1

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Services des thèses canadiennes

Ottawa Canada
K1A 0N4

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses
on Microfiche Service

Services des thèses canadiennes
sur microfiches

Ottawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres imprimées ou dactylographier

Full Name of Author — Nom complet de l'auteur

JEAN PAGUETTE

Date of Birth — Date de naissance

12 DEC 1957

Country of Birth — Lieu de naissance

CANADA

Permanent Address — Résidence fixe

1 LE BARRON
GATINEAU, QUEBEC
JST 4C3

Title of Thesis — Titre de la thèse

THE NEXT-GENERATION AVIONIC SYSTEM - A PREDICTION

University — Université

CARLETON UNIVERSITY

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

MASTERS IN ELECTRICAL ENGINEERING

Year this degree conferred — Année d'obtention de ce grade

1985

Name of Supervisor — Nom du directeur de thèse

PROF. B.A. BOWEN

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date

24 SEP 1985

Signature

Jean Pagnette

THE NEXT GENERATION AVIONIC SYSTEM - A PREDICTION

by

JEAN J.R. PAQUETTE, B.Sc.

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of
Master of Engineering

Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University

September, 1985

© copyright

1985, Jean Paquette

The undersigned hereby recommends to the Faculty of Graduate Studies and Research acceptance of the thesis,

"~~THE~~ NEXT GENERATION AVIONIC SYSTEM - A PREDICTION"

submitted by JEAN PAQUETTE, B. SC., in partial fulfillment of the requirements for the degree of Master of Engineering.

Thesis Supervisor

Chairman,
Department of Systems and
Computer Engineering

Department of Systems and Computer Engineering

Faculty of Engineering

Carleton University

September, 1985

ABSTRACT.

The major characteristics and development mechanisms of the next generation of avionic systems are defined in terms of system architecture, integration approach, and design methodology. This definition is derived from the evolution of avionic systems over the past two decades, taking into account the technology, requirements, constraints, etc., expected to be prevalent in the 1990s. This description is then confronted with the various demands and constraints governing the implementation of such systems. This confrontation serves to outline the problem areas that will be critical to the achievability of the proposed system. Specific observations are made regarding the necessary actions required, both by government and industry, for the successful implementation of the proposed system.

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to his supervisor Professor B. A. Bowen, for his guidance and advice during the preparation of this thesis.

The author also wishes to thank Captain William R. Brown for his enlightening comments and advice throughout the formation of this thesis.

Je voudrais surtout remercier ma femme Francine pour son support, son courage, ainsi que la patience inepuisable dont elle a fait preuve au cours de ces deux dernieres annees.

TABLE OF CONTENTS

Abstract	1
Acknowledgements	11
Table of Contents	111
Chapter 1: THESIS STATEMENT AND CLAIMS	1
Introduction	1
Claims	4
Implications	7
Thesis structure	18
Chapter 2: CLASSES AND METHODOLOGIES	21
2.0 Introduction	21
2.1 Avionics Architecture	22
2.1.0 Introduction	22
2.1.1 Centralized Systems	23
2.1.2 Federated Systems	26
2.1.3 Distributed Systems	29
2.1.4 Hierarchical Systems	33
2.1.5 Summary	36
2.2 Integration Approaches	37
2.2.0 Introduction	37
2.2.1 Integration By Accretion	38
2.2.2 Integration By Evolution	40
2.2.3 Planned Integration	43
2.2.4 Total Integration	46
2.2.5 Summary	47
2.3 Design Methodologies	48
2.3.0 Introduction	48
2.3.1 Bottom-Up Design	49
2.3.2 Top-Down Design	51
2.3.3 Summary	55
2.4 Chapter Summary	56

Chapter 3: EVOLUTION · CAUSES AND EFFECTS	57
3.0 Introduction	57
3.1 Avionic Systems Evolution	58
3.1.0 Introduction	58
3.1.1 Threat versus Technology	58
3.1.2 Technology Evolution	61
3.1.3 Requirements Evolution	64
3.1.4 The "Integration Problem"	70
3.2 Avionics Architecture	72
3.2.0 Introduction	72
3.2.1 Pre-Centralized	73
3.2.2 Centralized to Federated	75
3.2.3 Federated to Distributed	78
3.2.4 State-Of-The-Art Architecture	81
3.2.5 Summary	82
3.3 Integration Approaches	83
3.3.0 Introduction	83
3.3.1 Pre-Accretion	84
3.3.2 Accretion to Evolution	86
3.3.3 Evolution to Planned	90
3.3.4 State-Of-The-Art Integration Approach	93
3.3.5 Summary	94
3.4 Design Methodologies	95
3.4.0 Introduction	95
3.4.1 Pre-Bottom-Up	95
3.4.2 Bottom-Up to Top-Down	99
3.4.3 State-Of-The-Art Design Methodology	106
3.4.4 Summary	107
3.5 Chapter Summary	107
 Chapter 4: THE NEXT GENERATION	 111
4.0 Introduction	111
4.1 Avionics Architecture	112
4.1.0 Introduction	112
4.1.1 Projected Technology	112
4.1.2 Projected Constraints	117
4.1.3 Projected Requirements	118
4.1.4 Evolutionary Trends	125
4.1.5 Bottom Line	126

4.1.6 Next Generation Architecture	130
4.1.7 Summary	132
4.2 Integration Methodology	133
4.2.0 Introduction	133
4.2.1 Projected Technology	134
4.2.2 Projected Architecture	134
4.2.3 Integration Problems	135
4.2.4 Projected Constraints	136
4.2.5 Projected Requirements	136
4.2.6 Evolutionary Trend	137
4.2.7 Bottom Line	138
4.2.8 Next Generation Integration Concept	140
4.2.9 Summary	140
4.3 Design Methodology	141
4.3.0 Introduction	141
4.3.1 Projected Technology	142
4.3.2 Projected Architecture	143
4.3.3 Projected Integration Concept	144
4.3.4 Projected Constraints	145
4.3.5 Projected Requirements	146
4.3.6 Evolutionary Trend	148
4.3.7 Bottom Line	148
4.3.8 Next Generation Design Methodology	149
4.3.9 Summary	150
4.4 Chapter Summary	150
 Chapter 5: ACHIEVING THE NEXT GENERATION	 153
5.0 Introduction	153
5.1 Avionics Architecture	154
5.1.0 Introduction	154
5.1.1 Information Transfer System	154
5.1.2 Artificial Intelligence	158
5.1.3 Two Level Maintenance	161
5.1.4 Standardization Issues	165
5.1.5 Miscellaneous	166
5.1.6 Summary	167
5.2 Integration Methodology	167
5.2.0 Introduction	167
5.2.1 Design versus Integration Methodology	168
5.2.2 Avionics Architecture	171
5.2.3 Standardization Issues	172
5.2.4 Summary	173

5.3 Design Methodology	174
5.3.0 Introduction	174
5.3.1 Inertia of Major Development Programs	174
5.3.2 Support Tools	176
5.3.3 Requirement Specification	181
5.3.4 Standardization Issues	183
5.3.5 Summary	185
5.4 Chapter Summary	186
 Chapter 6: CONCLUSIONS	 187
Introduction	187
Claims	187
Implications	188
 References	 201

LIST OF FIGURES

Figure 1.1	A Perspective for Viewing the Major Issues in the Development of the Next Generation of Avionic Systems	3
Figure 2.1	Central Architecture	24
2.2	Federated Architecture	27
2.3	Distributed Architecture	30
2.4	Hierarchical Architecture	34
Figure 3.1	Influence of Technology on Avionic Performance Requirements	60
3.2	Speed Power Product versus Time	65
3.3	Past and Predicted Logic Density	66
3.4	Current Avionic System Development Program Structure	71
Figure 4.1	Growing Influence of Software on Avionic System Design and Development	115
4.2	Pave Pillar Implementation	131
Figure 5.1	Information Transfer System Comparison	156
5.2	Modular Concept Supports Two Level Maintenance	163
5.3	Hardware Modularity Trade-Off	164
5.4	Program Structure for Standardization	184

CHAPTER 1 THESIS STATEMENT AND CLAIMS

INTRODUCTION

Before discussing avionic systems to any extent, it is essential to define what avionic systems are. To this purpose, the following definitions are presented:

Avionics

"Electrical and electronic devices for use in aviation" [40,p2]

Aviation Weapon System

"A system formed by the combination of a flying platform (aircraft) with a set of avionics, and whose function is to fulfill specific mission requirements"

Avionic System

"A subsystem of the weapon system whose function is to detect and process observable data about the environment inside and outside the aircraft, to derive information in a form suitable for crew utilization, to accept crew inputs and combine it with other derived information to activate controls within the aircraft and stimulate the outside environment as dictated by the mission requirements of the weapon system" [40]

Avionic systems are the backbone of contemporary military Air Forces of all major countries throughout the world. It is through avionics that demanding operational requirements are met and translated into increased operational capabilities. These avionics, accounting for approximately one third of a weapon system's total cost, represent investments in the tens of billions of dollars for a procurement of several hundred air-

craft. This is why avionic systems are an important variable in the power struggle presently going on between the major superpowers. The next generation of avionic systems are those that will become operational in the 1990s. These systems are crucial in determining the operational capabilities of the Air Forces of the 1990s.

This thesis proposes that by viewing the overall concept of avionic systems development from the perspective of figure 1.1, and by analyzing past and present trends of this development, it is possible to achieve the following: first, to predict the general characteristics of the next generation of avionic systems; second, to define the appropriate system implementation philosophy, along with its supporting mechanisms, necessary to achieve such a system; and finally, to determine several real world implications which must be taken into account if the enormous investments required by modern avionic system development is to be justified. Such justification will be achieved if the implemented weapon system meets specified operational requirements at a minimum, and cost-effective, level of development time and resource expenditure.

Figure 1.1 presents an abstract representation of a three level perspective of the overall problem of developing the next generation of avionic systems. At the center, level 1, is the avionic system itself. This circle represents the outer virtual machine layer of the logical system perspective as presented by Bowen and Brown in [4]. Within this level are all the concentric layers of virtual systems comprising the logical description perspective of the actual hardware and software that

Figure 1.1 A Perspective for viewing the Major Issues in the Development of the Next Generation of Avionic Systems.

make up the final physical avionic system.

Immediately surrounding this logical system representation is a higher level of abstraction, level 2, which represents the implementation issues of creating the system of level 1. Outside of level 2 is the highest level of abstraction of figure 1.1. This level 3 deals with the real world considerations imposed on the implementation of the desired system. The design path that will lead to the desired avionic system originates outside of level 3, and proceeds through levels 3 and 2, to end up in level 1 with the appropriate system. This is the only path that will lead to a satisfactory system.

CLAIMS

This thesis sets forth three basic claims. These claims define the necessary characteristics and development mechanisms of the next generation of avionic systems. The first claim predicts the general characteristics of the next generation avionic system corresponding to level 1 of figure 1.1.

- 1.1 The next generation avionic system must be characterized by a fully integrated, hierarchical system architecture (FIHA).

An examination of the evolutionary trends of avionic systems, as presented in Chapters 2 and 3, reveals that the present day partially integrated distributed architecture is only a stepping stone in the evolution toward the next generation avionic systems. Partially integrated distributed systems

cannot fulfill the demands for the avionic systems of the 1990s which are currently being formulated. These demands include such operational and functional requirements as information fusion, combined centralized and distributed processing, functional isolation, reduced bus traffic, increased information handling and reduction capabilities, and ease of reconfiguration and control. Chapter 4, Sections 4.1 and 4.2, conclude that the only foreseeable system capable of meeting these requirements must be based on a fully integrated, hierarchical system architecture.

The successful implementation of a fully integrated hierarchical avionic system will depend heavily on the adoption of an appropriate system development and integration philosophy or overall approach, supported by appropriate implementation mechanisms. This philosophy and the supporting mechanisms correspond to the two major aspects of system implementation represented by level 2 of figure 1.1.

An examination of past and present trends with relation to the general approaches taken toward avionic systems design and integration shows that these have traditionally been treated as separate issues. This has given rise to major avionic system implementation problems generally referred to as the "integration problem". The following two thesis claims deal specifically with the issues of system implementation as defined by level 2.

- 2.1 The successful implementation of the next generation avionic system requires a Total Integration approach.

The present standard approach of Planned Integration, whereby system integration is attempted through software development following subsystem selection, is not adequately meeting the system integration requirements of current systems. It will not meet the requirements of the next generation systems which include: implementation on a Hierarchical architecture; partial solution to functional overpopulation and high performance requirements of modern avionics; increased availability; reduced maintenance and life cycle costs; standardized approach; adequately consider all the interactions in the system; and, consistent with the evolutionary trend of increasing integration. Given the examination of past and present system integration approaches presented in Chapters 2 and 3, Chapter 4, Section 4.2 concludes that only an overall Total Integration approach/philosophy, whereby all aspects of system and subsystem interactions are considered from the outset of the project, can result in the successful implementation of a fully integrated hierarchical avionic system.

- 2.2 The next generation avionic systems must be designed and implemented by the systematic application of a Top-Down design methodology, making full use of the associated support mechanisms.

Several variations of Top-Down design methodologies and support mechanisms are presently widely accepted and applied in

many aspects of avionic and electronic systems design. However, the Top-Down design methodology is not yet being applied consistently at the overall system level throughout industry, as the basic implementation mechanism for avionic systems. This is why many of the problems of Bottom-Up design are still apparent. Chapter 4, Section 4.3 concludes that the Top-Down design methodology is the only methodology capable of both meeting the design requirements of the next generation avionic systems and supporting the concept of total integration. Hence a Top-Down design methodology must be used to successfully implement a fully integrated hierarchical avionic system.

Given the previous prediction that the next generation avionic system will be a fully integrated hierarchical system, the claims that its implementation must be based on a philosophy of Total Integration and supported by a Top-Down methodology would appear to be nearly obvious. Yet, it is important to realize that neither the concept of a Total Integration approach, nor the strict application of a Top-Down design methodology, are in widespread use, at the system level, throughout the present day avionics industry.

IMPLICATIONS

Knowledge of the characteristics and implementation mechanisms necessary for creating the next generation avionic system, does not guarantee the achievability of such a system. However, the thesis claims

give rise to several implications which not only govern the implementation of the FIHA system, but are pre-requisites for its achievability.

The implementation of the FIHA system imposes specific demands on both government and industry. These demands must be fulfilled. Along with the demands, there are constraints imposed as well. Designers must account for these constraints. The combination of the thesis claims about the next generation avionic system and of the demands and constraints relative to its implementation, inevitably lead to several conclusions as to the achievability of such a system. These conclusions provide an indication of what can realistically be expected, and when. In the following summary the demands are presented first, followed by the constraints and conclusions respectively.

Demands

There are three main classes of demands imposed on the implementation of the next generation avionic system. The first deals with the production of appropriate requirement specifications.

- 3.1 a) The FIHA avionic system will not be achieved unless government and military personnel are able to produce complete and static requirement specifications.

Chapter 5, Section 5.3.4, examines the implementation demands of the Top-Down design methodology and concludes that a crucial pre-requisite for the proper application of the Top-Down design methodology is the production of complete and static requirement specifications. However, the avionics industry

does not produce the requirement specification. This is done by military and government personnel. Therefore, these persons must be able to produce suitable requirement specifications if the development of the next generation of avionic systems is to be successful.

The second class of demands addresses the so-called "integration problem". This is a problem characterized by the integration phase of avionic system development programs, both new systems and updates/upgrades, becoming unacceptably complicated, lengthy, and costly. For the next generation avionic system to become a reality, this integration problem must be solved.

- 3.1 b) The "integration problem" can only be solved by the following two actions: implement the Total Integration approach; and, implement a suitable standardization program.

The integration problem has two aspects. The first is that of developing new avionic systems. Current integration approaches are not well suited for the design of the sophisticated systems already in use. Thus, the integration phase has become unacceptably complicated and this translates in long development times and high development costs. Next generation requirements of reduced development time and cost, coupled with increasing avionics complexity, mean that the integration phase must be rationalized and made more cost-effective. This is accomplished through the implementation of the Total Integration approach. However, this solution is only valid for the initial

phase of the weapon system life cycle, i.e. while it's design specification of mission requirements is still valid.

The second aspect of the problem is that of performing system integration upon the update/upgrade of an avionic system. The combination of rapidly advancing technology and of increasing mission performance requirements has the effect of quickly rendering the avionic system obsolete. The ideal solution at this point is to design a new system to match the new mission performance requirements. However, Chapter 3, Section 3.1, concludes that the combination of the high cost of avionics with the rapid rate of requirements evolution, forces an update/upgrade of existing systems well before the end of their useful service life, instead of new development programs. This update/upgrade generally takes the form of the introduction of either new technology or new functional capabilities or both. With today's highly complex and highly integrated systems, this update/upgrade has a major impact on the overall system structure and, thus, results in a complicated and costly integration attempt. This integration can be simplified, if not eliminated, by having a technology transparent system and a system where functional changes can be effected with little or no impact on the remainder of the system. Such a system requires the application of extensive standardization. However, standardization has its limits of applicability.

3.1 c) Total Integration must be achieved through a Top-Down design methodology.

Total integration is achieved by finding all the interactions within the avionic system and by coming up with suitable compromises to each interaction. The Top-Down design methodology is the process by which all interactions are found and all suitable compromises are implemented. Thus, the Top-Down design methodology performs all the actions required by the Total Integration approach. This fact suggests that the design methodology is an appropriate implementation mechanism for the Total Integration approach.

Furthermore, Chapters 3 and 4 emphasize the fact that there is a yet-to-be-fulfilled requirement for integration as early as possible in the development process. This requirement, coupled with the fact that the design methodology is the initial phase in the development process, imply that the Top-Down design methodology is the only mechanism by which to achieve all the goals of the Total Integration approach and thus solve part of the integration problem.

3.1 d) The FIHA avionic system will not be achieved unless the avionic system development program structure is changed to include integration as part of it's design phase.

The Total Integration approach will be achieved through the Top-Down design methodology which is the initial phase of the development program structure. Traditionally, system in-

tegration is accomplished in the next to last phase of the development program structure. Therefore, the traditional development program structure will have to be changed to include integration and verification as a part of the initial system design phase.

The last set of demands outlines the need for the use of a simulation/integration lab facility in all aspects of avionic development programs.

- 3.1 e) The FIHA avionic system will not be achieved unless it is developed with the help of a simulation/integration lab facility.

The Total Integration approach and Top-Down design methodology imply finding all interactions within the avionic system, finding compromises to each interaction, evaluating the impact of each compromise, and implementing the suitable compromises. With the complexity of current and expected avionic systems, it is impossible for the designer to think about all interactions, let alone find suitable compromises to every one. However, a simulation/integration lab, through system simulation, can help the designer to find "hidden" interactions. It can also help in evaluating the impact of the FIHA compromises. The simulation/integration lab is the only tool that allow the designer to find a maximum of interactions along with suitable compromises.

- 3.1 f) The design of updates/upgrades of totally integrated avionic systems will only be possible through the use of a simulation/integration lab.

The determination of the nature of an update/upgrade of a totally integrated avionic system can only be done through the evaluation of the impact that new requirements have on the system. This evaluation of impact implies finding all new interactions to be brought into the system, as well as the changes to be made to the existing interactions. Due to the complexity of modern aircraft, this evaluation can only be done with the help of an existing avionic system serving as a hot bench simulator. There are two possible ways to achieve this. The first one is to monopolize an aircraft for the duration of the update/upgrade development program. With the high cost of modern aircraft and their decreasing number to fulfill increasing operational requirements, such a solution is detrimental to the operational capability of any aircraft fleet. The alternate solution is to use a simulation/integration lab facility. Since such a facility is a prerequisite to initial system design, this is the only cost-effective way to design updates/upgrades.

- 3.1 g) The FIHA avionic system will not be achieved before, approximately, the year 2010 unless suitable simulation/integration lab facilities are used by industry to demonstrate the changes required by this system to existing concepts and procedures.

Avionic system development programs are high cost programs (several billion dollars or more). Such high investments create a need to minimize the risks involved in the development programs. This risk minimization takes on the form of a slow acceptance of changes (i.e. an inertia) within government and industry. Therefore, significant changes have a tendency to occur over several incremental steps. The FIHA avionic system represents a major change from current systems such that it is likely to be developed over several iterations of avionic systems. Since the next generation avionic system is, by definition, only one incremental step away, the acceptance of changes must be accelerated.

The only way to achieve an acceleration in the acceptance of new concepts is through the demonstration of the proposed changes. Such demonstrations can only be achieved cost-effectively by a simulation/integration lab. Since industry performs the development of avionic systems, under the control of the Project Management Office (PMO), the above statement follows.

Constraints

Standardization is seen to be a major part of the solution of the integration problem. It is needed to achieve Total Integration as well as for updating and upgrading avionic systems. Standardization will also

help to reduce life cycle costs through the elimination of the proliferation of incompatible systems, through an increased ease of growth and modification, and, finally, through the simplification of maintenance procedures. However, the designer has to live with specific constraints regarding this needed standardization.

- 3.2 a) The standardization to be applied to the next generation avionic system must be as high level as possible.

Chapter 3, Section 3.1, examines avionic system requirements and concludes that there is a requirement for continually increasing mission capabilities. It is also observed that such a requirement has been traditionally fulfilled by the application of new technology products. Low level standardization prevents the insertion of new technology components. Therefore, the standardization to be applied must be as high level as possible.

- 3.2 b) There is a limit to the amount of standardization to be applied to avionic systems.

Part of the integration problem, that of updating/upgrading existing systems, can be ultimately solved by the application of total standardization. However, such standardization prevents the insertion of new technology, thereby negating all the benefits of updating/upgrading an avionic system. Therefore, a compromise between the extent of standardization and new technology insertion is required.

