined in the Air Conditioning subsystem (ATA 21).

7 Conclusion

The challenges posed by the study of the technology impact at the subsystem level on the aircraft system necessitate the development of new tools for systems integrators and design engineers. Indeed, the complexity of the interactions among the subsystems, and between these and the overall system can easily render of the integration of a promising technology less transparent, to the point where the designer might not realize its full implications for the system until late in the design process. In order to respond to this need for more efficiency in the design effort, a framework for technology evaluation was developed.

First, a functional and physical decomposition of a typical commercial aircraft was made using the ATA 100 specification for the subsystem definitions. In conformance to standard systems engineering practices, the main parameters of the subsystems as well as the major interfaces between the subsystems were identified. Requirements and constraints were also defined.

Once a satisfactory model of the aircraft was established, a methodology for the evaluation of the impact of a technology insertion was formulated. This methodology builds on a three-fold object-oriented framework that allows for the distributed, offline definition of individual subsystems in the Subsystem Modeler, for the storage of a subsystem database managed by the Subsystem Library, and for the "run-time" assembly of subsystem models in the Virtual Simulator. This latter module simulates the operation of the aircraft during the major mission segments in order to track the behavior of the various interfaces. The resulting new set of requirements, such as electrical power required from the Electrical Power subsystem, can then be fed to an aircraft analysis tool in order to determine the impact on the aircraft-level parameters.

This object-oriented framework was implemented in the Pacelab Suite, which offered graphical features and a mathematical engine that proved helpful to create an interactive, integrated and modular design environment.

This paper reflects the status of a work still in progress. The framework enabling the modeling of the aircraft and its subsystems has been developed. However, the proposed methodology still needs to be fully validated. This will require defining a sample aircraft, by assigning pertinent values for all its system-level and subsystem parameters. A candidate technology, for which the localized impact on the subsystem parameters will have been determined, will then be evaluated using the methodology. The latter may then be extended and automated in order to facilitate the evaluation of a technology portfolio. The physical decomposition of the aircraft may also be refined such that the ATA chapter subsystems, currently the deeper level of the architecture, may be modeled as an aggregation of physical components should the need for such detail arise. The concepts and methods presented in this paper should be applicable regardless of the level of aircraft decomposition chosen, providing transparency to the implementation of new technologies and architectures.

Acknowledgments

The authors wish to acknowledge the contribution of the other members of the research group, Fairuz Romli, Cyril de Tenorio, and Michael Sielemann.

Also, special gratitude is expressed towards Airbus and Pace for their support.

Appendix : the ATA chapters included in the model

AIRFRAME SYSTEMS	
Chapter No.	**Subsystem**
21	Air Conditioning
22	Auto Flight
23	Communications
24	Electrical Power
25	Equipment and Furnishings
26	Fire Protection
27	Flight Controls
28	Fuel
29	Hydraulic Power
30	Ice and Rain Protection
31	Indicating and Recording
32	Landing Gear
33	Lights
34	Navigation
35	Oxygen
36	Pneumatic
38	Water and Waste
49	Auxiliary Power
STRUCTURES	
Chapter No.	**Subsystem**
52	Doors and Openings
53	Fuselage
54	Nacelles and Pylons
55	Empennages
56	Windows
57	Wings
POWER PLANT	
Chapter No.	**Subsystem**
71	Power Plant
72 (T)	Engine - Turbine
73	Engine Fuel and Control
74	Ignition
75	Bleed Air
76	Engine Controls
77	Engine Indicating
78	Engine Exhaust
79	Engine Oil
80	Engine Starting

References

[1] Defense Acquisition University. *Systems engineering fundamentals*. Defense Systems Management College Press, 1999.

[2] Moir I and Seabridge A. *Design and development of aircraft systems*. 1st edition, AIAA Education Series, 2004.

[3] Jackson S. *Systems engineering for commercial aircraft*. Ashgate Publishing, 1997.

[4] Air Transport Association. *ATA Specification 100: Manufacturers' Technical Data*. Air Transport Association of America (www.airlines.org), 1999.

[5] SysML Partners. *Systems Modeling Language (SysML) specification*. www.sysml.org, 2005.

[6] Price M. et Al. Identifying interfaces in engineering systems. *AIAA Journal*, Vol. 44, No. 3, pp 529-540, 2006.

[7] Schrage D. Technology for rotorcraft affordability through integrated product/process development (IPPD). *American Helicopter Society 55th Annual Forum*, Montreal, Canada, 1999.

[8] Krus P. Systems engineering in aircraft system design. *11th Annual International Council on Systems Engineering*, July 2001.

[9] Tooren M.J.L. van, M. Nawijn, J.P.T.J. Berends and E.J. Schut. Aircraft Design Support using Knowledge Engineering and Optimisation Techniques. *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Austin, Texas, USA, 2005.

[10] Pace. *Pacelab Suite, Administrator guide*. Pace Aerospace Engineering and information Technology GmbH (http://www.pace.de/en/), 2006.