Conclusions

As a result of the previous discussion, several conclusions can be drawn.

- 3.3 a) A fully integrated Hierarchical architecture will not be developed in time for the next generation avionic systems.

Chapter 4 observes that the full development of a Hierarchical architecture is dependent upon the existence of suitable artificial intelligence technology. Such technology will not become reality without extensive research and development. Extensive research and development implies a significant investment in terms of money and time. The next generation of avionic systems should appear in the 1990s, therefore, only limited research and development can be done before this next generation. This limit in the research and development translates directly into a limit on the available artificial intelligence technology. This will prevent the FIHA system from being fully implemented as the next generation of avionic systems.

- 3.3 b) The application of the Total Integration approach is possible for the next generation of avionic systems.

The achievability of the Total Integration approach is dependent on three factors. The first one is the application of high level standardization. This is already underway and sufficient standardization will be available for total integra-

tion. The second factor is the existence of a multileveled, multiplexed information transfer system (ITS) with a contention protocol. Such an ITS is not in existence now but has a high probability of being implemented by the mid-1990s. The final factor is the possibility of performing Top-Down design. This will be possible.

- 3.3 c) The "integration problem" will not be eliminated completely. It will only be minimized through the application of Total Integration as a part of system design, and of a compromised standardization program.

The Total Integration approach, achieved as part of initial system design, takes care of only one part of the integration problem. The other part, that of updating/upgrading existing systems, can only be solved by the application of total standardization. However, there is a requirement for a "compromised amount" of standardization to be applied to avionics. Therefore, the integration problem will not be eliminated completely. The amount of standardization allowed will help minimize the problem of updating/upgrading existing systems.

- 3.3 d) The application of the Top-Down design methodology is possible for the next generation of avionic systems.

The achievability of Top-Down design is dependent on three factors. The first one is a change in the development structure of avionic systems. This should not be a problem provided that industry and government start working on this change immediately. The second factor is the production of complete and

static requirements specifications. This should not be a problem either, provided that military and government personnel are made aware of the importance of these requirement specifications and that a close cooperation between government and industry be started. Such a cooperation has the most potential of producing suitable requirement specifications. The last factor is the existence of a simulation/integration lab. Such facilities are already in existence but many more are needed. If funds can be awarded in the very short term, these facilities can be built in time.

3.3 e) The FIHA avionic system will not be fully implemented before approximately year 2005.

The major factor affecting the achievability of the FIHA system is the acceptance of the significant changes required by this new system. A demonstration of the changes can accelerate this acceptance, but the huge investment, required by the development program coupled with the human natural "resistance to changes" will still result in a delay in the appearance of the FIHA system in an operational context.

Thesis Structure

The thesis is divided in three parts. The first part is the prediction of the characteristics of future systems, based on an extrapolation of their evolution over the past two decades. For the purpose of this defense, the "integration level" characteristic of avionic systems will

be treated as a single entity with the integration approach. This is only logical as the integration approach determines the extent of integration applied to a specific system. Furthermore the integration approach/philosophy and the design methodology will be treated as totally separate entities. This is done on the one hand to cater to current industry practices and on the other hand, to arrive at conclusions not encumbered by pre-conceived ideas about the end result.

Before attempting to describe the ideal characteristics of future avionic systems, one must first be knowledgeable of the characteristics of avionic systems in general. Chapter 2 describes the various system architectures, integration approaches, and design methodologies that have characterized the evolution of avionic systems for the past twenty years. Once the characteristics of past and present avionic systems have been established, it is possible to analyze and understand the evolution of these systems. This is the subject of Chapter 3. Chapter 3 identifies the important evolutionary trends, which in turn leads to several conclusions regarding the nature of the existing problems of avionic systems, their causes, and their most likely solution. Such an analysis is a pre-requisite for the prediction of the ideal characteristics of the next generation of avionic systems.

Chapter 4 develops a description of the characteristics of the next generation systems through an extrapolation of the evolution of these systems presented in Chapter 3. This extrapolation takes into account such factors as the continually changing technology, the ever increasing

requirements, the restrictive constraints, and the pertinent evolutionary trends.

The second part is the determination of the problems areas that will be encountered in the implementation of the FIHA system. Chapter 5 confronts the theory of Chapter 4 with the real-life considerations that impact the realization of avionic systems. These considerations include: the information transfer system; artificial intelligence; the maintenance philosophy; standardization; the integration approach implementation; the requirement specification; and, the support tools.

The third part is the analysis of the issues set forth in the previous chapters to arrive at conclusions regarding the solutions to the identified problems and the resulting achievability of the FIHA system. These solutions can range from the acceptance of conceptual changes, to the immediate allocation of resources for their implementation. Chapter 6 considers these solutions and draws conclusions as to the achievability of the FIHA system, based on the confrontation of Chapter 5.

CHAPTER 2 CLASSES AND METHODOLOGIES

2.0 INTRODUCTION

To understand the evolution of avionic systems engineering, one must first become acquainted with the various architectural classes, integration approaches, and design methodologies that have characterized this evolution. It is, therefore, essential to categorize and describe these architectures, approaches and methodologies.

This chapter describes the various architectural classes, integration approaches, and design methodologies that have played a major role in avionic systems engineering from the 1960's to the present. The specifics of the integration approaches and design methodologies may vary from one program to the other but what is presented here is a generalization of the various approaches and methodologies. Such a generalization is essential if one is to have an overall view of avionic systems engineering and its evolution.

Section 2.1 presents the architectural classes of importance in the evolution of avionic systems. Section 2.2 discusses a categorization of integration approaches. Finally, section 2.3 describes the various design methodology categories identified.

2.1 AVIONICS ARCHITECTURE

2.1.0 Introduction

This section identifies the avionic system architectures found in the literature that have characterized the evolution of avionic systems since the early 60's. It will classify them and describe the resulting architectural classes so as to provide the reader with a working knowledge of system architectures. This description will cover the characteristics, advantages, and disadvantages of each of the classes.

Before going any further, it is important to define what is understood by avionic system architecture. Strictly speaking, one can differentiate between topology and architecture. The topology of the system describes the information paths that interconnect the system resources [36,p82]. The architecture of the system includes such considerations as the mechanisms for controlling the actions of the system, the management of the overall use of system resources, and the management of error conditions [36,p82]. However, in the most general terms, the architecture includes the topology [36]. For the purposes of this thesis, the definition of an avionic system architecture is:

"The way system and sensors are connected together and the method used to control the data communications"[39]

Sections 2.1.1 to 2.1.4 describes the following classes of architectures:

- Centralized (figure 2.1): shared computational capability.
- Federated (figure 2.2): functional independence.
- Distributed (figure 2.3): partitioned system, and
- Hierarchical (figure 2.4): array of intelligent subsystems.

2.1.1 Centralized Systems

The centralized system basically collects information from various sources, provides a shared computational capability and provides some form of integrated data to the pilot and/or tactical crew [36].

Characteristics

A central processing unit provides all system control [42]. It acts as a central manager for all subsystems [21]. There is a single complex executive resident in the central processor's memory [42]. The application programs cover all avionic system functions and are contained in the memory of the central unit [42].

The central unit consists of a powerful time-shared central computing center which handles the computation and information exchange required of the subsystems [21], [42]. All communication passes through this central unit and the various subsystems are interconnected together and to it by a multiplexed bus [21], [42]. All other computers look like peripherals to the central unit [42]. The system does some failure detection and reconfiguration using alternate sensors or actuators thus

Figure 2.1 Central Architecture

providing limited fault tolerance [21].

Advantages

The centralized system advantages are as follows:

- There is a minimum amount of hardware required for all computing and information flow [21].
- The system provides for maximum access to all sensors in minimum time paths [21].
- There is a maximum of data available at a central source for decision making and system status updating [21]. and
- By performing some of the subsystem to subsystem tasks, the system allows some reduction of the pilot and/or operator's workload [16].

Disadvantages

The main disadvantages of the centralized system are as follows:

- The system provides low integrity. That is, the probability of all systems failing at once is the same as for only one system to fail. This is because it merges in one box many functions traditionally performed by several boxes [9].
- The integrity can be increased with the provision of a backup capability but this is possible only with the addition of a second

redundant central processor. This is not only expensive but redundant systems quickly become difficult to manage [36], [16].

- It does not provide the computational capability required of modern avionics [16].
- Growth is difficult [16].
- There is no dynamic reconfiguration capability [42].
- The executive software is quite complex. In fact, it is so complex that it generally cannot be validated completely [2]. This makes the system quite expensive to maintain both in time and cost, and
- With high performance requirements, it imposes a high bandwidth requirement on the data bus. Such bandwidth causes significant electro-magnetic compatibility (EMC) problems [10].

2.1.2 Federated Systems

Such a system consists of several processing elements, each dedicated to a particular pre-assigned task within the element, but under the control of a single central executive [12], [22].

Characteristics

The main characteristic of the Federated System is its functional independence. That is, each equipment performs its own computation.

Figure 2.2 Federated Architecture

contains its own power supply, and performs its own self-test [35].

All the processing elements are linked together through a multiplexed data bus. One unit provides for general system control with the single central executive being resident in this unit [42]. All application programs are limited to local functions in any one processor [42]. The processors and their associated memories are independently programmed and operated [32]. The only transfer path is at the I/O level [36], [32].

As can be seen, the hardware is tailored to a function [42]. The system exhibits low data bus rate communication, with the bus being treated like a peripheral [42]. Finally, the processors may have different architectures [42].

Advantages

The Federated System advantages are as follows:

- Flexibility and ease of growth [35].
- Low development cost and reduced development time [35].
- More system design flexibility than centralized [12].
- Reconfiguration automatic and simple [12].
- Gracefull degradation of performance [35].
- Low system vulnerability (with backup implemented) [12].

- Increased maintainability [35].
- Small measure of technological independence [36], and
- Some processor functional redundancy (by duplication) [36].

Disadvantages

The main disadvantages of the Federated System include:

- Extensive intersystem communication [12].
- High storage requirements [12], and
- No processing at a remote location [36].

2.1.3 Distributed Systems

A Distributed system is a combination of processing elements performing dedicated functions as a part of a partitioned system. Architecturally, it implies distributed resources, distributed management, and distributed control [36].

Characteristics

There is no single source for system control [42]. The system executive is distributed among the processing elements as individual executives. These are low complexity local executives [42]. The application programs are limited to local functions in any processor [42].

Figure 2.3 Distributed Architecture

The processors communicate with one another through a high speed bus [42]. The processors themselves have dedicated resources which can communicate with the remainder of the system only through the distributed processor [36]. There is an interchange of information among all the processors as well as among the other elements of the system [22]. This is facilitated by the fact that processors have access to all memories and are directly related to one I/O channel [32].

Finally, a distributed system exhibits one of two kinds of coupling. In the loosely coupled system, the processing in the various areas is mostly autonomous and asynchronous, with data being exchanged via the low bandwidth multiplexed bus [10]. On the other hand, in the tightly coupled system, processing elements are concerned with parallel aspects of the same tasks. They are not autonomous, are probably not geographically separated and will probably share common resources [10].

Advantages

The distributed system advantages are as follows:

- The central computer is eliminated as a central node or switching element [6].
- The hierarchical decomposition of subsystem functions can be directly implemented yielding a set of smaller more manageable programs instead of one large central program difficult to manage [6].

- Reduces the bandwidth requirement through the transmission of processed data between subsystems [10].
- It offers the most flexibility in the partitioning and allocation of functional program units to software, firmware, or new technology hardware [32].
- Autonomy of subsystems is possible and accounts for better reliability and survivability [6], and
- The vulnerability is reduced through locally dispersed computing resources which have reconfiguration capability [6].

Disadvantages

The main disadvantages of the Distributed System are:

- The interplay of the individual algorithms is less deterministic than in the centralized case. This implies that each part must operate without a complete instantaneous knowledge of the state of all other parts [6], and
- Even with the application programs being distributed it is still necessary that their functional authority and the validity of data bases be system-wide managed [6].

2.1.4 Hierarchical Systems

A Hierarchical System is basically:

"an array of intelligent subsystems under the control of an upper echelon complex of processors that coordinate the operation of all the system elements to perform the required function." [8]

Characteristics

This system provides an implied control scheme that is both natural and simplistic in structure. It simply matches the system's hierarchy [16]. A Hierarchical system has multiple levels of data bus [16]. It provides for a global data bus which interconnects all major systems [39]. This global bus system looks distributed [42]. Individual subsystems retain their functional isolation and identity through the use of dedicated subsystem buses [39]. This local bus system looks federated with the global bus interface computer acting as the executive [42]. Sensors shared by several systems are interconnected to these systems using a common sensor data bus [39].

The global bus interconnecting major systems is a high speed bus. However the local bus used for control only needs to be a low speed bus [42]. The processors may have the same architecture or their capabilities may be tailored to specific needs [42]. Finally reconfiguration is not difficult [42].

Figure 2.4 Hierarchical Architecture

Advantages

The advantages of a Hierarchical System are as follows:

- Eases the tailoring of the system capabilities to the system requirements [16].
- Preserves system integrity [39].
- Composed of isolated, independent, and integrated subsystems [39].
- Standardized bus structure [39].
- Ease of growth [16].
- Ease of integration [16], [39].
- Ease of modification [8].
- Minimal operational and support costs [8].
- Independent redundancy and control [39].
- Fault-tolerant reconfiguration [39], and
- Compatible with efficient use of available resources [8].

Disadvantages

The main disadvantages of the Hierarchical System are as follows:

- There are potential problems in system allocation and control [16], and
- The architecture does not minimize the number of processors required to support the total computational load of the system. However, this is less significant with low cost microprocessors [8].

2.1.5 Summary

The major classes of architectures that have characterized avionic systems for the past twenty years are: the Centralized, Federated, and Distributed architectures. Due to the long service life of aircraft, each of these architectural classes are still in existence today but the Centralized and Federated architectures are disappearing. The Hierarchical architecture is not yet in existence but is expected to be in the near future. It was included in this chapter because of its expected impact on avionic systems. Even if it was not going to be implemented, it represents a logical evolution from the Distributed system and thus has significant theoretical value. Chapter 4 will determine if the value of the Hierarchical architecture will remain theoretical or if such an architecture will be implemented.

2.2 INTEGRATION APPROACHES

2.2.0 Introduction

This section identifies the integration approaches that have emerged in the literature and that are deemed important in the evolution of avionic systems. It classifies them and describes the resulting categories so as to provide the reader with a working knowledge of integration approaches. This description covers the characteristics, advantages, and disadvantages of each of the approaches as well as their applicability to architectures and design methodologies. It is important to note here that only integration done at system level is of interest and hence the corresponding approaches.

However, at this point it is important to define what is understood by an integration approach:

"The process through which different functions of a system and their implementing mechanisms are combined together to achieve a higher level function" [3,p2]

Sections 2.2.1 to 2.2.4 describe the following integration approaches:

- Integration by Accretion: limited to interface considerations.
- Integration by Evolution: combines functions with natural affinities.
- Planned Integration: controls architecture through core subsystems.

tems, and

- Total Integration: identifies all interactions within the system.

2.2.1 Integration By Accretion

The philosophy behind this approach is to build a working system through the addition of independently designed subsystems. These subsystems are off-the-shelf, state-of-the-art subsystems usually originating from several different manufacturers.

Characteristics

In this integration approach, the avionic functions developed are added into the aircraft where space can be found [3]. The designer's attention is limited to the particular interface that affects the individual black box operation [41]. This results in little or no interaction between the subsystems except through the crew [16]. Therefore, the integration is performed by the crew who note the outputs, correlate the data, and control the function selection and modes [3].

Advantages

The only ~~advantage~~ that is exhibited by this approach is its simplicity.

Disadvantages

This approach is rather simplistic in nature. Rather than enhancing integration, it frustrates effective integration of avionic systems to each other and to the system [1]. The other disadvantages of this approach are:

- It cannot cope with complex systems because it rapidly leads to inflexible and highly expensive avionic suites,
- There is no standardization across platforms possible which results in a proliferation of incompatible systems from platform to platform [1], and
- The integration is delayed until all the parts of the avionic suite have been delivered [19].

Applicability

This integration approach can be applied to Pre-Centralized, centralized, and even federated architectures. But it must be remembered that it is only applicable to non-complex systems and as such, it becomes quite inefficient when multiplexing and distribution of processing are introduced.

This form of integration is achieved through a bottom-up, unstructured design methodology. As soon as one introduces structure in the system design methodology, this kind of integration becomes too simplis-

tic to be useful.

2.2.2 Integration By Evolution

This approach combines together functions with natural affinities [3]. This has come about because of a natural expansion of a particular discipline [3]. As technology evolves, ways are found such that components can be coupled more tightly together. With the Accretion approach, there was a minimum of interaction between subsystems. The Evolution approach is one which tries as much as possible to increase the interactions between the subsystems such that the resulting system will have increased performance. This increased interaction ranges from something as simple as coupling range measurements of a distance measuring equipment (DME) to a localizer signal to improve the sensitivity of the localizer needle, up to the F-18 avionics with dual redundant bus architecture, integrated cockpit with multifunction displays, extensive flight control automation and avionics programmability [3].

Characteristics

This approach covers a broad range of integration strategies. Therefore, its characteristics, advantages, and disadvantages vary with the application. However, there is one development which stands out as having had a major impact on system integration and that is the advent of the digital multiplexed bus. Therefore, the characteristics, advantages, and disadvantages of this approach will be given with relation to the multiplexed bus impact.

The main vehicle for the combination process discussed above is the multiplexed data bus. It allows sensors to be tied in parallel [19]. There is a limited processing capability associated with this kind of communication channel [16]. Finally, a large amount of integration can be done through software [19].

Advantages

Most of these advantages result from the use of a standard digital multiplexed data bus:

- Integration is no longer delayed until all parts are delivered [19].
- System integration and architecting is simplified [19].
- Allows for standardization throughout the whole weapon system [19].
- Proliferation is reduced through standardization [19].
- Increased reliability and flexibility of hardware [19].
- System configuration modifications are done in software [19].
- Allows interchange of information among all processors as well as among the other elements of the system [19], [27], and
- Simplifies system retrofits, maintenance, and logistics [19].

Disadvantages

The main disadvantages of this integration approach are as follows:

- It is not a very well formalized approach.
- It does not contain enough aspects of preplanning.
- The scope of the approach is too vast.
- It lacks clear cut aims as well as precisely defined steps to achieve these aims. Everybody goes about it the way he deems best, and to the extent he chooses, and
- Rather than being an integration approach per se, it appears to be an amalgamation of integration strategies as practised by various organizations.

Applicability

This integration approach can be applied to every class of architecture as indeed it is. However, when the system becomes very complex as with a distributed or hierarchical architecture, this approach cannot cope satisfactorily with the integration problem. However, it is a good starting point for a more complete integration approach. Suffice it to say that the Planned and Total Integration approaches build on this evolutionary approach to achieve their goals.

This approach can be achieved through any kind of structured design methodology.

2.2.3 Planned Integration

This approach tries to control the evolution of avionic architectures by dividing it into core and mission specific subsystems [3]. The traditional approach has been to physically partition the system in accordance with the functional partitioning of the mission [28]. However, if the avionics are viewed as a whole and not as a collection of prior defined subsystems, then common functions, required by each of the mission functions, can be identified [28]. This has lead to the concept of core elements.

The basic definition of a core element is one which is used irrespective of the mission being accomplished. Such core elements are found in the processing hardware, software, information transfer system, operator interface, and data acquisition and control systems. By extension, mission peculiar elements are ones only used in the execution of specific missions.

Again, the planned integration approach covers a broad range of integration strategies and thus its characteristics, advantages, and disadvantages can vary with the application. However, the DAIS (Digital Avionics Information System) is considered as typical of this integration approach. Thus the characteristics, advantages, and disadvantages of the DAIS program will be given, augmented with some from other programs.

Characteristics

This approach addresses the distribution, processing, control, and display of information in a subsystem intended to be a core or foundation for a fully integrated suite [3]. To achieve this, it uses the concepts of standard interfaces, standard computer languages, and standard instruction sets (MIL-STDs 1553, 1589, and 1750) as well as the concept of transferable software [3]. These were all introduced by the DAIS program [26]. This approach tries to simplify and increase the interactions between the subsystems.

Each set of mission dependent processing elements and their associated sensors are configured so that they can be detached [13]. The various standard modules established are common to all subsystems on an integrated basis. The data processing is performed by common processors interconnected by a global bus [3]. All this produces a partially integrated system but it is closer to a fully integrated system than what the evolutionary approach can produce.

Advantages

The main advantages of this approach are as follows:

- The core elements can be operated in a stand-alone mode [13].
- The core elements contain a minimal interface to mission peculiar hardware and software [13].

- Mission peculiar hardware and software can be developed with little concern for the avionic core elements except at the mission/core interface [13].
- It provides the opportunity to easily share information between subsystems, and
- Reduces cost due to proliferation and non-standardization [19].

Disadvantages

The main disadvantage of this approach is:

- The system hardware and software for all missions cannot be checked out as a single package [13]. This implies that when it operates as a single package, it can exhibit certain behaviors that can hamper the performance of the system and these behaviors cannot be observed and corrected at test time.

Applicability

This integration approach can be applied to every class of architecture. However, for very complex distributed or hierarchical systems, this approach cannot cope satisfactorily with every aspect of the integration problem.

This approach is best achieved through a top-down structured design methodology.

2.2.4 Total Integration

Total Integration is a concept whereby all interactions in the system are considered in the design phase of the system. The integration is extended in a system hierarchy network considering all interactions in the network to arrive at a concept of total integration [40 p4].

Characteristics

This approach examines the processing capacity and configuration required to satisfy the combination of all of the mission dependent and independent processing requirements [13]. The process is complete when all significant interactions between the subsystems have been identified, and suitable compromises have been made along the interaction links [40 p4].

This integration approach is a means by which all the necessary rather than essential information paths can be provided [26]. It must be remembered that this integration approach is defined in theory only. It has never been achieved practically but it is a goal to be attained.

Advantages

The advantages of this integration approach are as follows:

- Software for any one mission type can be activated by setting a mode switch in the flight deck [13].
- Software checkout and integration tasks are easier to perform for a full system that does not change software [13], and

- An optimum use of processing element resources and their inter-communication can be realized using the full set of integration software [13].

Disadvantages

The main disadvantages are as follows:

- More equipment is used than is necessary for any one mission [13], and

In practice, a totally integrated system would be open-ended and there would probably be no way to implement the system [13].

Applicability

This integration approach can be applied to every class of architecture.

This approach can only be achieved through a top-down structured design methodology. However, even with the proper design methodology, it may not be possible to achieve a totally integrated system. This problem will be addressed in chapters 4 and 5.

2.2.5 Summary

The major integration approaches practised throughout the evolution of avionic systems are: the Integration by Accretion, Integration by Evolution, and Planned Integration approaches. Some of these were seen to

be quite broad in application.

The reason for such a broad scope, is the lack of well defined aims and procedures. These approaches were not applied identically throughout industry, every organization went about it the way they deemed best. Such unorganized integration attempts are not conducive to solving the integration problem. Everyone is on his own trying to solve a problem that is larger than what he can handle. What is needed is a standard integration approach that would have every organization working in the same direction to achieve the same goals. The Total Integration approach has well defined aims and procedures such that it has the potential to become a standard integration approach.

2.3 DESIGN METHODOLOGIES

2.3.0 Introduction

This section identifies the various design methodologies that have emerged in the literature as being widely practiced by the avionics community at some point in time since the early 60's. These methodologies will be described along the lines of their characteristics, advantages, and disadvantages. Their applicability will be given with relation to architectural classes and integration approaches. Before going any further, let's define what is understood by a design methodology:

"It is an algorithm that is executed by a designer, based on a set of input data, which results in a system description in a form suitable for guiding the detailed physical implementation"[4]

In avionic system engineering, only two major classes of methodologies have emerged: Bottom-Up Design and Top-Down Design.

2.3.1 Bottom-Up Design

This design methodology performs successive composition of the desired system. It assumes that a set of basic implementation elements exists and, based on a set of requirements, combines these elements together to achieve a system that will satisfy the given requirements.

The system design task can be summed up as follows:

"Produce compatible interfaces between one subsystem and another and attempt to produce, with these collections of subsystems, a final product that approximates to the original requirements of the customer and intentions of the designer"[27]

Therefore, the total aircraft avionics end up as a very loosely coupled collection of separate subsystems rather than forming a totally unified system [27,p3.4].

Characteristics

As its name indicates, the direction of design of this methodology is from the bottom up. In other words, the system is developed as an amalgamation of independent black boxes. However, the physical partitioning of the system is usually in accordance with the functional partitioning of the mission [28].

This methodology can only produce partially integrated systems because the emphasis of design is on subsystems or black-boxes as opposed to a total system. However, the extent of integration depends on the methodology used.

Clearly, this methodology is hardware oriented. The designer first defines the hardware and then he worries about the software required to make the avionic suite work as a system. The reason for this is that the software has traditionally been considered to be very flexible as hardware was not. Therefore, the designers concentrated their efforts in designing hardware assuming that the software would adapt easily to the hardware. Finally, it is evident that this methodology is poorly structured. One could even say that it is an unstructured design methodology. First, design the hardware then, try to put it together as best as possible and hope that a workable system will result.

Advantages

The only advantage of this methodology is its relative simplicity. However, one must not confuse simplicity of concept with simplicity of implementation.

Disadvantages

The disadvantages of the Bottom-Up Design methodology are as follows:

- It provides a system unique to a particular aircraft [28].

- It has limited adaptability to changing threat environments or expanding mission roles [28], and
- It rapidly becomes very complex, costly, and difficult to manage as the complexity of the system to be designed increases. Thus, simplicity of concept did not translate in simplicity of implementation. In fact, it is quite the opposite.

Applicability

This methodology can be applied to centralized, federated, or distributed architectures. However, it will have more success with non-complex centralized architectures.

As far as integration is concerned, this design methodology is closely related to the integration by accretion concept. As a matter of fact, one could almost say that it is a design by accretion methodology. Therefore, it is not well suited for any other integration approach.

2.3.2 Top-Down Design

This design methodology performs successive decomposition of the desired system. It starts with a set of functional requirements and then proceeds to break these down into successively lower levels of the system, designing each current level before attempting the next lower one.

The general outline of the methodology is as follows:

- Start the process with the specification of the avionic system

requirements [6].

- Break down the system in a set of functional algorithms [6].
- Partition the system functional algorithms into elemental program units [32].
- Compile the interdependency and interface structure of the elemental program units taking into account the design constraints of the system. This will allow the assessment of the implementation media and the most economical networking of resources [32].
- Simplify the interdependency and interface structure by iterations of the decomposition process [32].
- Now, the requirements can be checked. If the result is positive, the designer can move on. If the result is negative, further iterations of the decomposition process are required [6].
- Assign each program element to an appropriate implementation in hardware, software, or firmware in accordance with a criterion [32].
- Construct a processing network using the structured data flow and control requirements of the decomposed functions [32], and
- Repeat this procedure at successively lower system levels until the lowest possible level is attained [6].

Characteristics

This methodology is systems oriented. The designer treats the avionics suite as a system which is then decomposed in the functional subsystems required to fulfill the system's requirements. This procedure is done in a Top-Down fashion, from the system level (top) down to the component level (bottom).

The methodology in itself is neither hardware nor software oriented. It all depends on the criterion used to implement the program elements. Clearly, it has the potential to tailor the system to the requirements and/or constraints of the software as well as the hardware elements.

Finally, it is evident that this methodology is highly structured. It is broken down into specific steps which, if followed, will lead to a design satisfying the requirement specification of the avionic system. Furthermore, the system will produce a hierarchical description of the avionic system [6].

Advantages

The main advantages of the Top-Down methodology are as follows:

- The functional implementation and cost trade-offs are put to more advantage [32].
- Ease of modification and debugging [32].
- Simplified structured programming [32].

- Reduced program complexity due to hardware instructions [32].
- Increased reliability [32].
- The process involves technology available in the 1975-1985 time-frame [32].
- Independent thinking designers are forced at the onset to admit dependence to a benefit and not to a hindering [32], and (check on this)
- It offers the strongest and most realistic approach to uniting the advantages of LSI technology, language interfaces, and distributed resource techniques [32].

Disadvantages

The main disadvantages of this methodology have not been determined yet. Top-Down design is just starting to be applied such that the experience gained so far is not sufficient to isolate the major problems of the methodology.

Applicability

This methodology is applicable to all classes of architectures. It is conceptually transparent to the architecture you are trying to produce. It is a matter of selecting the proper requirements and design criteria to produce the desired architecture. However, it is particularly well suited for hierarchical architectures.

As far as integration is concerned, this methodology will theoretically support any integration concept.

2.3.3 Summary

The major design methodologies that were practiced by the avionics community since the early 60's are: the Bottom-Up and Top-Down methodologies. These two methodologies were seen to be diametrically opposed. The Bottom-Up method performs successive composition, whereas the Top-Down one performs successive decomposition. It is interesting to note that the evolution from one to the other happened without reverting to a methodology situated somewhere in between. The reason for this state of affairs will be analyzed in Chapter 3.

2.4 CHAPTER SUMMARY

The role of this chapter has been to identify and present the avionic architectures, integration approaches, and design methodologies that have played an important role in the evolution of avionic systems for the past two decades.

This was done to establish the definitions and a description of the building blocks that will be used in the next chapter to trace the evolution of avionic systems. The classes of architectures that were presented are as follows: A Centralized system where a central unit manages, controls, and provides all the processing required of the subsystems. A Federated system which is mainly characterized by the functional indepen-

dence of its subsystems. A Distributed system where the management, control, and resources are distributed throughout the partitioned system. And finally, a Hierarchical system which is an array of intelligent subsystems under the control of an upper echelon complex of processors which coordinates the operations of all the system elements.

The integration approaches presented are the following: Integration by Accretion in which the avionic functions are added in the aircraft where space permits; Integration by Evolution which combines together functions with natural affinities to increase the interactions between them; Planned Integration which divides the avionic architectures in core subsystems; and finally, Total Integration which identifies all the interactions between the subsystems and finds suitable compromises to them.

The design methodologies presented are the Bottom-Up methodology, which tries to produce a system through an unstructured amalgamation of independently defined subsystems, and the Top-Down methodology which starts at the top of the system and, through a structured process, successively designs lower levels of the system.

All of these, then, constitute the foundation for chapter 3.

CHAPTER 3 EVOLUTION - CAUSES AND EFFECTS

3.0 INTRODUCTION

To be able to project what the next generation avionic system should be, one must have a good understanding of the evolution of avionic systems up to present times. The understanding of this evolution will come from the factors that brought about every evolutionary step, the impact of each step on its times, and the impact of each step on the next one.

This chapter describes the evolution of avionic systems engineering from the 60's to the present. The main factors affecting this evolution are the technology, requirements, and constraints imposed on avionic systems. Other factors will be considered depending on which evolution we are describing. The milestones used for this description are the various architectural classes, integration approaches, and design methodologies presented in Chapter 2.

Section 3.1 describes the evolutionary factors common to all avionic systems. Section 3.2 presents the evolution of avionic system architectures. Section 3.3 discusses the evolution of integration approaches and Section 3.4 traces the evolution of the design methodologies.

3.1 AVIONIC SYSTEMS EVOLUTION

3.1.0 Introduction

This section describes the factors influencing the evolution of avionic systems in general. It is important to understand the impact of these factors because they will influence the evolution of avionic system architectures, integration approaches, and design methodologies as described in later Sections.

Section 3.1.1 analyses the interplay of the enemy threat and the technology as it impacts the requirements imposed on both technology and mission capabilities. Section 3.1.2 presents the evolution of technology since the early 60's. Section 3.1.3 describes the evolution of the requirements imposed on avionic systems as a whole. Finally, Section 3.1.4 defines the "integration problem" that has become a crucial issue to resolve.

3.1.1 Threat versus Technology

The "raison d'être" of military avionics is to provide adequate mission capabilities to the military aircraft fleet such that it will be able to counter a perceived enemy threat. The important words here are "perceived enemy threat" which really means: what top level strategists think the enemy can achieve technologically. In such an assessment, the strategists cannot afford to be conservative. Underestimating the enemy's capabilities would spell defeat.

Electronic technology is the most important factor influencing avionic capabilities. Technology translates into avionic performance requirements as follows (see fig. 3.1) [33]: The technology determines the enemy, threat and the appropriate countermeasures. These, coupled with design capabilities, will determine the mission scenarios of the aircraft. Avionic functional requirements are drawn from these scenarios and translated in avionic performance requirements. This translation process is weighted by mission independent constraints such as cost and availability. Performance deficiencies are determined by the comparison of performance requirements with the performance evaluation. If this comparison shows the performance to be deficient, then, technology, system design, and component design will have to share the burden of increasing the performance. The technology supports most of the load.

It is logical to assume that the enemy will constantly try to increase his capabilities in order to get ahead performance wise. It is also logical to assume that whatever the level of performance attained by one side, the other side either has already attained such a level or will soon do so. Thus, avionic performance requirements are always increasing. This in turn means that technology is continually being pushed to its limits to fulfill the requirements. The situation is complicated by the fact that when a requirement is filled by the generation of a new capability, this new capability often has applications beyond its original intent [3]. These new uses generate new requirements which are filled by still new capabilities and so on [3]. These interactions form a closed

Figure 3.1 Influence of Technology on Avionic Performance Requirements

loop [3]. Thus, the requirements to counter a perceived threat create demands on technology which in turn creates new requirements which push technology development even harder. This process goes on indefinitely because both sides are trying to get ahead of one another and technology as their primary means of achievement.

3.1.2 Technology Evolution

The first generation of electronic computers was introduced around 1950 [23]. It was the vacuum tube invented some 40 years before that allowed the computers to transition from the analog to the digital world [23]. However, these computers were enormous and consumed large amounts of energy. It was the transistor (1956) that really opened up the world to computers. It allowed significant reductions in size, weight, power consumption, and price. The introduction of the integrated circuit in 1962 allowed a further reduction of these factors. For example, prior to the integrated circuit, it took 12 independent discrete components to make a single gate whereas two to four gate chips were common in 1964 [23]. Such technology of less than 10 gates per chips was termed small scale integration (SSI). It was soon to be followed by medium scale integration (MSI, less than 100 gates per chip) in 1966 [23]. By 1969 100 gate chips were off-the-shelf [23].

Technological improvements were far from over. MSI was followed by large scale integration (LSI, from 100 to 10,000 gates per chip) in 1970 when LSI circuits began to be produced in large quantities for computer

main memories and pocket calculators [24]. Already tremendous technological improvements had taken place. The ten year period from 1962 to 1972 saw the following computer improvements [23]:

- A cost reduction of 20 to 1.
- An airborne reliability increase of 450 to 1.
- A size reduction of 20 to 1.
- A weight reduction of 12 to 1.
- A power reduction of 10 to 1, and
- A speed increase of 40 to 1.

All of this resulted in an effective instruction per dollar improvement of 800 to 1 [23].

The major impact of the LSI technology is that it made possible the fabrication of a central processing unit (CPU) or even an entire computer (excluding I/O devices) on a single integrated circuit (IC) [24]. A CPU or similar programmable processor on a single IC is called a microprocessor. A computer on a single IC is called a microcomputer. The first commercially available microprocessor was the Intel Corporation's 4004. It appeared in 1971 and used MSI and LSI technology [24]. The one-chip microcomputer is considered a landmark development in LSI technology [24]. Typical of such development is the Intel 8748 microcomputer introduced in 1976 [24]. Computers were now small, light, and cheap enough to

be used wherever they were needed (i.e. sensors). Technology improvements continued at a breathtaking rate such that very large scale integration (VLSI, more than 10,000 gates per chip) appeared around 1979.

Throughout the 1970's, there has been a lag-time of several years between the introduction of a technology base in commercial market and its appearance in military systems. The reasons for this are the length of military development programs (8-10 years), the rapid rate of change of technology, and most importantly, the small market impact of military systems. With military IC sales making up only seven or eight percent of the total market, IC manufacturers are reluctant to invest in the costly militarization of their product when there exists a large commercial market for the unmilitarized version [7]. As a result, it was, in 1979, taking up to three years for the transition of a product from the commercial market to military systems, if a transition does occur [7]. In an effort to hasten the introduction of new technology products in military systems, DoD introduced the very high speed integrated circuit (VHSIC) program in 1978. The main objectives were to get ESI and VLSI technology into DoD weapon systems, particularly in high speed signal processing applications, by the mid 1980's [7]. The unvoiced objective was to advance the state-of-the-art in the semiconductor industry for specific military applications [7]. This is illustrated by the requirement to produce ICs designed around sub-micron rules (.5-.8 micron) and to achieve speeds on the order of 10 to the 13 gates per second [22]. This VHSIC program will have significant impact on the future development of avionic systems.

A good illustration of the progress of technology is given by figures 3.2 and 3.3. Figure 3.2 shows the decrease of the speed power product (switching speed times power consumed) over the years 1968 to 1978 with projections through 1986. Figure 3.3 shows the increase of logic gate density from 1970 to 1978 with predictions through 1986. The X on the figures illustrates the 1982 state-of-the-art. It can be seen that the projections underestimated the actual evolution of technology. This stems from the fact that the projections were based on the reduction of device sizes without taking into account improvements in design and implementation [5]. The important point of these two figures is to show that technology has not stopped its trend toward smaller, faster, and cheaper computers, and that the progress has been continuous as opposed to sudden jumps in improvements.

Since 1979, the technology has entered the VLSI era and is progressing along an exponential curve. It is projected [5] that VLSI will continue its exponential growth until 1992. Since 1982, the exponential climb is in the vertical part of the curve [5]. This situation combined with the VHSIC program will ensure that military system designers of the next generation avionic systems and beyond will have the smallest, fastest, and cheapest ICs at their disposal.

3.1.3 Requirements Evolution

The main requirement imposed on avionic systems is that of increased mission capabilities at the lowest possible cost consistent with

Figure 3.2 Speed Power Product versus Time

Figure 3.3 Past and Predicted Logic Density

available resources [8]. As pointed out in Section 3.1.1, this requirement is always present and will remain present as long as one side poses a threat for the other.

It is essential to economically maximize the effectiveness of tactical aircraft avionics [8]. This need points to the following requirements [8]:

- Lethality,
- Survivability,
- Availability, and
- Affordability.

The lethality and survivability requirements can be filled by the introduction of evermore sophisticated subsystems performing functions such as weapons delivery, target recognition and acquisition, threat warning/location, countermeasures, and others [8]. The improvement of the availability implies increased reliability and/or maintainability of the avionic system [8]. The affordability requirement results from the fact that increased capability costs money and raises the acquisition cost of aircraft [8]. The affordability of an avionic system will be realized if the increased acquisition cost is matched by a decrease in operation and support costs resulting from improved reliability and maintainability [8].

These requirements translate in lower level requirements of improvements in performance, speed, throughput, and flexibility as well as reductions in equipment size, weight, power and cooling, and crew workload [11].

The above discussion deals with requirements that are present throughout the evolution of avionic systems and thus will be implied in later Sections. There are, however, requirements that have emerged at some point in time in the evolution of avionic systems and have stayed present ever since. These requirements are discussed below.

Prior to the mid-60's increased mission capabilities coincided with new weapon system developments. The technology changed slow enough that when new capabilities appeared, it was also time to replace the existing weapon system. In the late 60's technology was starting to change quite fast. With the procurement cycle of a weapon system taking from 8 to 10 years, and the life of the fielded weapon system being 10 to 20 years; the technology incorporated in the weapon system became obsolete well before the end of the service life of the system [11], [22]. The technological changes translated in new capabilities. These new capabilities implied new mission requirements since it had to be assumed that the enemy also had these new capabilities. When the new requirements have evolved beyond the point where the original system can be considered effective, a new weapon system must be designed. However, avionic development and procurement programs are high cost programs (several billion dollars). This fact, coupled with the rapid rate of change of requirements, made

the solution of an update/upgrade of existing systems the only possible way to meet the new requirements cost-effectively. The success of such updates depends on the possibility of inserting new technology products in the regular life of the weapon system without major impacts on the system and at an acceptable cost. This is only achievable through technology independence of the basic system design, at the level of insertion of technology [11], [22].

The constant demand for increased mission capabilities resulted in a steady and rapid increase in digital processing requirements [10]. This is exemplified by the Jaguar (1960's) and Toronado (1970's) programs which showed an increase of 300% in digital processing requirements between them [10]. With the continued technology progress and hence the increase in mission capabilities, digital processing requirements have grown steadily and will likely continue to do so in the future.

Increased mission capabilities account for yet another requirement, that of increased functional densities. In the 70's, it was envisaged that weapon systems would be required to perform well beyond the existing levels of avionic capabilities and would include many more functions than contained in existing systems (1977) [1]. This problem worsened in the late 70's and early 80's because of the decreasing numbers of aircraft in typical arsenals which, coupled with the normal increase in enemy threat meant that each aircraft had to be doubly capable in mission effectiveness and the protection of its crew [3]. This situation is still prevalent and will be present for the next generation avionic system.

3.1.4 The "Integration Problem"

The "integration problem" is a situation whereby the integration phase of avionic system development programs has become unacceptably complicated, lengthy, and costly. This situation has resulted from the fact that, on the one hand, the integration approaches currently practiced are not suitable for the systems being designed, and, on the other hand, the integration phase is too late in the development program structure. The latter is the major contributor to the "integration problem".

The currently practiced program development structure is illustrated at figure 3.3. This figure points out that integration is attempted at test time, after the system has been designed and built. This is a highly inefficient procedure because the system has not been designed with integration in mind therefore, at integration time, problems invariably develop because the subsystems will not work together as a unified system unless a significant amount of time and money is spent developing interfacing software. Even then, problems might still exist such that hardware changes are required. This implies re-designing the system. Hence, the integration phase becomes a lengthy and costly process. Obviously, the more complicated the system, the more complex and costly the integration phase. Such integration is nothing more than an attempt at bottom-up design, a process through which a set of previously designed hardware and software components are grouped together to form a total, unified, homogeneous and working system. As Chapter 3 will point out, bottom-up design only results in a costly and unsatisfactory system.

Figure 3.4 Current Avionic System Development Program Structure.

With constantly increasing operational requirements, next generation avionic systems will be more sophisticated and more complex than ever, thus making the integration problem even worse.

Integration problems also appear in the design of updates/upgrades of avionic systems. Such updates/upgrades are, by definition, a bottom-up design exercise. For the same reasons as before, the integration of these updates/upgrades is complex, lengthy, and costly. However, in this case, the integration phase is necessarily after the design phase since the system is already operational.

3.2 AVIONICS ARCHITECTURE

3.2.0 Introduction

This section describes the evolution of avionic system architectures and outlines the reasons that brought about this evolution. This description spans from the early 60's to the present. The architecture of the early 60's, the pre-centralized architecture, has been in existence since the introduction of avionics in aircraft. The starting point of the evolutionary description is the transition from the pre-centralized to the centralized architecture. It is at this point that the digital computer was introduced in aircraft leading to the development of digital avionic system architectures.

Section 3.2.1 describes the transition from the Pre-Centralized to the Centralized architecture. The salient point of this evolution is a

shift from analog to digital technology. Section 3.2.2 discusses the transition from Centralized to Federated architectures and Section 3.2.3 presents the transition from Federated to Distributed architectures.

3.2.1 Pre-Centralized to Centralized

The pre-1960 avionic system was composed of several independent self-supporting subsystems (black boxes) which were being procured and installed with very little intra-system data flow [32]. The pre-centralized architecture was merely a very loosely coupled grouping of these independent subsystems [27]. This grouping of subsystems did not form a system per se. Since the aircraft was of relatively low performance, the pilot served as a central processor and communication link between the subsystems [16]. It is through the pilot that the independent subsystems formed a system.

Technology

The technology that existed prior to the introduction of centralized architectures is one which was transitioning from the analog to the digital world. The advent of the transistor and SSI, along with the ensuing reductions in weight, size, and cost meant that general purpose digital computers could be considered for aircraft implementation where they could be used to carry out some of the computation necessary within the existing avionic subsystems [27].

Constraints

Prior to MSI and LSI, the cost of technology was quite high. Digital computers were in their infancy and were very expensive both to buy and to operate. In particular, CPU time and memory were expensive enough to be a constraint to the designer. Thus, efficient use had to be made of both CPU time and memory space [2]. Furthermore, even though the transistor and integrated circuits allowed significant savings in weight and volume, early digital computers were still quite bulky and heavy. Since an aircraft is always constrained in terms of space and weight, the system designer had to minimize both the size and weight of the avionic components.

Requirements

The requirement for increased mission capabilities had to be fulfilled. Since the crew spent most of their time controlling the airplane, they had less time to perform complex system management and decision making functions [16]. The fulfillment of the above requirement meant reducing the pilot's task associated with flying the aircraft.

Bottom Line

The digital computer used as a central processing unit was seen to be the answer to the problems faced by the avionics designers of the time. The basic concept was that the economics of the general purpose computers favored their use in solving large, complex problems rather

than small ones [9]. Thus it was more economical to centralize all the processing required in the aircraft. Besides, because of their cost and size, general purpose computers could not be added to any subsystems that needed to carry out a calculation [27]. The centralized approach made sense since it minimized hardware which means less weight and space taken by the system [32]. Furthermore the crew's task associated with flying the aircraft could be significantly reduced by allowing the central computer to perform some of the subsystem to subsystem tasks [16].

The efficiency of the digital computer as a switching center for intercommunication made the central computer even more attractive [21], [2]. Finally, a central computer unit makes efficient use of CPU time and memory space [2]. As can be seen, the central computer concept appeared to be the answer needed at that time.

3.2.2 Centralized to Federated

Technology

With the advent of MSI and LSI, it became affordable to incorporate processors in the various subsystems composing the avionic system [2], [27]. Furthermore, the processing capabilities of the airborne computers increased significantly. Nevertheless, processing capabilities had not yet been incorporated in the sensors themselves.

Constraints

While the cost of the technology had gone down it wasn't so low as to be insignificant. Cost remained a constraint on the system designer. Size and weight were still constraints also but with the reductions offered by IC technology, much more could be accomplished for less size and weight.

While the use of digital computers allowed the introduction of many new operational capabilities, management also had to live with costly highly complex, and in many instances, inefficient use of the computer as an operational resource due to the late 60's lack of quality software development and support tools [42.p 4-1]. What this means is that the increased capabilities provided by hardware developments translated into software complexity.

The problem with software development and support stems from the fact that software was written in assembly code. In the beginning of airborne digital computing, the most important criterion for the development of software was its efficiency [11]. The hardware was bulky, expensive, and slow in its operation; it was then necessary to squeeze as much as possible in the smallest space, and to code it as tightly as possible so that it ran with no inefficiency in time [11.p6]. This is the software complexity discussed above. Such software complexity inevitably implied high costs in development and support. This was the start of the software problem of the centralized approach which would eventually sound

the end of the concept of centrally processed systems [11].

Requirements

The requirements for ease of growth and technology transparency started to appear in the late 60's. The designer now had to take them into account.

Bottom Line

The Centralized system soon ran into problems because the sensors were designed with no concern for interconnections with other systems [13] [27]. Each sensor was specialized and had its own interface. The majority of avionic sensors were designed to provide data in an analog form most convenient to the sensor manufacturer or peculiar to one-time application [19,p83]. Since the central unit had the responsibility of handling all of the incoming and outgoing data, it was often lost in the size and complexity of the I/O [13]. A large part of the centralized computer task was taken up by performing analog to digital and digital to analog conversions and in performing digitally the necessary unit conversions [27,p3-3]. Furthermore, the centralized system had no amount of technology transparency.

Clearly, something had to be done about these problems and the solution was to partition and distribute the I/O circuitry along with some processing capabilities thus alleviating some of the central unit's complexity [13]. Fortunately, multiplexing existed such that it was possi-

ble to implement the extended computer I/O philosophy as it was called [13]. However, multiplexing alone was not sufficient, the subsystems required some processing capabilities. But this was already taking place. Processors had now (early 70's) become sufficiently light-weight and low cost that embedded computers were a reality [42]. Thus was born the multi-processor avionic architecture [13]. This multi-processor architecture rapidly evolved in what is known as a Federated architecture [42].

Although the central unit's complexity had been reduced, software still remained a problem. There are two main reasons for this. First, the concept of central management implies large software programs controlling evermore complex systems. Secondly, software was still developed in assembly code with efficiency as the sole criterion [11].

3.2.3 Federated to Distributed

Technology

Pre-LSI computers were large, heavy, power hungry, and very costly such that the avionic system designer had to squeeze every ounce of performance out of a given computer [11]. With LSI processors, the concept of inefficient use of the CPU was now possible [2]. Thus, processors could be distributed throughout the system without imposing such a severe space, weight, or cost penalty to the system.

With the introduction of VLSI and the microcomputer, the avionics industry had at their disposal computers that were smaller, lighter, cheaper, and more powerful than ever before. The remaining barriers to the distribution of processing resources had been literally dissolved.

Constraints

Suddenly, with LSI and VLSI, constraints experienced in previous avionic systems had now disappeared. Weight, space, power, and cost of hardware were not a problem any more. However, what was growing into a big problem was the complexity of the software systems. Central management of a system, as in Centralized and Federated architectures, becomes increasingly difficult and costly as the capabilities and requirements imposed on the system increase. The trends in avionic systems being toward an increase of the number of functions implemented in digital processors, avionic software complexity was continually increasing [38]. This complexity is a major constraint for the avionic system designer. Something had to be done to make the software easier to produce and manage.

Requirements

Although digital processing requirements were steadily increasing, system data flow requirements could not be allowed to grow in proportion [10]. Thus designers now needed to minimize total system data flow bandwidth.

The requirements for ease of growth and technology transparency were becoming more and more important.

Bottom Line

The technological improvements of the 70's had several implications. The microcomputers were now capable of meeting 95% of the speed requirements of avionic systems [32]. As early as 1977, information transfer system components were becoming available in adequate quantities and cost to allow distributed architectures to become cost-competitive with centralized processing and dedicated subsystems [13]. Direct, all-to-all, communication between processors was now economically implementable [6]. All of this made possible and actually accelerated the trend toward Distributed architectures with functional partitioned and modular avionics and control [38]. All that was needed now was the motivation to move in that direction.

The first motivator was that the designer had to minimize the total system data flow requirements. Centralized and Federated systems, requiring extensive intersystem communications, only contribute to the problem. The most effective way to reduce data flow is to process raw data at the source and transmit only the processed data between subsystems [10]. This implies a functional distribution of processing [10].

The second motivator was to reduce the system complexity. The obvious way to alleviate this problem was to divide the hardware and software of the system into more manageable modules [10]. Again, this points to-

ward the distribution of processing and control.

The next motivator was to introduce technology independence into the system. This requirement for technology independence implies structuring the system to provide processing and memory resources through horizontal system partitioning [22]. This favors the implementation of a Distributed architecture [22].

Finally, a Distributed architecture exhibits more reliability, maintainability, and survivability while being less vulnerable than the Centralized and Federated architectures. Thus, the logical answer was to move to the distributed architectures.

3.2.4 State-Of-The-Art Architecture

The Distributed architecture is described in Chapter 2. It represents a necessary and logical evolution from the Centralized and Federated architectures, and has been widely accepted by industry. However, this architecture, like any other, is not without its problems.

The first problem is one of control. Due to the nature of this architecture, each part of the system must operate without an instantaneous knowledge of the state of all other parts [6]. The system can easily live with this fact for most of the time. However, it creates problems for the modification of control as a result of changing mission priorities or system health [33]. In this case, complete instantaneous knowledge of the state of the system is a prerequisite for a proper

analysis of the situation and the subsequent determination of the resulting state to be implemented.

The second problem is one of functional isolation. The flight critical components of the system may want to impose some restriction on the transfer of information and who has access to it [22]. Such isolation is not possible with a completely distributed system which is characterized by a single level bus architecture connecting all the components on this one bus.

As can be seen, there are some problems with the Distributed architecture and these problems will have to be solved for the next generation avionic system. Furthermore, these identified problems are not all encompassing, others may exist. Chapter 4 will deal with this situation in more detail.

3.2.5 Summary

The evolution of avionic systems architecture started with the Pre-Centralized systems and changed to the Centralized, Federated, and Distributed architectures. This evolution was both consistent and stable. The architectures were continually evolving and their evolution suffered no significant breaks or departures from the trends previously established. Every step was evolutionary as opposed to revolutionary. The reason for such consistency is that the factors influencing avionic architectures also exhibited a consistent evolution. Technology continually moved toward smaller, cheaper, and more powerful solid state com-

ponents, the rate of change of this technology was consistently rapid, and weapon systems continually had to show increases in capabilities and performance at a reduced life cycle cost.

The present state-of-the-art avionic system architecture is the Distributed architecture. This means that the systems being produced today tend to have such an architecture. However, since technology is still evolving and requirements imposed on weapon systems are still increasing, it is not evident that the Distributed architecture will be applicable to the next generation avionic systems.

3.3 INTEGRATION APPROACH

3.3.0 Introduction

This section describes the evolution of integration approaches and examines the reasons that brought about this evolution. This description will concentrate on the period from the early 60's to the present. Although avionics were being "sort of integrated" in aircraft long before 1960, it was not until the advent of digital computers that the word integration took a meaning all its own. Prior to that, integration was merely a synonym for installation.

Section 3.3.1 describes the evolution of integration concepts that lead to the Integration by Accretion approach. Section 3.3.2 presents the evolution from the Accretion approach to the Integration by Evolution approach while Section 3.3.3 discusses the evolution from the Evolution-

ary to the Planned Integration approach.

3.3.1 Accretion

Technology

The Integration by Accretion approach has been practiced from the introduction of avionics in aircraft to the application of multiplexing to avionic systems. As can be seen, the pre-1960 technology had virtually no impact on the integration approach.

There were several reasons for this lack of impact. First and foremost, the technology was changing at a slow rate. As far as avionics is concerned, the past two decades have seen more technological changes than the period from 1920 to 1960. The second reason was that the Integration by Accretion was easily able to handle the technological changes that occurred in the pre-1970 era. During that time, technological advances consisted of increased black box sophistication [19], [17]. With the avionic system composed of independent black boxes, when a new box was introduced, it could be installed in the aircraft with minimal impact to the other ones already there. The crew was doing whatever integration there was to be done [16], [3].

Architecture

Prior to the early 60's, the avionic system architecture was that of a very loosely coupled grouping of black boxes with a minimum of interaction between them [27]. With such an architecture, the Integration by

Accretion approach was quite adequate. In fact it was the the most cost effective.

Then, in the early 60's, the centralized architecture appeared. The avionic system was now more tightly coupled than before but it was still simple enough to be handled with the Accretion approach. The integration process was a bit more involved but it was still the Accretion approach.

Constraints

The avionic cost and weight constraints were present but were not influenced by the integration approach. There was nothing it could do to reduce the cost and/or weight of the avionic system. Since these were composed of an amalgamation of black boxes, cost and weight could then only be reduced through a price and weight reduction of the black boxes themselves, a situation which was still prevalent with the early centralized systems.

Requirements

There was no integration requirements imposed on avionic systems in their early years other than making sure that each black box worked properly [27]. As airborne computers and centralized systems appeared, the integration requirements evolved to that of providing suitable interfaces between two communicating black boxes [27]. This was easily handled by the Accretion approach as long as the avionic system remained relatively simple.

Bottom Line

From the introduction of avionics in aircraft up to the early centralized systems, the Integration by Accretion approach was adequate to cope with the avionic systems as they existed and were designed. It handled all the requirements and constraints imposed on avionic systems and was supported by the existing architecture.

3.3.2 Accretion to Evolution

Technology

The technological improvements of the 60s confirmed that digital avionics was the way of the future [11]. This improved technology led to cost-effective implementation of new concepts. One of the most important was the introduction of the multiplexed data bus in 1972 [20]. This in turn opened the door to new integration concepts [19].

Architecture

As the technology evolved, avionic system capabilities increased steadily, and systems became more complex. With a complex centralized architecture, the designers were finding it more difficult to apply the Integration by Accretion approach. The interactions between subsystems had grown in amount and complexity such that a gentleman's agreement on interface and data signal definition was not sufficient any more [27].

2

With the advent of the Federated architecture in the early 70s, the problem just got worse. Now, several processors were working simultaneously on closely related tasks and thus the interactions between the processors themselves and with the central manager had grown out of the proportions suitable for the Accretion approach. Something else had to be found. This feeling was reenforced by the fact that technology trends pointed toward even more capable and hence complex avionic systems.

Constraints

The major constraints imposed on avionic systems were to reduce their size, weight, and cost. The Integration by Accretion did nothing to help in this area. It could not influence the size, weight, and cost of the avionic subsystems. Furthermore, as systems became more sophisticated, the integration task grew in complexity and hence in required completion time. This implies increased development costs which is the opposite of the designer's objective. System designers needed a approach that would help alleviate these problems.

Requirements

As early as 1969, it was recognized that avionic integration needed to take place in the beginning of the development cycle [41]. This was later reenforced when it was proposed that the whole avionic development needed to be considered much earlier in the aircraft development cycle and remain in step with the airframe and engines [18]. The main advantage was that it would allow cross-integration planning to occur while

everything was still flexible [18]. A development cycle paced to airframe and engine schedules with the avionic being made up largely of off-the-shelf or low risk components frustrates effective integration of avionic systems to each other and to the weapon system [1]. This situation arises because avionic integration requirements may impact airframe and engine development and require engineering trade-offs to be made. With the airframe and engines design cast in concrete, such trade-offs will not take place and thus the designer may not be able to integrate the avionics as required.

Another requirement is that of standard equipment interfaces and a standard approach to data communication [19]. This would help alleviate the problem of proliferation of incompatible systems from platform to platform. The final requirements are ease of modification and ease of growth.

All these requirements point to the need for a more effective system integration approach.

Bottom Line

In the early '70's, avionic systems were constantly evolving and becoming more complex. This made apparent the fact that the Accretion approach was ill-suited to existing weapon systems because the use of dedicated hardwired interfaces only lead to inflexible and expensive avionic suites [20]. Furthermore, requirements for future modification and growth could only be supported through great pains and costs due to

the lack of standard interfaces.

Because of a lack of a standard integration approach, the equipment introduced in inventory was in low quantity and low reliability [18]. This created a logistic and maintenance problem [18]. It was proposed that part of the solution to the high cost, low reliability, and heavy operation and maintenance costs of black box proliferation was to view the avionics as an integrated system rather than a conglomerate of functional sensors [19]. Thus, something had to be done to promote the integration of avionic systems.

Fortunately, the concept of time-division multiplexing was introduced in the early 70s and industry was quick to endorse it. Soon after, different multiplexed (mux) data buses were starting to proliferate to an extent where it would obliterate the gains made by the introduction of mux data buses [19]. Thus came MIL-STD-1553 (1973) which standardized mux buses and allowed for standard digital interfaces and a standard approach to data communication. It also forced the designer to perform system level analysis [19]. MIL-STD-1553 became a key element in the drive toward a standard integration process for avionic systems [19]. It is also the key element for the new Integration by Evolution approach which is broadly defined as "combining together functions with natural affinities". This Evolution concept replaced the Accretion approach.

The Evolutionary approach was an improvement over the Accretion concept but it did not completely fulfill the requirements imposed on avion-

ic systems. Modifications and growth were still difficult. It did not represent a standard integration approach, avionics were not sufficiently treated as a system, the integration process was still costly in time and money, and integration was still effected at test time, a situation recognized as undesirable.

3.3.3 Evolution to Planned

Technology

IC technology improvements produced small, light weight, and cheap processors. These processors are amenable to distribution and modularization.

Architecture

The Distributed architecture requires more integration than the centralized and Federated architectures. The distribution of processing resources implies a distribution of interactions among the subsystems and sensors which now have some processing power. This means that there are many more players on the field. This, coupled with the fact that the tasks to be handled are more complex and more numerous, means that the integration approach has to handle more interactions which are more sophisticated and not at all obvious. Ways had to be found to make this integration process more cost-effective and more complete.

Constraints

The designer's answer to continually increasing mission capability requirements was to increase the number of functions performed by the avionic system [1], [38]. Such functional densities create a problem of manageability and implementation. All these functions have to be incorporated in the system without significantly increasing the weight, size, power consumption, and cost of the resulting system. Furthermore, these functions have to work in harmony with the existing functions of the system. The more functions are added, the more difficult the system is to manage.

Requirements

As mentioned in Section 3.3.2, many requirements that lead to the Evolutionary approach still existed since they were only partially fulfilled. Thus, they were still pushing toward more integration. The requirement for integration earlier in the development process not only remained but was generally re-enforced. There is a consensus in the avionics community that effective "integration" can improve mission effectiveness and operational readiness and reduce costs to an extent not possible in the approach in which integration occurs after the development and local suboptimization of relatively independent avionic subsystems [3,p1].

Finally, it was noted, in 1977, that integrated development and procurement were not practiced [1]. The technology existed to achieve the

integration required but lacked a development process, a management structure and an acquisition approach to achieve this integration [1].

Bottom Line

The drive toward more integration had not stopped. It started when the Accretion approach became more of a hindering than a help and then moved to the Evolutionary approach with the standard mux data bus. But MIL-STD-1553 did not insure integration of subsystems through general procurement practices [1]. More was still needed, specifically in areas of procurement, development, and management. Furthermore, it was hoped that advancing technologies would answer the problem of the required increased functional densities [1]. However, these advanced technologies could only provide a partial solution to functional overpopulation and high performance requirements [1]. What was needed to complete the solution was carefully planned and well orchestrated systems integration [1]. What this meant was that an integration approach was needed that was more structured, more clearly defined, and that would promote integration in the above mentioned areas.

The solution proposed was what is called Planned Integration [3]. It introduced further concepts of standardization with standard computer languages and standard instruction set architectures (MIL-STDs 1589 and 1750) in the late 70s. It also promoted modularity and commonality through the concept of core subsystems and transferable software. All this also forced more integration concepts in the procurement and

development practices. Now, modular and standardized subsystems would be developed and procured.

3.3.4 State-of-the-Art Integration

The Planned approach is an improvement over the Evolution approach. Through modularity, commonality, and standardization it enhances modification and growth and treats the avionics more as a system than before. It also provides more structure to the integration task making it easier and less costly. However, it does not represent a standard integration approach nor does it satisfy the requirement of integration earlier in the development process. Thus, this approach is still not the answer to all the integration problems encountered in the design of new avionic systems. It is merely an improvement over the Evolutionary approach.

Even though the state-of-the-art integration is the Planned approach, the Evolution approach is still present. Industry doesn't necessarily apply one or the other but rather a mixture of both and sometimes a bit of the Accretion approach. The truth is that there is still no single clear cut, well defined integration approach accepted by industry as a whole. Therefore, some of the problems of the Accretion and Evolutionary approach are still present. Efforts are directed toward the establishment of a standard integration approach but the road to industry-wide acceptance is long and difficult. There is a drive toward more system integration [3] and this may lead to a standardized integration approach.

3.3.5 Summary

The evolution of integration approaches was not as consistent as the evolution of architectures. Integration by Accretion dominated the initial introduction of avionics systems into aircraft and continued to be practiced beyond the introduction of digital technology and Centralized architectures. This can be explained by the fact that technology and requirements could still be cost-effectively accommodated by this approach. It was not until the Federated architectures, the concept of a digital multiplexed bus, and the rapid rate of change of technology that the integration problem became severe enough to warrant a change in integration approach. The integration concept changed to the Evolution approach. The need for cost savings, the emergence of new technologies, and increased weapon system capabilities requirements were the main reasons for the change. The Evolution approach was followed by the Planned Integration approach for the same reasons that brought the Evolution concept. The evolution of integration approaches is now closely tied to the evolution of technology and weapon systems requirements.

The present state-of-the-art in integration approaches is the Planned Integration approach. It is currently being incorporated into several development programs. However, it is still not the answer to the "integration problem". This, coupled with the fact that technology is still changing at a fast rate and that requirements are increasing, raises the question about the applicability of the Planned Integration approach for the next generation avionic system.

3.4 DESIGN METHODOLOGY

3.4.0 Introduction

This section describes the evolution of system design methodologies and the reasons that brought about this evolution. As with the previous Sections, this discussion begins with the introduction of digital systems in aircraft. It is obvious that some design methodology was used prior to that time but only one methodology was used from the introduction of avionics into aircraft until the advent of Distributed architectures.

Section 3.4.1 presents the evolution of design methodologies that lead to the Bottom-Up approach. Section 3.4.2 describes the evolution from the Bottom-Up to the Top-Down methodology.

3.4.1 Bottom-Up

Technology

The Bottom-Up design methodology has been in existence ever since the first avionics were introduced in an aircraft. In this sense, it was immune to technological changes just as the Integration by Accretion approach and for the same reasons.

The pre-1970 design concept was to design the aircraft along the aeronautical requirements then install the most readily available electronic black boxes in the aircraft [41]. There was no concept of an avionic system as such, just an amalgamation of evermore sophisticated

black boxes each doing its own thing.

Architecture

Up to the early 60's the avionic system architecture was that of a very loosely coupled grouping of black boxes with a minimum of interaction between them [27]. The Bottom-Up design methodology could easily handle that kind of a system.

Then, computers appeared in aircraft and the architecture became centralized. The subsystems were a bit more tightly coupled than before and the design procedure was more involved [27], but the Bottom-Up methodology was still used for the design of such systems.

Integration

For the period up to the 60's, there wasn't much integration being performed. Off-the-shelf avionics were simply installed in the aircraft where space could be found. During those years, there was no real distinction between design methodologies and integration approaches. Whatever integration needed to be done was taken care of by the design methodology. Both design and integration were concerned with the particular interfaces that affected the individual black box operation [27].

As technology and architecture evolved, integration became a problem. For reasons stated in Section 3.3, the mid-70s avionic system was now viewed as an integrated system and integration goals were moving toward standardization of interfaces and data communication with the intro-

duction of TDM data buses and MIL-STD-1553. However the design methodology remained the same [27] [17].

Constraints

The main constraints imposed on early avionic system design were that of size and weight. There was limited room for avionics and they had to stay within weight limits. The Bottom-Up approach was more than sufficient to deal with this.

Even as the technology evolved, the situation remained relatively the same. The majority of the cost of the avionic system was with the hardware and designers simply had to live with that fact. The development process was still paced to the airframe such that pre-planning was limited [1], the rate of change of technology was not very fast yet, and proliferation of non-standardized hardware was only starting to be a problem. Thus, even though constraints were tightening up they were still minimal.

Requirements

The requirement to include as many capabilities in the system as possible for the least cost, space, weight, and power could easily be handled by the Bottom-Up approach. Hardware costs were straightforward and inevitable, integration requirements were minimal, and architectures were simple. The reliability, maintainability, and survivability characteristics of the system were made up of the summed characteristics of its

components such that the related requirements were dealt with at the component level [27].

Bottom Line

In the pre-1960 era, avionics weren't organized in a system per se, the top and bottom sort of coincided. The system approach was merely that of installation. Avionic systems tended to be snapshots of the technology available at the time of development with little or no relationship to previous work [28]. As more avionics entered the scene, they were formed into systems composed of very loosely coupled subsystems [27]. The Bottom-Up design methodology was not only adequate but also the most cost-effective methodology. It was only concerned with ensuring that the various subsystems on board the aircraft worked properly [27].

Then, in the 1960s, computers entered the scene and architectures became centralized. The Bottom-Up approach was still capable of cost-effectively handling the design problem because the systems were still simple, there was not much integration going on, proliferation and standardization were not a problem yet, and the various requirements were still tied to the various hardware components making up the system.

As systems evolved and became more complex, the integration requirements increased and the new Integration by Evolution approach appeared. However the design methodology didn't change. It is at this point, around 1973, that a split between integration and design methodologies occurred. Before, both were more or less doing the same thing but now

they were going their separate ways. The Bottom-Up methodology could still cope with the problem of designing the now partially integrated systems but such systems were more complicated, more capable, and more tightly coupled. Thus, on the way from the bottom to the top, the designer frequently ran into integration problems that forced him to start back at the bottom. The Bottom-Up approach wasn't well suited for the Evolutionary integration approach now practiced but designers still kept it. It is conjectured that they thought it would be easier to deal with the integration problems than to change their design methodology altogether.

3.4.2 Bottom-Up to Top-Down

Technology

The advent of small and affordable computers and MIL-STD-1553 meant that every piece of data produced by one subsystem is potential information for every other subsystem [27]. This implied that the subsystem inter-relationships became very sophisticated. Thus avionic systems became highly complex systems as opposed to a gathering of independent subsystems.

A problem developed for system designers in that the leverage on technology shifted toward the commercial and scientific markets [11]. Thus, the designer was now faced with the problem of designing the avionic system with the available technology not necessarily being tailored to his requirements. This implied that he now needed a more structured

design methodology that would be as technology independent as possible.

Architecture

The change from a Centralized to a Federated architecture made it more difficult for the system designer to apply the Bottom-Up methodology. Processing capabilities were now incorporated in some subsystems such that communication between them became more sophisticated. A design methodology limited to interface considerations is not well suited to deal with a system having complicated subsystem inter-relations. However, although more difficult, the Bottom-Up design methodology was kept by system designers. They were more familiar with it and it wasn't evident that it incurred a significant increase in cost.

When the architectures turned Distributed, the problem got worse. Every subsystem and even sensors now had processing capabilities. Given both distributed computing and subsystems interconnected by a common highway, the Bottom-Up methodology could no longer cope with the problem to be handled [27]. The inter-subsystem relationships were just too complex to be handled by this methodology. Designing a Distributed system using such an approach meant constantly reverting to the bottom of the system every time an unpredicted situation occurred. Since these situations were quite common, the system design task tended to be long, arduous, and very costly.

Integration

Integration started to be a problem to the designers because their design methodology wasn't well suited for the integration approaches advocated in the mid-70s. The design philosophy avoided the risk of advanced technology in favor of low risk integration of present technology [32]. This lead to complex integration problems and ensuing high costs [32]. With more advanced technology and architectures, the situation just got worse. After the mux bus and MIL-STD-1553, there came standard computer languages (MIL-STD-1589), standard instruction set architectures (MIL-STD-1750), the modularity concept, core subsystems, and transferable software. Now, integration problems were too much for the Bottom-Up methodology to cope with. With a partially integrated distributed system, the main limitation is the ability of the design methodology to cope with the design of the system, not of the subsystems to cope with their tasks [27]. It wasn't a question of the Bottom-Up design method having greater difficulty achieving such an integrated system, this design methodology just could not produce such an integrated system.

Constraints

As usual, cost was a major constraint on the system designer. However, hardware was not so much a problem any more. Processors were cheaper than they had ever been, but the software had become a problem. With more sophisticated systems, software complexity had grown in proportion but software engineering techniques had not been used for the design

of avionic systems as had been the case for commercial and scientific applications [11]. This resulted in significantly increased software generation costs even up to the point where it exceeded the hardware costs [38]. The Bottom-Up approach did nothing to minimize software and associated costs. It is a process through which the designer assembles predefined and independent black boxes. Thus, it actually relied on software to try to link the various subsystems into a workable system [31].

Another prohibitive cost was the life-cycle cost. Through its life, the system had to be operated and maintained not to mention updated. With a system designed through the Bottom-Up approach, life cycle operation and maintenance is very costly. The Bottom-Up approach did not provide for reliability and maintainability of the designed system. Thus the system had a tendency to malfunction often and, every time the repair costs were high. However, the updating costs are even worse. Since the system was designed as a conglomerate of subsystems arranged in a specific configuration [27], changes to that configuration often implied re-design of the system. Such a cost is unacceptable.

Finally, development costs were too high. For designing simple systems the Bottom-Up approach is cost-effective. For complex systems, Bottom-Up design is such that the designer has to hope that on his way to the top of the system, there doesn't come an unforeseen situation. If this occurs, he has to go back to the bottom and re-design the system. Since designers are only human and thus don't have all encompassing and

f

unlimited foresight, these unforeseen situations occurred quite often in the design process. Therefore, the development process became a long and arduous endeavor. The associated costs became prohibitive not to mention that time constraints may not permit such a long development phase.

Requirements

With the technology changing at a rapid rate, system designers must plan for systems to be supported over a life cycle which is certain to see the obsolescence of the technology incorporated [11]. The obvious solution is to provide a design methodology that produces technology independent systems [11]. The Bottom-Up design method could not fulfill this requirement.

As the importance of software grew, it became increasingly necessary to select and design hardware based on software constraints [30]. It is even reported that contemporary (1976) avionic system developments were paced by software [30]. Software had become pivotal to the successful design and development of avionic systems [30]. This fact suggested that software should have greater influence on system design [30]. To achieve this, the designer must make sure that the earliest of system trade-offs include software [31]. This is not possible with the Bottom-Up design methodology.

Up to the 1980s, avionic system design has been largely driven by hardware implementation considerations [34]. Possible reasons for this are [34]:

- Most avionic system engineers are trained in hardware related discipline.
- Most major innovations occur as a result of hardware related activities or the reporting of it gives that impression.
- Aircraft functions have been traditionally identified in terms of a specific piece of hardware, on a one to one basis, and
- The believed software flexibility allowed system engineers to concentrate on the hardware design of systems.

This hardware driven approach was becoming increasingly difficult to apply for a number of reasons. Among them are [34]:

- With IC technology and bused transmission techniques, pieces of hardware and functions are becoming disassociated.
- Hardware first approach is preventing new technology advances to be incorporated in a newly developed aircraft, and
- Late software definition in the project leads to a need for software retrofits which follow the aircraft into service and lead to high software life cycle costs.

To alleviate these problems, the system requirements must be divorced, as far as possible, from the implementation of the system [34]. Thus, early system design must be performed in functional terms only and the functions should be as abstract as possible to distance themselves

from implementation considerations [34].

Bottom Line

By definition, the Bottom-Up approach produces a system that is not flexible and potentially has many loopholes that have gone unnoticed. Therefore, such a system is not reliable, is difficult to maintain, and cannot adapt to changing situations which accounts for low survivability and high vulnerability.

As can be seen, the Bottom-Up design methodology was not able to cope with the design task any more. This is not surprising because ever since WW 2, devices and software have shown significant improvements, but the system level design has changed very little [17].

What was needed was a design methodology capable of dealing with the existing technological, architectural, and integration requirements and constraints. The most important attribute of the new design methodology was that it had to be able to cope with the task of designing the total system as a unit rather than a collection of subsystems [27]. This is a first indication that the design task must start at the top of the system. Other indications are that the optimization of the design must proceed in step with the generation of the individual subsystems specification and that the optimization of any criterion may produce a sub-optimal solution to the others [11]. This means that trade-offs must be made throughout the design process. The design process must be Top-Down with iterations [11].

All this leads to, is the solution proposed to solve most of the design problems, the Top-Down structured design methodology described in Chapter 2. This methodology was seen to be the best available answer at the time but it did not necessarily solve all the existing problems of the time and it may or may not be suitable for future problems.

3.4.3 State-of-the-Art Design Methodology

As was mentioned in the previous section, the Top-Down design methodology is the state-of-the-art. It was a badly needed improvement over the Bottom-Up methodology. Although it solves all the problems of the Bottom-Up methodology, it is likely to have problems of its own. However, these problems haven't been identified as of yet. The reason for this is simple; the Top-Down design methodology has not been applied extensively to new development programs. It had been recognized as early as 1974 [2] as being required, but there is quite a gap between recognition and actual industry-wide implementation. Such implementation is taking place now.

With the scarcity of new development programs, sufficient experience with the design methodology might not be gained in time for the next generation systems. This implies that the next generation design methodology will likely be the Top-Down methodology described in Chapter 2. However, other considerations must be taken into account before selecting the next generation design methodology. This situation is addressed in Chapter 4.

3.4.4 Summary

The evolution of design methodologies has been quite inconsistent. The Bottom-Up methodology was used for a very long period of time and was applied to the design of the analog loosely coupled systems as well as the digital Centralized and Federated systems. The development process became longer and more costly as systems became more sophisticated but the designers held on to the Bottom-Up methodology because it was mature and proven. Then, when the development process reached a stage where it was critically long and unacceptably costly, the avionics community effected a complete shift in design philosophy. This shift brought the Top-Down design methodology. Such a radical change can only be explained by the fact that a change in design methodology was long overdue. Whenever this happens, a radical change is the only way to get back on the proper evolutionary track.

The state-of-the-art design methodology is the Top-Down methodology and it should be applied to current development programs. Even with changing technology and requirements, the Top-Down concept is the only one that allows the designer to treat the avionics as a total system. The requirement to do so is not expected to change in the foreseeable future such that the Top-Down methodology will be used for some time.

3.5 CHAPTER SUMMARY

The role of this Chapter has been to describe the evolution of avionics system engineering over the past two decades. This was done by

looking at each evolutionary step with relation to the reasons it came about, the impact it had on its times, and how it paved the way for the next evolutionary step.

The evolution of avionic system architectures went from a system composed of totally independent analog subsystems to the centralized architecture using digital components. This was brought about by the all new cost-effectiveness of airborne computers. Then, with digital technology becoming smaller and cheaper, some processing capabilities were given out to selected subsystems thus creating the Federated architecture. The hardware technology still became smaller and cheaper thus allowing a distribution of processing capabilities throughout the system at the same time easing a bit on the growing complexity of central management software by also distributing the control.

The Federated architecture did not play a major role in the avionic system architecture evolution. This is the impression conveyed by the lack of literature on Federated architectures. What little literature exists is centered in the late 60s and early 70s. A survey done in 1970 [12] reported two avionic systems that were truly Federated; the Integrated Light Attack Avionic System (ILAAS) and the Saturn Guidance and Control System. However, in the remaining systems surveyed, all were to a greater or lesser degree federated. Each contained dedicated processors assigned to specific functions which provide inputs to the central computer or utilize outputs. This fact suggests that the importance of the federated concept lies not so much as a total system architecture but

more as a subsystem architecture.

Architecturally, the Federated system is a transition between the Centralized and Distributed systems. It has both the disadvantages of centralization and some of the advantages of decentralization. Thus, it could not be seen as the answer to central processing. Furthermore, the rapid technological changes of the 70's were such that designers could see the Distributed systems coming quite fast. It is thus conjectured that the fast approaching Distributed systems, coupled with lengthy development times, incited designers to keep the centralized systems until it was possible to develop Distributed systems. On the other hand, the benefits of Federation could be gained, in the near term, by using it in the development of subsystems.

Finally, the importance of Federated systems should not be underestimated. The transition from the Centralized to the Distributed systems could not have been made without the transition to Federated systems. Such systems were the first to provide designers with practical experience of the distribution concept.

The evolution of system integration approaches started with the Integration by Accretion from the very beginning. The scope of this approach is quite broad and is mainly concerned with making sure the subsystems can be fitted in the aircraft somewhere and that they will work properly. This approach was kept until technology made it profitable to combine together functions with natural affinities. But it was not until

the multiplexed data bus appeared that this Integration by Evolution approach departed significantly from the Accretion approach. Then, with avionic systems becoming more sophisticated, it was necessary to control the evolution of avionic systems with such things as standardization, modularity, and core subsystems, i.e. with the Planned Integration approach.

The evolution of design methodologies started with the Bottom-Up methodology which was nothing more than grouping together a number of independently designed subsystems and call that a system. Again, this is quite broad in scope and this accounts for the fact that this methodology was applied up to the 1980's. It was only when it really became impossible to design the required complex system with this approach that designers turned to the logical alternative, the Top-Down design methodology.

CHAPTER 4 THE NEXT GENERATION

4.0 INTRODUCTION

Once the evolution of avionic systems engineering is understood a prediction can be made about the next generation of avionic systems. This prediction will come from an extension of the evolution of avionic systems, given that the next generation of avionic systems must solve the major problems of present day avionic systems and fulfill the requirements of the future systems.

This chapter presents an extrapolation on the evolution of avionic system engineering, as presented in Chapter 3, to arrive at a description of the next generation avionic system. This extrapolation will be done with respect to projected technology capabilities, requirements and constraints, based on past evolutionary trends and other pertinent factors depending on which subject will be discussed.

Section 4.1 arrives at a description of the next generation avionic system architecture. Section 4.2 discusses what direction the integration approach should take to be applicable to next generation avionics and, finally, Section 4.3 describes the design methodology that should be applied by the designers of the next generation avionic system.

4.1. AVIONICS ARCHITECTURE

4.1.0 Introduction

This section proposes a realistic description of the next generation avionic system architecture. The state-of-the-art architecture is the Distributed architecture but Chapter 3 has identified two problem areas of this architecture that need to be corrected. Furthermore, requirements and technology are evolving constantly such that the Distributed architecture is not likely to be the optimal architecture for the next generation of avionic systems.

Sections 4.1.1 to 4.1.3 cover the projected technology, constraints, and requirements aspects of the future architecture while Section 4.1.4 looks at the evolutionary trends expected to influence the next generation systems. Section 4.1.5 brings together all the previously discussed factors to arrive at a conclusion. Finally, Section 4.1.6 gives a description of the proposed next generation avionic system architecture.

4.1.1 Projected Technology

Software

Electronic hardware has made gigantic improvements over the last 25 years. These improvements have had a major influence on the evolution of avionic systems. These improvements have created the following hardware trends [31]:

- Inexpensive, modular computer components interconnected via high speed data buses.
- Extensive microprogrammable repertoire variability and emulation capabilities.
- Improved sensors with multiple embedded computer elements.
- Self-contained digital built-in-test-equipment (BITE) and in-flight-performance-monitoring (IFPM) in all major functional components.

During the same period of time, software has made significant improvements which have created the following software trends [31]:

- Emerging software engineering disciplines promise breakthroughs in productivity and reliability.
- Increased high order language (HOL) use and standardization.
- Improved documentation.
- Emulation of candidate target machines for "software first" approach to machine selection and system sizing.
- Early on-line simulation of sensor subsystems to support application software development in advance of hardware availability.

While hardware improvements have promoted a decentralization of computational resources, software has made the centralized approaches more

achievable and desirable [31]. In the past, hardware has had the most influence on avionic systems and this is illustrated by the evolution of architectures up to now. However, the influence of software has grown steadily and has now surpassed that of hardware, a situation expected to stabilize as such [31]. Figure 4.1 illustrates this software influence growth. Thus, centralized approaches should appear in one form or another in the next generation avionic system.

VHSIC

VLSI is expected to take care of the size requirements of the next generation integrated circuits and VHSIC will ensure the availability of such ICs to military systems. The VHSIC program is well under way and it is even expected that the aims of the program will be surpassed. The advances in microprocessor and microcomputers coupled with the VHSIC program will provide increased processing power thereby accelerating the trend toward multiple processors in military systems [22].

Artificial Intelligence

It has been surmised [17] that the desired future system capabilities can be achieved with current and emerging hardware technology and with an extension of currently developing software and system design approaches. The key technological advances making this possible are [17]:

- Low-cost, single-chip digital processors,
- High-speed, single-chip digital multiplex terminals.

Figure 4.1 Growing Influence of Software on Avionic System Design and Development

- Single-chip high density and high speed technology
- Artificial intelligence software and
- Intent driven design approaches.

The artificial intelligence (AI) feature of software has been permitted by the advent of inexpensive and powerful software [14]. Thus it is possible to develop computers with reasoning capabilities able to evaluate alternatives and make value judgements [14]. AI has been successfully applied in other fields than avionics, and it shows great promise for avionic applications [14].

Fiber Optics

Another important technological advance is that of fiber optics. This technology is expected to be used in the next generation avionics. It has the advantages of very high bandwidth capability without EMC problems [39]. The possible bandwidths will be on the order of 50-100 MHz [11]. As a starting point, DoD has developed a fiber optics standard (MIL-STD-1773) which provides an equivalent capability to MIL-STD-1553 [39]. Future standards are expected to take advantage of the bandwidth capabilities of fiber optics [39]. In fact, it is expected that a fiber optics bus will be fully implemented and operational by 1990 [42].

4.1.2 Projected Constraints

System Control

System control can vary between two extremes: centralized, or completely distributed control. Neither of these extremes is workable in practise [33]. Completely distributed control is difficult to modify as a result of changing mission priorities or system health [33]. Centralized control requires a very high data transfer rate and raises concern about fault tolerance [33]. A compromise has to be found.

Whatever the control scheme, some overhead in hardware and software will be required to perform this control function. This raises the point that the degree of flexibility in configuration must not exceed the point at which the cost of resources required for its accomplishment exceeds the benefits derived [33].

Software

Software complexity will increase because of the programmability requirement of the next generation architecture [33]. The demand for additional data processing will bring an increase in the amount of software. It is projected that the software will increase by an order of magnitude for the next generation avionic system [33]. Management of such software will have to be closely monitored.

It is anticipated [33] that the software required for programmable signal processors and configuration control processors will be developed

in an assembly language. This situation could only worsen an already severe software development and maintenance problem [33]. The only logical solution is to use a higher order language [33].

4.1.3 Projected requirements

General

Severe mission requirements of the 1990's dictate that future aircraft be able to sustain operations with minimal support, fly successful missions day and night in any weather, face a numerically superior and technologically advancing enemy aircraft and defense system [52,p156]. This implies requirements for higher sustainability, increased mission effectiveness, and lower life cycle costs [25].

The improved sustainability can be achieved through component reliability gains, the use of redundancy and resource sharing, the application of fault tolerant techniques, and a reduction in maintenance test and repair times [25]. Mission effectiveness can be improved by raising the role of the operator from system integrator to that of system manager, by providing partial or degraded capability in the event of a failure, and by increasing other specific mission capabilities [25]. Lower life cycle costs can be achieved by the wide use of common hardware and software modules, the ability to do on-aircraft test and repair of avionics, and the use of the VHSIC technology [25].

Artificial Intelligence and Information Fusion

Allowing the operator to act as the system manager will require the system to have adequate artificial intelligence to make routine decisions, and decisions which rely upon large quantities of quantitative data, on its own [17]. This requires the system to perform information fusion. Information fusion refers to the process of integrating the data from many discrete sensors to obtain maximum benefit from their individual characteristics [17]. It is essential to providing the operator with information which is as accurate and high level as possible [17].

Software Requirements

The following are the software requirements imposed on the next generation avionic systems [31]:

- Exploitation of improved sensors,
- Multimission/platform capabilities,
- Time stringent data capture/fusion,
- Real-time (R/T) data base management,
- R/T compacted product communication,
- Auto recovery/degraded modes,
- Simulated R/T exercises,

- Automatic mission/flight planning.
- System adaptability/growth, and
- System self-test/maintenance.

From these and the technology trends, we can deduce the future software and system impacts of these requirements. On the one hand, the first three requirements suggest that there will be a greater use of embedded, modular ultrafast signal processors with mission and platform reprogrammability [31]. This implies distributed, redundant computer and memory elements [31]. On the other hand, all of the requirements lead to more flexible, reliable central processing and control [31]. This will take the form of highly structured, common data base with multiple operator interaction as well as greatly increased number of operating modes and operator-common data entry/display capabilities [31]. All this implies centralized, redundant computer and memory elements [31].

Central Processing

Economic considerations have been the primary cause of the proliferation of embedded computers for dedicated use in sensors and other hardware-only subsystems [31]. The hardware technology trends with VLSI and VHSIC might suggest a disappearance of the requirements for central computer/software elements. However, the past two decades have shown that computers generate more data than operators can deal with and, in the face of increasingly complex systems, this problem will only get

worse [31]. Thus, not only to increase mission effectiveness, but just to be able to handle the situation, the operator will have to become the system manager [17]. This implies that he be presented with high level information and that the system automatically deals with routine matters. Even though the distribution of processing resources has major advantages, there are limits to the amount of distribution to be performed.

"When it's time to pull things together for a clear focus and definite direction at the system (or subsystem) level, we don't need another committee: we need a common, capable nodal entity working from a well organized, single task queue and a good set of policies and rules (algorithms) for recognizing and communicating the essential information." [31,p14]

Necessarily, such central software must have full control over an appropriate set of resources [31]. Due to their role, these resources should be the first to be considered for redundancy [31].

Apart from the support of real-time mission operations, a capable central computer/software system may be required for the following reasons [31]:

- To support on-board single-point operational readiness and maintenance checkout procedures,
- To facilitate system-wide error detection, recovery, and degraded mode operations,
- To support single-point-controlled, simulated mission exercises for software verification, training, and mission planning, and

- To provide self-host capabilities for program development and maintenance throughout the system life cycle.

Key Architectural Features

Recent operational experience has proven successful several key architectural features. They are [8.p655]:

- Extensive use of bit serial, word serial, multiplexed data transmission between "intelligent" avionic subsystems and "mission-oriented" computational and data processing resources that serve as the executive control center for the avionic system, perform mission oriented computations, and provides information for display and control purposes.
- A hierarchical network of built-in-test and in-flight monitoring capability distributed throughout the avionic system and monitored by the "mission" processors via the multiplexed data bus system.
- System reconfiguration that automatically occurs in the event of malfunctions to substitute alternate resources that retain as much mission capability as feasible. Reconfiguration is also required to react to changes in the environment and system status.
- Use of standard instruction set architecture, and
- Use of higher order language.

The next generation avionic system must refine and continue the development of the above mentioned features [8].

Information Transfer System (ITS)

Multiple/independent bus systems may be needed for at least two reasons. First, certain systems or groups of equipment may want to impose some restrictions on the transfer of information and who has access to it, i.e. functional isolation [22]. Secondly, excessive traffic loads on buses can be alleviated by several independent buses [22]. Although these buses would be independent, there is a need for some interface between them. However, for multiple ITS systems, a change in one results in an unpleasant ripple effect throughout the others [29]. This is due to the changes necessary in the centralized command/response software packages in each of the supporting ITS [29]. The solution is to implement a hierarchical structure of the multilevel multiplexed ITS's [29], [8].

There are discussions about whether or not the multileveled ITS should be entirely implemented with data buses. It has been suggested in [17] and [25] that a switched network could replace the data bus on several levels of the ITS. Some advantages of the switched network are [25]:

- High information throughput with reasonable hardware costs and good reliability.

- Well supported reconfiguration and
- Ease of growth because information-carrying capacity increases as the number of terminals in the system is increased.

However, the use of data buses should be extended because of the technology trends toward greater computing power more memory for less money and bus interface units at much less cost [39]. The conclusion to be drawn from this is that the choice of data buses or switched networks is an implementation detail. It does not change the requirement for a multileveled, hierarchical ITS. For the purposes of discussion, data buses are assumed to be used at all levels of the ITS.

Integration

Section 4.2.6 points out that the evolutionary trend of integration approaches is toward more integration. Therefore, the projected architecture must allow more integration than is currently practiced.

Growth

As the evolution of avionic system architecture has indicated, there is a continued need for a flexible avionic system architecture that has growth capability to accept modification without extensive change to the entire system [8].

4.1.4 Evolutionary Trend

The evolution of architectures brings forth two basic concepts: centralization and distribution of processing resources. The first concept to be applied was that of centralization, in the early 60s. The reasons for the application of this concept were purely technological. Because of the size, weight, and cost of computers, centralized processing was the only possible solution. Then, when technology started producing smaller, lighter, and cheaper computers, limited distribution of processing resources was attempted. This resulted in the Federated architectures of the early 70s. With technology advancing at a rapid rate and weapon system complexity increasing accordingly, the central management concept became unsuitable and very costly. Software engineering techniques were not sophisticated enough to produce manageable and cost-effective software and the bandwidth requirements were too high to be handled economically.

Fortunately, by the late 70s, computers had become light, small, cheap, and powerful enough for the concept of total distribution to be cost-effective. Distributed architectures resulted. Thus, the evolutionary trend was one of increasing distribution. Today, computers are still getting smaller, cheaper, and more powerful than ever, which would indicate the need for more distribution. But, a system can not be more distributed than a totally distributed system and these systems already exist. The next logical step would consist of complementing the distributed system with some form of centralized processing. It must be remem-

bered that centralized processing was discarded for technological reasons not because of an error in concept. Now that distribution can go no further it is logical to turn to centralization and hope to come up with an architecture combining the advantages of both centralized and distributed processing. The hierarchical architecture does just that [8].

4.1.5 Bottom Line

The next generation avionic system architecture is determined by the interplay of several factors such as technological capabilities, system requirements, and the system constraints.

The technological trends observed seem to move in different directions with respect to hardware or software. While the hardware trends point toward further decentralization of processing resources, the software trends make the centralized approach more achievable and desirable [31]. This apparent dichotomy is not one at all. Both approaches can exist simultaneously. It is just a question of the level of application of each concept. It is not illogical to picture a system with centralized processing at the top and distribution being gradually introduced at the lower levels to end up with total distribution at the lowest level of the system.

Section 4.1.3 indicated that software requirements demand both greater numbers of distributed computers as well as the continued use of centralized computer resources. Furthermore, it has been recognized that the operator will act as the system manager in the next generation air-

craft [17], [33]. This imposes a further requirement for centralized processing at the apex of the avionic system. So, both technological trends and requirements points toward a mixture of centralized and distributed processing. Thus, the first desirable characteristic of the next generation avionic system has been determined. It should consist of a system using centralized processing at the apex and distributed processing throughout the lower levels of the system.

But, is centralized processing realizable. The main problem with it is that it creates very complex software programs [13]. Since assembly languages were used for the generation and maintenance of such programs, software management rapidly became a formidable problem. However, Higher Order Languages (HOL) are replacing assembly languages as the avionic software language [10]. This, coupled with the significant software engineering improvements of the last decade, alleviates the problem of complex software management. The other reasons that made centralized processing disappear is the high bandwidth requirements it imposed [10]. This high bandwidth caused EMC problems for which the only solution was to impose a limit on the bandwidth through the multiplexed data bus [10], and distributed processing became more attractive because it reduced the required bandwidth [10]. However, as indicated in Section 4.1.1, fiber optics will be available for the next generation avionics. This fiber optics technology has two extremely important features, namely: it is impervious to EMC problems; and, it can handle very high bandwidths [39], [22]. Therefore, the bandwidth problem of central processing is solved.

Thus, the two major problems of central processing have suddenly been significantly reduced such that central processing is technologically realizable again.

There is also a need for a multiple ITS to be implemented in a hierarchical structure. Other features of the next generation avionic system architecture will be: extensive use of multiplexed data buses; a hierarchical network of BITE and in-flight monitoring capabilities; automatic system reconfiguration; growth capabilities with small impact on cost; artificial intelligence; data fusion; and, amenable to more integration. Finally, system control will have to be a compromise between fully distributed and centralized control.

The next generation architecture must meet all these requirements. Starting with the role of the operator, the most logical approach to provide him with high level information and to relieve him of system integration details is to provide a hierarchical architecture. In such an architecture, there can be a powerful central processing unit at the apex of the system [31]. This central processing unit would provide the operator with high level information that it would have processed from the low level data provided to it by the distributed systems thus giving a system centralized at the operator interface, but distributed at the sensor level. Such a system is also compatible with the technological trends pointing toward central processing as well as distributed processing since it would encompass both. ■

The benefits derived from VLSI and VHSIC will also help to fulfill part of the requirements. These benefits are expected to be [8]:

- A reduction in acquisition and support costs,
- Effectiveness improvements,
- Higher reliability,
- Reduced power dissipation,
- Smaller size,
- Less weight, and
- Improved BITE and in-flight monitoring capability.

Hierarchical architectures exhibit other advantages which confirm it as the next generation architecture. These are: it takes advantage of new technology; it overcomes the limitations in bus densities through multilevel multiplexed ITS; it provides orderly system growth; and, it increases avionic information systems [16]. Furthermore, a hierarchical system permits a higher degree of integration while preserving system integrity and present hardware/software standardization principles and will also permit a greater degree of fault tolerance in each system design [39]. It is also consistent with the issue of providing a control structure which is a compromise between a central and distributed structure. Hierarchical control is a good compromise. This architecture meets all the identified requirements.

4.1.6 Next Generation Architecture

The next generation avionic system will have a hierarchical architecture. Some of the salient features of such an architecture are: a mix of centralized and distributed processing with the centralized processing at the apex of the system; a multileveled, multiplexed information transfer system; artificial intelligence; and information fusion. However this does not address the general philosophy and workings of such of a system. There is an Air Force program which addresses these issues among others and came up with proposals that are endorsed. That program is the Pave Pillar program [25,p157].

Pave Pillar treats the avionic system as if it were a commercial enterprise. As Figure 4.2 indicates the system is intent driven and structured as a hierarchy with its basic philosophy being to do as much as possible, as low as possible. The operator expresses a mission intent to the system. This is the highest level. The system responds by determining what is required in the context of the mission and by taking appropriate actions with available system resources. Thus the operator is no longer required to control individual sensors and systems on the aircraft. He can focus his attention on mission management and situation awareness. Each level of the hierarchy has its own charter and responsibility serving the needs of the next higher level. In turn each higher level passes directives or intents to its immediate lower level which acts on them according to its charter and ultimately provides the required answer. The performance of each level may be reviewed by higher

Figure 4.2 Pave Pillar Implementation

levels to insure that proper interpretation of intent has occurred. Finally these higher levels of the system act as exception managers responding to problems experienced by lower levels.

4.1.7 Summary

The next generation avionic system architecture is expected to be hierarchical. This might seem to be a departure from the established evolutionary trends which have promoted increased distribution of processing resources and control as illustrated by the evolution from Centralized to Federated to Distributed architectures. However, close examination of this evolution reveals that the demise of Centralized systems was due to the increased cost-effectiveness of the distribution of processing resources and not to an error in concept. Distributed processing is still more cost-effective than ever but the significant improvements in software engineering coupled with the use of HOLs, have made centralized processing attractive again. This is just as well because requirements, such as the operator's shift from a system integrator to a system manager demand centralized processing.

The Hierarchical architecture is the only one that can make full use of both centralized and distributed processing. It is a logical evolution from the strictly Distributed architecture in the sense that the useful limit of distribution has been reached and it is time to re-introduce centralized processing in a manner such that it complements the distributed processing. The proposed architecture will consist of cen-

tralized processing at the apex (i.e. at the man/machine interface) augmented by distributed processing at all the lower levels. This architecture solves all the present day problems and has the potential to fulfill its role as the next generation architecture.

4.2 INTEGRATION APPROACH

4.2.0 Introduction

This section determines which integration approach will be most appropriate for the next generation of avionic systems. This approach will not be the current Planned Integration approach. This observation stems from the fact that Planned Integration is not even optimal for today's systems. When introduced, it did not fill the existing requirements of integration earlier in the development process and of being a standard integration approach. Furthermore, the next generation of avionic systems will add several more requirements to these two such that a new integration approach will be required.

Section 4.2.1 analyzes the projected technology. Section 4.2.2 discusses the impact of the next generation architecture on integration concepts while Section 4.2.3 considers the integration problems not solved by the Planned Integration approach. Section 4.2.4 and 4.2.5 present the projected requirements and constraints that will be determinant in the choice of the new integration approach. Section 4.2.6 summarizes the evolutionary trends influencing the integration approach while Section 4.2.7 brings together all the previously discussed factors

to arrive at a conclusion. Finally, Section 4.2.8 gives a description of the next generation avionic system integration approach.

4.2.1 Projected Technology

The combination of VLSI and VHSIC will make the next generation microprocessor even smaller, faster and cheaper than any previous one. This implies that microprocessors will be used in even more application areas than before and that the present application areas will be more cost-effective. This opens up the door to achievements such as: bus interface units at much less cost [39]; and, cost-effective implementation of many more functions than what was previously possible.

4.2.2 Projected Architecture

The next generation avionic system is expected to have a hierarchical architecture as described in the previous Section. The first impact of this architecture is that it will permit a higher degree of integration while preserving existing integrity and standardization principles [39]. This implies that if more integration is required, it can be applied without undesirable side effects with relation to system integrity and standardization. This is important because it opens up the door to the concept of increased integration.

The second impact is that the hierarchical architecture requires a maximum of integration. The first indication of this is the fact that the system will be composed of a central processing unit at the apex with

the lower levels gradually moving toward totally distributed processors at the sensor level. The system operation will be intent driven which means that each level provides intents or directives to its immediate lower level which decides what actions are required, takes them, and feeds back the appropriate data. This implies information fusion and artificial intelligence such that the system can automatically make as many decisions as possible and ultimately present the operator with selected high level information.

Efficient operation of such a system is only possible if a maximum of interactions within the system have been identified and compromises found. Every unidentified interaction in the system is a potential source of failure in the form of an unrealizable directive. These potential failure sources must be minimized to satisfy the increased requirements for cost-effectiveness, reliability, availability, survivability, and smooth operation, i.e. a maximum of integration. The ideal solution is the identification of all the interactions within the system.

4.2.3 Integration Problems

The Planned Integration approach does not solve all of the integration problems. However, it is said that avionic integration problems arise from inadequate consideration of interactions in the system network [3]. Thus, an essential premise to solve these problems is to adequately consider all of the interactions in the system network. Clearly, this implies the concept of a totally integrated system.

4.2.4 Projected Constraints

There are two basic constraints imposed on the future integration approach. The first one is that integration must take place earlier in the avionic development process. This constraint would appear to be an implementation detail, i.e. simply implement the desired approach early in the development process. This is fine but it assumes that all integration approaches are amenable to being implemented early in the development process. This is not necessarily the case.

The second constraint is that the next generation integration approach must have the potential of being a standard approach. This implies clearly defined aims and procedures.

4.2.5 Projected requirements

Weapon System

The Planned integration approach was expected to be a partial solution to the functional overpopulation and high performance requirements of modern avionics, the advancing technology completing the solution [1]. Indeed, the Planned integration methodology fulfilled its role. However the situation prevalent in the 70's hasn't really changed in the sense that it is still envisaged that weapon systems will be required to perform well beyond existing levels and will include many more functions than current systems [39]. Again, it is hoped that the advancing technology can answer the problem of the required increased functional densi-

5

ties. This is where VLSI and the VHSIC program will have their major impact. However, the technology is only expected to be a partial solution. It is only logical to assume that given similar situations, solutions to similar problems will be identical. This means more integration.

Advanced System

The advanced avionic system has to fulfill the following requirements: realization of data fusion/system automation; increased availability; and, reduced maintenance and life cycle costs [33]. The realization of these requirements imply further integration of avionic functions and hardware [33].

4.2.6 Evolutionary Trend

The main trend that is observed from the evolution of integration approaches is one of ever increasing integration. This is shown with the change from the Accretion approach, performing very little integration to the Evolutionary approach, integration through multiplexing, and finally to the Planned approach characterized by core subsystems and MIL-STDs 1750 and 1589. The main reason for this evolution was that of technology getting smaller and cheaper thus allowing more processing resources to be incorporated in avionic systems, providing them with more and more functions. This created systems that were growing in performance and complexity. At the same time, the rapid rate of change of the technology created a need for cost-effective growth that could be achieved with a minimum impact on the system. All this translated in

problems in system management operability maintenance and life cycle costs. Increasing system integration was perceived as an element of the solution [3] and this was confirmed by later events.

The situation of ever increasing system performance coupled with rapid technological change is still present and is expected to continue well into the next generation avionic system [33]. Thus the observed integration approach trend is accordingly expected to continue toward more integration than what is currently practised the limit being total integration.

4.2.7 Bottom Line

The conclusion to be drawn from the previous discussion is that still more integration is required. The technology is there to support more integration the proposed architecture will support as much integration as the designer will include, the integration problems would be alleviated with more integration all the requirements point toward more integration and finally, it is consistent with the evolutionary trends.

Once it is known that more integration is required one needs to determine what form this new integration approach will take. The first indication comes from Section 4.2.3 where it is said that to minimize integration problems, all the interactions in the system need to be adequately considered. The definition of Total integration given in Chapter 2 is "this integration approach is extended in a system hierarchy network considering all interactions in the network to arrive at a concept of to-

tion problem" and not all of it. However, this was to be expected because of the nature of this problem. Chapter 3 stated that part of it is due to the need for updating/upgrading of the weapon system and the other part is due to a lack of a proper integration approach. The Total Integration approach solves the part due to an improper approach but it cannot help for the updating/upgrading part. This problem must be dealt with by other means and will be addressed in Chapter 5. The Total Integration approach does solve all the integration problems related to an integration approach.

The final point to be made is that total integration may not be achievable. It is defined as the identification of all the significant interactions within the system and the finding of suitable compromises along the interaction links. With today's highly complex systems, it is virtually impossible to identify "all" the interactions let alone finding suitable compromises to all. Thus, truly Total Integration is practically impossible. However, all the factors studied pointed toward as much integration as possible and the approach that will allow the maximum of integration is the Total Integration approach. Therefore, it is the only possible choice.

4.3 DESIGN METHODOLOGY

4.3.0 Introduction

This section identifies the design methodology that needs to be used for the design of the next generation avionic system. As opposed to the

4.2.8 Next Generation Integration Concept

The Total Integration approach has been described in Chapter 2 but what was not mentioned is how to implement the approach. It can either be through specific steps applied in a distinct integration phase or it can be incorporated within the design phase such that the design steps would also serve as integration steps.

Prior to the 60s, integration and design accomplished the same thing. However, with the advent of the Integration by Evolution approach, each went their separate ways and were implemented separately. This was amplified with the Planned Integration approach. The main reason for this was that as the integration approaches evolved, the design methodology didn't. With the advent of a new design methodology, there is the possibility of bridging the gap between the two concepts. This question will be settled in Chapter 5.

4.2.9 Summary

The next generation integration approach will be the Total Integration approach. All the factors examined point toward this approach, including the evolutionary trends. It then represents a logical evolution from the Planned Integration approach rather than a revolution.

One of the points made earlier in this section is that the new integration approach must solve all the current and anticipated integration problems. This Total Integration approach solves part of the "integra-

tion problem" and not all of it. However, this was to be expected because of the nature of this problem. Chapter 3 stated that part of it is due to the need for updating/upgrading of the weapon system and the other part is due to a lack of a proper integration approach. The Total Integration approach solves the part due to an improper approach but it cannot help for the updating/upgrading part. This problem must be dealt with by other means and will be addressed in Chapter 5. The Total Integration approach does solve all the integration problems related to an integration approach.

The final point to be made is that total integration may not be achievable. It is defined as the identification of all the significant interactions within the system and the finding of suitable compromises along the interaction links. With today's highly complex systems, it is virtually impossible to identify "all" the interactions let alone finding suitable compromises to all. Thus, truly Total Integration is practically impossible. However, all the factors studied pointed toward as much integration as possible and the approach that will allow the maximum of integration is the Total Integration approach. Therefore, it is the only possible choice.

4.3 DESIGN METHODOLOGY

4.3.0 Introduction

This section identifies the design methodology that needs to be used for the design of the next generation avionic system. As opposed to the

architecture and integration approach, the next generation design methodology is not expected to be very different from the current Top-Down design methodology. This methodology is the product of a recent revolution in design methodologies such that significant changes should not be required. In fact, there might not be a requirement for any changes at all.

Section 4.3.1 presents the projected technology. Section 4.3.2 discusses the impact of the next generation architecture on system design, while Section 4.3.3 analyses the influence of the next generation integration methodology. Sections 4.3.4 and 4.3.5 consider the projected constraints and requirements respectively. Section 4.3.6 examines the trends that emerge from the evolution of system design as discussed in Chapter 3. Section 4.3.7 wraps things up while Section 4.3.8 summarizes the next generation avionic system design methodology.

4.3.1 Projected Technology

With VLSI and VHSIC, technology is continuing its route toward smaller and cheaper microcomputers. Thus, technology is expected to continue on the following directions [42]:

- Software functions taken over by firmware in near term.
- VHSIC chip take over software functions in the long term.
- Emergence of hardware macros as basic building blocks.

- Signal processing as dominant thrust,
- More modular software,
- New systems will be fault-tolerant, redundant reconfigurable
- Emergence of the smart, reconfigurable memory system, and
- Greater thrust for extracting data from aircraft/avionics systems.

This implies that the next generation systems will be more complex and sophisticated than ever before. "Intelligence" will even find its way into memory systems. With virtually every component of the system having a degree of "intelligence", they will only be made to work together if they are designed in a highly structured manner that will treat them as a total, unified system.

4.3.2 Projected Architecture

The next generation design methodology will have to produce a fully integrated hierarchical architecture (FIHA). What needs to be determined are the design methodology characteristics that are required by such an architecture.

The first characteristic is suggested by the structure of the architecture. This hierarchical architecture has a well identified top to the system and every other levels are subordinate to this top level. The top is also where the information and accompanying decisions are at their

highest level. The lower the level in the hierarchy, the lower the level of information and decision. Such a structure suggests that the design process must be a top-down structured exercise.

The second characteristic is suggested by several factors. The proposed architecture requires a high degree of automation, modularity in the development and maintenance of software, and automatic reconfiguration. These requirements can only be fulfilled by a functional partitioning of the system [25].

4.3.3 Projected Integration Concept

The integration approach that will be applied to next generation avionics is the Total integration approach. It's main characteristic is that it identifies all the significant interactions between the subsystems and finds suitable compromises along the interaction links. This immediately suggests that the design methodology must allow the designer to identify all of these interactions and then find compromises to them. Thus the design methodology must be a structured exercise starting at the top of the hierarchical system and working its way to the bottom. Furthermore when a compromise has been defined, the design methodology must allow the designer to evaluate its impact on the system and act on his findings. This implies that the design process must have iterations at each level of design. These iterations can be within a single level of the hierarchy or they can go back as far as the top of the system depending on the scope of the compromise reached. This suggests that the

design must be as abstract as possible in the beginning and that technical implementation details must be considered as late as possible [34]. The reason for this is that once the implementation details have been set, the iterative process is no longer possible.

A totally integrated system might not be achievable but the design methodology must enable the designer to come as close to a fully integrated system as possible.

4.3.4 Projected Constraints

Software

The first constraint imposed on the system designer is related to software. Software is rapidly gaining importance with relation to the design of avionic systems. Software has become pivotal to the successful design and development of avionic systems [31]. Its main impact is on the cost of the design process. Software development is becoming a more and more expensive proportion of total system cost such that it represents a critical path item in system development [11]. This situation is expected to continue and stabilize in the 1990's [30] [31]. Therefore, the next generation system design methodology must allow the software considerations to have an early, in-depth impact on the design process [31].

in Section 4.3.2, it is logical to require the system to be a top-down structured exercise.

Such factors as automation, modularity, reconfiguration, and availability suggest that functional partitioning is a must. Furthermore, integration requirements as well as the complexity of the system imply that the design process be iterative whereas economic leverage and technological considerations require the design methodology to be technology transparent. Finally, software must have an early and in-depth impact on design and standardization must be allowed for throughout the design process.

All the mentioned factors suggest that the Top-Down Design Methodology described in Chapter 2 is suitable for the design of the next generation avionic systems.

4.3.8 Next Generation Design Methodology

The Top-Down design methodology described in Chapter 2 does fulfill all the requirements of the next generation design methodology. What has not been identified are the potential problems of this methodology which is currently being introduced. In Chapter 3, it was pointed out that the reason for this is that sufficient experience with the methodology has not yet been gained and will not be gained in time for the next generation avionic systems. It was also mentioned that theoretical considerations might dictate minor changes to the methodology. This has not been the case. Therefore, the proposed next generation design methodology is

total system as a unit rather than a collection of subsystems [27]. State-of-the-art systems are simply too sophisticated to be designed as a collection of subsystems. This is already the case and will remain as such for several generations of avionic systems at the very least. Furthermore, functional partitioning is impossible to achieve without treating the system as a unit.

Standardization

Standardization is expected to be used in more areas than ever before. Already, multiplexed data bus standardization has proven the concept. Standardized instruction set architectures and high order languages are also a reality. Therefore, the system design methodology must have provisions for the timely introduction of standardization in the system. Hardware standardization must not be imposed early in the system design process because it could preclude the realization of the cost reductions achievable with VHSIC/VLSI [8]. Initial standards must be restricted to technology transparent standards that avoid hardware implementation impact [8]. Such standards are: multiplex bus standards; instruction set standards; and, higher order languages for software [8]. Hardware standardization should be phased in the design process with finalization reached prior to the production prototype and test phase [8]. This phased introduction of desirable standardization in the design process permits the selection of standards concurrent with state-of-the-art design approaches [8]. This will be discussed in more details in Chapter 5.

System Availability

As always, system availability must be at a maximum. One way to enhance system availability is to include a functional rather than subsystem orientation in the design process [17].

4.3.6 Evolutionary Trend

There hasn't been much of an evolutionary trend with avionic system design methodologies. Until very recently, avionic systems were designed as they had always been, through the Bottom-Up design methodology. Then, for reasons of system complexity and sophistication along with a need to reduce the cost of the development process, the designers turned to the Top-Down design methodology. This represented a radical shift in design philosophy. The main reason for such a drastic change was that a change was long overdue. This will not be the case for the next generation avionic system, thus a drastic change in design methodology will not be required. In fact, the situation expected to be prevalent at the time of this next avionic system is not going to be significantly different than the present situation. Thus, the Top-Down design methodology should only require minor changes, if any, to be applicable to the next generation.

4.3.7 Bottom Line

The bottom line is that we have identified several attributes of the next generation design methodology. The first is for the design methodology to treat the system as a unit. For this and other reasons stated

in Section 4.3.2, it is logical to require the system to be a top-down structured exercise.

Such factors as automation, modularity, reconfiguration, and availability suggest that functional partitioning is a must. Furthermore, integration requirements as well as the complexity of the system imply that the design process be iterative whereas economic leverage and technological considerations require the design methodology to be technology transparent. Finally, software must have an early and in-depth impact on design and standardization must be allowed for throughout the design process.

All the mentioned factors suggest that the Top-Down Design Methodology described in Chapter 2 is suitable for the design of the next generation avionic systems.

4.3.8 Next Generation Design Methodology

The Top-Down design methodology described in Chapter 2 does fulfill all the requirements of the next generation design methodology. What has not been identified are the potential problems of this methodology which is currently being introduced. In Chapter 3, it was pointed out that the reason for this is that sufficient experience with the methodology has not yet been gained and will not be gained in time for the next generation avionic systems. It was also mentioned that theoretical considerations might dictate minor changes to the methodology. This has not been the case. Therefore, the proposed next generation design methodology is

indeed the Top-Down methodology of Chapter 2.

4.3.9 Summary

The aim of this section was to determine the next generation design methodology without pre-conceived answers. All the factors considered pointed toward a methodology with the following characteristics: top-down, structured algorithm; functionally oriented; systems oriented; technology transparent; software conscious; iterative; and, amenable to standardization.

The Top-Down methodology described in Chapter 2 does fulfill these requirements. Changes in the current methodology are not expected because the problems of this methodology have not yet been identified. It is presently flexible enough to cater to a variety of situations. In fact a good design methodology should not have to change because of changing technology and increasing mission capabilities requirements. It is only a guide to help the designer arrive at a suitable system, whatever that system may be.

4.4 CHAPTER SUMMARY

The aim of this Chapter was to give a theoretical description of the next generation of avionic systems. In doing so an evolutionary approach was predicated. The main reasons for this approach is to minimize the risks involved in the development of such a system. The importance of such risk minimization will be addressed in Chapter 5.

The proposed architecture is the Hierarchical architecture. The main reason for its selection is that it allows the introduction of such concepts as information fusion, multileveled ITSs, a mixture of central and distributed processing, and hierarchical control. It is also consistent with the observed evolutionary trend that requires the logical evolution from complete distributed processing, hierarchical processing, to be implemented. It is interesting to note that fully centralized and fully distributed architectures, although they have many advantages, were not suitable for the next generation of avionic systems. This was to be expected because there exists an architecture that combines the advantages of both architectures while minimizing their disadvantages. Although the concept of a Hierarchical architecture is not new, the technology has only now reached a stage where it is implementable at an acceptable cost.

The choice of a new integration approach was straightforward. The Planned Integration approach was not optimal for existing systems. Therefore, it could not be optimal for future systems, given more demanding requirements and more sophisticated systems. Furthermore, all the factors considered pointed toward more integration than what presently exists, including the evolutionary trend. This is one of a few situations where the "if a little is good, a lot is better" syndrome does apply. In this case, the maximum is the best. Several factors such as the architecture, the need to solve existing integration problems, and the need for a standardized approach all point to the Total Integration ap-

proach. This is not surprising because Total Integration is the means by which all the possible behaviors of the system have been accounted for. The only drawback being that the more complex the system, the more difficult it is to account for all possible behaviors. In fact, it rapidly becomes impossible to achieve but what is important is to try to account for a maximum number of them. Once this is done, all the designer can do is hope that the ones he missed will have little impact on the performance of the system.

The proposed next generation design methodology is the current state-of-the-art Top-Down methodology. This was to be expected due to the recent revolution in design methodologies which introduced the Top-Down design methodology. However, such considerations did not enter in the analysis of the next generation design methodology. Top-Down design was selected based on its merits only. The main characteristics of the next generation design methodology were determined to be: top-down, highly structured exercise; iterative process; functional partitioning; and technology transparency. Only the Top-Down methodology fulfilled these requirements. It is comforting to note that this analysis confirmed that the revolution from Bottom-Up design to Top-Down design was justified.

CHAPTER 5 ACHIEVING THE NEXT GENERATION

5.0 INTRODUCTION

The aim of this chapter is to describe the specific actions that need to be taken if the avionic system described in Chapter 4 is to be implemented. This involves identifying the problems of realizing the proposed avionic system and finding suitable solutions to these problems.

In this chapter, one main action will be performed. This action is to determine the impact, on the proposed avionic system, of factors such as support systems, standardization issues and any other factors deemed relevant. This analysis will be divided in three areas: the architecture; the integration approach; and the design methodology. The purpose of the analysis is to find compromises to any encountered problems. These compromises can take on several forms: they can be a re-adjustment of the proposed avionic system, resulting from unrealistic expectations; they can be a specific action to be taken within a particular area to make it suitable for the proposed system; or they can be a bit of both. In short one can either change the proposal, change the world in which to implement it, or change both. However changing the proposal of Chapter 4 will be considered as a last resort only.

Section 5.1 assesses the impact of the proposed architecture along the lines mentioned above. Section 5.2 considers the proposed integra-

tion approach with relation to its implementation while section 5.3 does the same for the design methodology.

5.1 AVIONICS ARCHITECTURE

5.1.0 Introduction

This section identifies and proposes solutions to the potential problems of implementing the Hierarchical architecture. This will be done by analyzing the specific factors affecting the implementation of such an architecture. More specifically, this means confronting it with factors such as standardization, ITS, artificial intelligence, and maintenance philosophy.

Section 5.1.1 discusses the architectural requirements imposed on the information transfer system and what kind of an ITS would satisfy them. Section 5.1.2 considers the aspect of artificial intelligence as it applies to avionic systems. Section 5.1.3 determines which maintenance philosophy is most appropriate for future aircraft. Section 5.1.4 addresses the standardization issue, is there more or less standardization required and in what areas. Finally, Section 5.1.5 briefly introduces the problem of resistance to changes on the part of government and industry.

5.1.1 Information Transfer System

One of the most important aspect of the implementation of any architecture is how to move information throughout the system, i.e. what in-

formation transfer system (ITS) to use. In the case of the proposed hierarchical architecture, it has already been decided (section 4.1.3) that a multilevel multiplexed ITS was required. Now, what needs to be decided is whether or not the existing ITS can meet the requirements of the hierarchical architecture.

Bus Protocol

A typical hierarchical ITS is composed of the following buses: a global bus connecting all major subsystems, a common sensor bus connecting all sensors common to several subsystems, and finally buses local to each subsystem [39]. Presently, the state-of-the-art in ITS is the MIL-STD-1553B data bus with a stationary master protocol (SMITS). SMITS implies that one processor governs the use of the bus. All other devices on the bus transmit only when directed to do so by the master bus controller [39]. It is generally accepted that MIL-STD-1553B will remain an effective protocol for local buses but not necessarily for the other buses [39], [29]. As shown by Figure 5.1, the main advantages of a SMITS are minimal risk during development, the use of a standard protocol, and a simple and reliable bus interface unit [39]. However, its disadvantages are a lack of autonomy, multimission inflexibility, and limited functional enhancement [39]. For buses like the global and common sensor bus, a contention master ITS (CMITS) is a viable candidate. The advantages of the CMITS are a higher degree of autonomy and no elaborate bus controller backup scheme [39]. The disadvantage deals with the asynchronous communication system requiring different application algorithms

	STATIONARY MASTER ITS	CONTENTION ITS
DESCRIPTION:	DAIS control procedures were modified to include MIL-STD-1553B and multiple bus interfaces, but is essentially DAIS in concept.	New control procedures are required as well as a new bus protocol. The operation of the system is significantly different from the other ITS.
BIU:	Simple/reliable BIU. One is already in design which would suffice (two chips).	Simple/reliable BIU. Random delay function is only addition to transmitter.
RISK OF DEVELOPMENT:	Minimal. (DAIS is the base from which changes would be made.)	Hardware/software development; asynchronously operating software is more difficult to integrate than synchronously operating software.
CONTROL DEPENDABILITY:	Single point control.	Greatest independence of functions.
BUS EFFICIENCY:	Overhead is primarily concerned with message movement.	Low efficiency with high collisions.
SYNCHRONOUS MESSAGE PROCESSING OVERHEAD:	Efficient synchronous message processing.	Not applicable - all message processing asynchronous.
TIME CRITICAL RESPONSES:	Asynchronous (trigger) messages require longer wait and processing times than the other two ITS.	Much faster than the other ITS.
FUNCTIONAL ENHANCEMENT:	Any function added to the distributed system would require a change to the executive table.	Functions added only affect the processor in which the function is added.

Figure 5.1 Information Transfer System Comparison

Figure 5.2 Modular Concept Supports Two Level Maintenance

and a very high data capacity [39]. However where weight is of prime importance MIL-STD-1773 (fiber optics equivalent of 1553B) will replace MIL-STD-1553B.

Fiber Optics

Fiber optics being the preferred transmission media for the high bandwidth buses, one needs to know if the fiber optics technology will be ready for the next generation avionic system. The following statements are representative of the view of industry. The feasibility of fiber optics for aircraft avionic interconnects and buses appears probable during this decade (i.e. 1980's) [38,p1-2]. It is expected that provisions will be made for higher speed fiber optics buses in the next few years (1980-90) [39,p20]. The transition from copper to fiber optics for transfer of signals within a weapon system may take place within this decade (i.e. 1980's) [38,p23]. It is logical to expect that as a minimum, a fiber optics bus will be fully implemented and operational by 1990 [42,p4-2]. Thus, it is generally accepted that fiber optics buses will be a reality by 1990, which is just in time for the next generation avionic system.

5.1.2 Artificial Intelligence

The FIHA avionic system will require the use of artificial intelligence to provide the increased automation and decision aiding assistance which is required to transition the role of the operator from system integrator to that of system strategist and manager. Because of a few re-

cent successes in industrial and commercial areas the subfield of expert systems (ES) seems particularly attractive [14].

ES Advantages

An important issue is to determine what practical advantages ES technologies can have for avionics. The first one is that they may facilitate automation and decision aiding in situations where this was traditionally impractical [14]. This ultimately means an increase in performance but it has yet to be determined how much of an increase is possible for the next generation avionics. The second advantage lies in the quality of the solution. The best example of this is the inclusion of an explanation capability [14]. This means that the AI processor would provide the operator with the main reasons for having reached a specific conclusion [14]. This may help the operator to better understand the capabilities and limitations of the system [14]. The third advantage deals with the design process and has two aspects. First early validation of requirements and key design aspects are facilitated by the early prototyping approach typically used in ES [14]. Secondly, the symbolic reasoning capabilities of ES allow much of the design to be developed and evaluated in the jargon of the pilot or operator [14].

Avionics Constraints

The avionics domain imposes specific constraints on ES technology. These constraints can be divided in three broad areas [14].

- Requirements to operate in real-time as part of a larger integrated system
- Designing for future systems and
- The use of explanation in a cockpit environment.

The "real-time" constraints mainly translate in computational resource limitations and situation dynamics limitations [14]. Existing ES tend to be resource intensive and built on the premise that the problem is relatively static [14]. The "future systems" constraints deals with the fact that the ES will be developed for inclusion in a future weapon system for which an expert does not yet exist. The participation of such an expert is a key distinguishing characteristic of the technology [14]. The "explanation capabilities" constraints arise because results are typically displayed in partially codified form on a CRT i.e. "Rule 122 concluded that ..." [14]. This, although appropriate for pre- or post-mission activities as well as training, is less than optimal for many kinds of decision aiding that could occur during a mission [14]. There might be time and cockpit display space limitations and ES might impose unreasonable memory or information processing requirements on the operator (i.e. what is Rule 122?) [14]. ES avionics will need to present the explanation in a manner that can be assimilated rapidly [14].

Technology Development

Technology must be developed such that it can compensate for the above mentioned constraints. Such development is required in the following areas [14]:

- Real-time issues.
- Integration of both ES and traditional approaches into synergistic systems.
- Explanation in the cockpit.
- Design for future systems.
- Training of personnel.
- Implementation language (LISP vs ADA), and
- Prototype systems (ES is an empirical discipline).

5.1.3 Two Level Maintenance

An important requirement for the next generation aircraft is to reduce the cost-of-ownership as well as increase maintainability [25]. To achieve these goals, the maintenance concept must move from a three level (flight line, intermediate shop, and depot) to a two level (flight line and depot) concept [3], [25], [37]. Presently, a three level concept is required because it would be too expensive to buy the spare line replaceable units (LRUs) for the base/depot pipeline as needed in a two

level concept [37]. This high cost has two causes: the high removal rate and the high cost of current LRU computers. A significant reduction in the removal rate and the cost of the removed units would render the two level concept economically feasible [37].

The proposed reductions can be accomplished by the implementation of standardized modular avionics [37]. "Modularized electronics in small packages coupled with carefully selected common module functions and improved fault isolation techniques can provide the basis for eliminating intermediate-shop-level maintenance" [25,p158]. Figure 5.2 depicts the modular concept used to achieve these goals.

The mainstay of this modularity concept is the proper selection of the common modules. The current approach of black boxes with internal circuit boards/cards is not optimum because too many capabilities are housed within each LRU [25]. This results in large, expensive, and unique cards and LRUs which require intermediate shop maintenance because complex and expensive boxes fail frequently and cannot be cost-effectively procured in sufficient quantities [25]. On the other hand, too little capacity in modules requires too many system interface connections and much too complicated fault-isolation techniques [25]. What is needed is a compromise position offering the potential for significant cost avoidance [25]. Figure 5.3 illustrates these relationships.

The choice for a compromise is based on a figure of merit (FOM) computed as the quotient of spares unit cost and mean-time-between-removals

Figure 5.2 Modular Concept Supports Two Level Maintenance

Figure 5.3 Hardware Modularity Trade-Off

(MTBR) [25]. Values between one and ten dollars per flight hours appear to be optimum [25], [37].

5.1.4 Standardization Issues

Data Bus

The FIHA system will use multilevel multiplexed buses. These buses will require different protocols and bandwidths depending on their application. To complement the use of the stationary master protocol MIL-STD-1553B, a contention master protocol is required. When defined, this protocol will have to form part of a MIL-STD if it is to reap the same benefits as did MIL-STD-1553B when introduced.

Another area requiring standardization is the bus bandwidth. Buses with bandwidths higher than 1 MHz are required and thus will have to be included in a MIL-STD. Depending on their application, the bandwidth requirements of the bus vary such that several bandwidth standards will be required [40]. Finally, since the high bandwidth buses (>1 MHz) are expected to be implemented with fiber optics, appropriate standards should be defined for that transmission media. DoD has started work on this area with MIL-STD-1773 but more standards are still needed.

In conclusion, new bus standards are needed to cover a contention master protocol, a new transmission media, and several new bandwidth requirements. Such standards are necessary and sufficient for the next generation ITS.

Instruction Set Architecture (ISA)

Another area requiring standardization is the instruction set architecture. The main reason for this requirement is to allow the design of embedded computers with low cost state-of-the-art microprocessors [39]. MIL-STD-1750 is a first attempt at such standardization but several more standard ISAs need to be approved [39]. These standard ISAs will probably be included in DoD instruction 5000.5x.

Higher Order Languages (HOL)

HOL was earlier identified for the cost-effective development of the next generation avionic software. DoD recognized this fact as early as 1976 with the directive 5000.29 which stipulated that a HOL will be used in DoD systems unless proven not to be cost-effective over the life cycle of the system and, directive 5000.31 which specified six standard HOLs for interim use in new DoD systems [32]. The present effort of DoD is to standardize on one common HOL, ADA. This standardization effort will be sufficient to support the implementation of the next generation avionic system in this area.

5.1.5 Miscellaneous

It has been said [17] that the software and hardware technology required to achieve architectural improvements are already here and simply need to be improved, properly applied, and integrated (i.e. incorporated through a design process). Although this is simple to say, it is not

simple to achieve. The main problem lies in getting new ideas adopted and implemented in high cost programs where the pressures of conservative low risk development and the inertia of sticking to old standards is the norm. This translates into two basic problems: getting funding approved for programs based on new (mostly unproven) approaches; and, getting industry to adopt new methodologies. Section 5.3.1 will deal with this in more detail.

5.1.6 Summary

The implementation of a hierarchical architecture requires work in several areas. A contention protocol will need to be developed, high bandwidth buses will need to be implemented in fiber optics with appropriate standards developed, further standardization is required in the areas of ISA and HOL, a two-level maintenance concept needs to be proven with appropriate strategies developed, and finally, the ES technology will require significant research and development.

5.2 INTEGRATION APPROACH

5.2.0 Introduction

This section addresses the problem of implementing the Total Integration approach. The main factor affecting the implementation of this concept is the choice of an implementation mechanism: integration methodology or design methodology. The other factors to be considered are the suitability of the Hierarchical architecture and the impact of the stan-

standardization goals.

Section 5.2.1 considers the best way to implement the proposed integration concept. Section 5.2.2 examines whether or not the hierarchical architecture can support the requirements of the Total Integration approach. Finally, Section 5.2.3 discusses the standardization issues specific to the integration domain.

5.2.1 Design versus Integration Methodology

What needs to be determined at this point is how to perform total integration, or at least, as near to it as possible. Is it through a specific, separate integration approach, or can it be achieved through a design methodology?

First, consider what is meant by total integration. Total integration examines the processing capacity and configuration required to satisfy the combination of all of the mission dependent and independent processing requirements. The process is complete when all significant interactions between subsystems have been identified and suitable compromise have been made along the interaction links. Examining the required processing capacity and configuration to satisfy the processing requirements is a system design task. This task is part of the functional decomposition process of Top-Down structured design. Furthermore, to continue until all significant interactions between subsystems have been identified and compromises found is a task achieved through iterations of the decomposition process. This is best illustrated by a close examina-

tion of the Top-Down design methodology given at Chapter 2.

The design process starts with the specification of the avionic system requirements. These include the mission dependent and independent processing requirements. The next step is to partition the system in a set of functional algorithms which are in turn partitioned in elemental program units. This is followed by the definition of the interdependency and interface structure of the elemental program units. Thus, the various subsystems along with some significant interactions between them have been identified. By defining the interdependency and interface structure, suitable compromises to the identified interactions are being made. The next step, in the design process, is to refine the interdependency and interface structure by iterations of the decomposition process. This implies identifying more interactions along with suitable compromises to them. Obviously, the iterations stop when all significant interactions have been identified and suitable compromises found. At this point, the requirements are checked. If all of the requirements have been fulfilled, the designer moves on. If not, further iterations of the decomposition process are required. This implies a re-assessment of the processing capacities and configuration previously identified such that they now satisfy the combination of all of the identified requirements. This means that more interactions need to be identified and/or the compromises need revision such that they will completely satisfy all of the requirements. When this is achieved, the designer selects the appropriate implementation for the program elements and constructs a processing net-

work. This process is repeated at successively lower system levels until the lowest level is attained.

This design process performed all of the actions required by the Total Integration concept. These actions are part of the design process and are incorporated within other actions performed by it. It is not possible to isolate and separate these actions, they form an integral part of the design process. Taking them out would only result in an incomplete and unsuitable design. Therefore, there is no need to revert to a separate integration approach since the appropriate application of the Top-Down design methodology achieves all of the goals of Total Integration, or at least, as close to it as possible.

Furthermore, there is a requirement to do the integration as early as possible in the development process. The design of the system is the initial phase of the development process such that the achievement of Total Integration through the design methodology implies that the integration is done as a part of the initial phase of system development. This confirms that the Top-Down design methodology is indeed the proper implementation mechanism for the Total Integration approach. This is why it is said that the quality of integration is the direct result of the quality and scope of the system design [3]. By extension, it can also be stated that concepts of system integration are also concepts of system design [3]. All this has lead to the following definition of an integrated avionic system: one which has been partitioned through an orderly system design process [3].

In conclusion, it can be said that part of the "integration problem" is solved by rigorously applying the proper system design methodology. However this must not be construed to be an easy way out of an annoying problem. It merely forces the designer to focus his attention at the right place, the design process. Obviously, this means that the design process will become more involved since the onus of total integration will be on it. However, in my opinion, it represents the most cost-effective way to produce a totally integrated avionic system.

The final point to be made is that applying the appropriate design methodology does not guarantee the absence of integration problems. The problems of updating and upgrading the system with an already designed hardware and/or software module are still present. Such problems can be ameliorated through proper standardization. This will be discussed further in section 5.2.3.

5.2.2 Avionics Architecture

The Total Integration concept will be implemented provided that among other things, the Hierarchical architecture will support it. The following statement addresses this problem: hierarchical architectures show great promise in providing the means to achieve the benefits of a totally integrated avionic system without jeopardizing flight critical isolation requirements [39,p20]. This statement means that hierarchical architectures not only allow total integration but they also allow the effective integration of flight critical subsystems requiring functional

isolation. This is possible because of the multilevel multiplexed ITS with contention protocol, expected to characterize these architectures. A complementary statement is: a new HOX type system (multilevel multiplexed with contention protocol to be defined as a MIL-STD) is needed as the medium to interconnect avionic subsystems for total weapon system integration on the next generation aircraft [39,p31-7]. These two statements mean that not only is the hierarchical architecture well suited for total integration, but it is required for its achievement.

5.2.3 Standardization Issues

As early as 1978 it was recognized that a cost-effective application of integration required standardization at its highest level [13]. At that time, standardization was limited to mux data buses (MIL-STD-1553). More standardization requirements lead to the introduction of instruction set architecture standardization (MIL-STD-1750). With the projected total integration concept, still more standardization is required. This required standardization is in the areas of processing elements, interconnect systems, communication protocols, and maintenance approaches [33]. There are significant efforts directed at this problem such as the development of fiber optics bus standards to match MIL-STD 1553B as well as to provide increased bandwidth, the development of a standard HOL (Ada), and the development of a standard ITS (by a Task Group within the SAE/A-2K Subcommittee on Multiplexing [29]).

Apart from the requirement for increased standardization to allow total integration of new systems such increased standardization is also required to alleviate costly integration problems encountered when updating and/or upgrading an avionic system. If there is across-the-board standardization (interface protocol, ISA, HCL, processing element, etc.) then addition of new capabilities can be performed without integration problems. It will simply be a matter of physically connecting the proper component/subsystem at the right place and the standardization guarantees that the added component will work appropriately as a part of a homogeneous system. However standardization must be at the highest level possible so as not to impair new technology insertion [33]. Thus a compromise must be made between elimination of integration problems and new technology insertion.

5.2.4 Summary

In addressing the problem of implementing the Total Integration approach, it is proposed that the best and only way to achieve such goals is through the application of the Top-Down design methodology. The application of this methodology allows the designer to perform total integration at the earliest time possible in the development process. It is also proposed that a multilevel, multiplexed information transfer system, with a contention protocol, is required for the implementation of the Total Integration approach. Finally, with respect to standardization, much more is needed but how much more is a question of trade-off with the problem of new technology insertion.

5.3 DESIGN METHODOLOGY

5.3.0 Introduction

This section deals specifically with the problem of designing the proposed avionic system with the Top-Down design methodology. This problem can be divided in three distinct parts.

First, are there the appropriate tools to support the design methodology. Secondly, what is the importance of the requirement specification and how does it impact the design methodology. Finally, what kind of a standardization program is required and is it consistent with the Top-Down design methodology. Before addressing these problems, however, it is important to note the effect of the magnitude of avionic development programs on the acceptance of changes.

Section 5.3.1 outlines the problem of inertia in the acceptance of changes that is present in all major development programs. Section 5.3.2 determines if the appropriate support tools are in existence or can be built in time. Section 5.3.3 explains the need for clearly stated and well defined requirement specifications. Finally, Section 5.3.4 examines the standardization issue.

5.3.1 Inertia of Major Development Programs

There is one factor that is crucial to the development of avionic systems and that is the enormous size, in terms of cost and resources, of today's avionics development programs.

Such programs are big programs managed by big business with the help of big government fundings. To get an idea of the size of avionic programs, one only has to look at the Canadian purchase of the F-18 aircraft. The Canadian government settled for approximately 135 aircraft at a total cost of about 2.5 billion dollars. By the time the spares are bought and all aircraft become operational, this figure will have risen to four or five billion dollars. This is a very small procurement program by United States standards. DoD as a whole is buying thousands of aircraft which represent investments in the hundred billion dollar range. With the avionics representing one third of the weapon system cost, such programs require investments in the tens of billion of dollars. This is big!

The problem with such big programs is one of inertia, i.e. changes occur very slowly. This is understandable because of the kinds of money involved. The companies and government cannot afford to try out approaches that are not thoroughly proven or, at least have a very high probability of success. What this means is that when a significant change of approach is required, it will likely occur in several intermediate steps. This way, the risks are minimized as each intermediate step is proven before the next one is initiated. This represents an evolutionary approach as opposed to a revolutionary one.

5.3.2 Support Tools

Acceleration of Acceptance

No matter how good a system design methodology can be, the designer has to live with certain facts of life from which he cannot escape. One of them is that an avionic system as a whole is developed by a Program Management office (PMO) and that such a development program requires huge investments (several billion dollars or more). Typically, such a development program is plagued with constraints which result in demands for low risk demonstrated implementation, where possible [3]. Also, a PMO has predefined, unchangeable organization roles. This, combined with the everyday conflicting trade-offs facing the designer, results in a strong resistance to radical departures from familiar system architectures [3]. Sufficient confidence cannot be gained in untried approaches because of a lack of complete system models [3]. Finally, a PMO lacks both the charter and resources to solve problems having applications to other systems [3].

But, one might ask, why is it that this is a fairly recent problem. The answer is as follows: in the past, each new weapon system development brought with it an increment of innovation [3]. The rate of change of technology was such that each new increment was understood and matured before the next new or strange increment of progress [3,p5]. Radical changes were simply the result of several incremental steps. However, the situation has evolved such that the rate of change of technology has

increased tremendously, that there are fewer opportunities in new weapon system developments, and that the mission capability requirements are continually increasing. This means that progress may not be fast enough in achieving integration within the avionics and between avionics and other interacting systems [3,p5]. What needs to be done, then, is to accelerate progress towards new capabilities.

A demonstration of the architecture could provide sufficient maturity to an integrated system concept to accelerate progress towards the new capabilities required [3,p5]. The most cost-effective way to produce such a demonstration is through a ground-based laboratory facility using advanced software engineering techniques such as the host-target approach. Such a facility is commonly known as a simulation/integration lab. This situation was recognized by the US Navy as early as 1977 when they embarked on the BASIC (basic avionic system integration concept) program [1], [22]. The above mentioned trends continued and this prompted the UK Ministry of Defense (MoD) to establish, in 1981, an Avionic System Demonstrator Rig (ASDR) [15]. Both of these programs are dedicated to improve the transition of technology into new aircraft projects and to reduce the risk factor to allow the development and implementation of a more advanced system than would otherwise be possible. For the design of the next avionic system, such a program is even more critical since the technology is changing faster, there are few opportunities of new development programs, and mission capabilities requirements are still increasing. Typical objectives of such a program are [15]:

- To support the development of avionic systems for a future tactical aircraft.
- To provide practical experience with an advanced avionic system using data bus communication.
- To demonstrate the techniques of software management.
- To support the development of new maintenance procedures.
- To investigate the specification, procurement and management procedures required to produce future avionic systems.
- To develop new system control strategies and to demonstrate acceptability.

Avionic System Development

The next generation of avionic systems will be developed as totally integrated systems achieved by the Top-Down design methodology. This implies several actions. The first one is to identify all interactions within the avionic system and find compromises to each one. With the complexity of current and expected systems, this is a formidable task. Section 5.2.4 concluded that although total integration might not be possible, designers must work toward achieving a maximum of integration. This maximum of integration is only possible through the use of a simulation/integration lab. The designer cannot be expected to think about all possible interactions going on in a highly complex system.

However system simulation can point out "hidden" interactions such that the designer can achieve a maximum of integration.

The second action to be taken is that of evaluating the impact of the selected compromises. The criteria for the selection of suitable compromises is that they have the required behavior without undesirable side effects. These side effects can only be determined by performing system simulation. This implies the use of a simulation/integration lab.

The final action is that of implementing the selected compromises. Again system simulation will help the designer to iron out implementation details that might have gone unnoticed.

Designing Updates/Upgrades

Before the end of their useful service life, avionic systems will undergo one or more updates/upgrades in order to meet evolving requirements. By definition, these updates/upgrades imply Bottom-Up design, i.e. the usage of existing components to design a system that will meet specified requirements. However, they will be effected on totally integrated systems, not on a set of independent components. Furthermore, the resulting system must also be a totally integrated system. This means that Top-Down design must be used, to the fullest extent possible, for the design of updates/upgrades. Here, there is a nuance to be given to the word "top". The "top" to be used is not the top of the avionic system but rather the highest level at which the update will affect the system. The designer will start at this level, just as if it was the top

of a system to be designed and design the update/upgrade level by level until the lowest level possible is attained. Through this process however the designer encounters unchangeable constraints (i.e. space restrictions weight restrictions pre-defined bus structure etc.) such that the design process ends up being an attempt at Top-Down design but is bounded by Bottom-Up design restrictions. A less than optimal situation to be avoided where possible.

The design itself implies the evaluation of the impact that the new requirements have on the system. This evaluation is done by finding all new interactions to be brought into the system as well as the changes to be made to the existing interactions. As is the case with the design of the whole avionic system, the designer needs the help of system simulation to achieve his goals. However there is now a choice as to the manner in which to do system simulation. The first method is to monopolize an aircraft for the duration of the update/upgrade development program and use it as a simulator. This is not a good solution because the present situation is one of a decreasing number of aircraft having to meet increasing operational requirements. In such cases, the loss of even one aircraft is a significant handicap. Therefore, this solution, being detrimental to the operational capability of an aircraft fleet, is unacceptable.

The second, and preferred method, is to use a simulation/integration lab facility. This solution is not detrimental to the operational capabilities of an aircraft fleet. A simulation/integration lab facility

represents a significant investment. If such a lab facility was used for the original design of the system then the investment has already been made. If not the most cost-effective way to implement such a facility is to buy an existing avionic system and use it to produce a dedicated simulation/integration lab. The price paid for this solution is a loss in flexibility (i.e. impossibility of designing updates for different avionic systems) but when money is in short supply, such flexibility is not so important.

5.3.3 Requirement Specification

"The specification of the avionic system requirements is the basic document - containing the technical points of reference for the system to be designed" [6.p8-2]. Thus, the quality of the resulting design is highly dependent on, among other things, the requirement specification. Badly specified or ever-changing requirements inevitably lead to unsatisfactory designs regardless of the quality of the designer or the design methodology. This is especially true for the implementation of a totally integrated system through a Top-Down design methodology. In this case, the requirement specification is the basic document from which the interactions within the system can be identified. A nebulous or ever changing document will result in nebulous or ever changing interactions thus preventing the efficient application of the design methodology.

The requirements specification is not produced by the designers in industry but rather by military personnel (operators) and government of-

officials. It is these people who must be made aware of the importance of well defined requirements specification. In our world of rapidly changing technology, the enemy's perceived threat is continually evolving. The operators thus have a tendency to change their requirements specification such that it will reflect the ever changing threat. This can only result in a system that is almost impossible to design completely. It is very difficult and time consuming to accurately evaluate the effects of changing requirements such that the designer generally has to start from scratch every time his requirements change. Obviously, this statement is dependent on the level and impact of the changing requirements but, more often than not, the requirements that do change are ones of far reaching impact.

The solution to this problem is to produce a requirements specification that will not change throughout the design process. Such requirements will result from a thorough and precise evaluation of the threat anticipated for the time frame of the service life of the designed system. Military and government personnel must be made aware that an accurate evaluation of the anticipated threat is crucial for the successful development of a suitable avionic system.

The pessimists, or realists (too often both are synonyms), will hastily point out that the proposed solution is not a panacea for all ills. Two problems still remain. First, it is virtually impossible to have absolute accuracy in the projection of the perceived threat. Secondly, producing a requirement specification for a highly complex avionic system

is by no means trivial, errors have a habit of going unnoticed until the design is well underway. The solution to both problems is simply to be aware that these problems exist and to do everything humanly possible to minimize them. Extensive use of automated design analysis tools and data-base management aids can help produce close to an ideal requirement specification.

5.3.4 Standardization Issues

In Chapter 4, section 4.3.5, it was mentioned that standardization implementation must be restricted to technology transparent standards in the initial stages of development followed by a phased introduction of hardware standards. This can be accomplished through a proposed [8] program structure implying significant changes in the system development process. This program structure also aims to simplify logistics and training of maintenance personnel [8,p657]. Figure 5.4 describes the proposed program structure.

The specifics of the program are as follows [8,p657]. Initial hardware standardization studies identify specific candidates with positive life cycle cost payoffs. These are then communicated to subsystem designers prior to engineering model design and test. The engineering model test cycle includes the evaluation and implementation of the design changes needed to incorporate the standardization features. Final hardware standardization decisions are reached prior to the production prototype design and test phase and incorporated along with design

3 3

OF / DE

Chapter 3, Section 3.3.4, outlined that the integration problems encountered in the design of new avionic systems arise because, on the one hand, the existing integration approach is not entirely suitable for the highly complex systems currently used, and, on the other hand, there is no single clear-cut, well-defined integration approach accepted by industry as a whole. This means that these integration problems can be solved by the following two actions. First, implement an integration approach that will satisfy all the requirements of present and future avionic systems and, secondly, implement a standardized integration approach acceptable to industry as a whole. Chapter 4, Section 4.2, concluded that the only approach satisfying these requirements is the Total Integration approach.

As for the integration problems of developing updates/upgrades⁶ of existing systems, there are two possible ways of solving them. The trivial solution is to not do updates/upgrades. However, Chapter 3, Section 3.1.3, determined that updates/upgrades of avionic systems are inevitable. Therefore, this leads to the second, and only viable, solution: a standardization program. Chapter 5, Section 5.2.3, concluded that proper standardization allows the addition of new capabilities without integration problems. Thus, the application of a suitable standardization program will alleviate the integration problems of developing updates/upgrades. Therefore, the "integration problem" is solved by the application of the Total Integration approach and by the implementation of suitable standardization.

modifications found desirable from the engineering model evaluations. This program structure permits the selection of standards concurrent with state-of-the-art design methodologies and results in the timely application of the selected standards.

5.3.5 Summary

The implementation of the Top-Down design methodology is dependent on several factors. The first one is the development of a clearly stated and accurate requirement specification. The writers of this document must strive to produce as close to a perfect specification as possible, such a document being a pre-requisite to the successful implementation of the Top-Down methodology.

The second factor is that of standardization. The development process must allow for the timely application of the required standardization. This can be accomplished through a proposed standardization program structure which is incorporated within the design process.

The last factor is that of support systems. The development of advanced systems will be possible only through the use of a simulation /integration lab facility. This lab facility will be used in several areas of the development, namely, the development of entirely new systems, the development of updates/upgrades, and, the demonstration of the changes being implemented. A simulation/integration lab is a very expensive support system but it is nevertheless required.

5.4 CHAPTER SUMMARY

This chapter has attacked the problem of implementing the FIHA system. Such an implementation is far from being straightforward. Extensive research and development is required in the areas of artificial intelligence (expert systems) and information transfer system (contention protocol). further standardization is required in the areas of ITS, ISA, and HOL with the extent of standardization being restricted, a standardization program must be incorporated in the design process, a simulation/integration lab facility must be provided, a two-level maintenance concept needs to be proven, and government and military personnel must produce adequate requirement specifications.

What this boils down to is that, for the FIHA system to become a reality, government and industry must act immediately to allocate the proper resources (money, manpower, and equipment) at the appropriate place and to change the established development and support procedures to the new procedures. Since this process takes time, it is imperative that the appropriate actions be taken immediately. Only thus will the FIHA system be implemented in time for the next generation of avionic systems.

CHAPTER 6 CONCLUSIONS

INTRODUCTION

The next generation avionic system must be characterized by a fully integrated, hierarchical system architecture. Furthermore, this system requires an implementation philosophy based on a Total Integration concept. The mechanism of such an implementation must be a Top-Down design methodology. Given these claims, several implications follow. These implications are pre-requisites for the achievability of the FIHA system and also govern its implementation. Before analyzing the implications of the claims above, their correctness must be established.

CLAIMS

Chapter 4, Section 4.1, examined the issue of the next generation avionic system architecture. Having taken into account such factors as requirements, constraints, and evolutionary trends, it was concluded that the present day distributed architecture will not be suitable for the next generation of avionic systems. It was also determined that the Hierarchical architecture is the only architecture that will satisfy all the requirements of the next generation avionic system architecture.

Section 4.2 investigated the integration issue. Chapter 3 pointed out that the current Planned Integration approach is not suitable for present day systems and thus, will not be suitable for future systems.

This being the starting point, Section 4.2 analyzed all the factors determinant in the choice of the next generation system integration approach, and concluded that the Total Integration approach is the only approach that meets all the integration requirements of the next generation avionic system. This conclusion implies two things. The first is that not only must the next generation architecture be hierarchical, but it must also be a fully integrated architecture. The second point is that the implementation of the system must be based on a Total Integration concept. These two points seem very similar but they are nevertheless different. One is a characteristic of the avionic system while the other is an implementation philosophy. Their similarity lies in the fact that the latter implies the former and vice versa.

Finally, Section 4.3, considered all of the requirements, constraints, evolutionary trends, and other factors imposed on the next generation system design methodology and arrived at a set of essential characteristics for it. These characteristics, combined with the requirement to support a Total Integration philosophy, inevitably lead to the conclusion that the Top-Down design methodology, described in Chapter 2, must be the next generation design methodology.

IMPLICATIONS

The previous discussion establishes the thesis claims. It is now possible to examine the implications of these claims.

Demands

The first implications are three classes of demands imposed on the implementation of the FIHA system. The first class deals with requirement specifications.

Chapter 5, Section 5.3.3, examined the implementation demands of the Top-Down methodology and concluded that a crucial pre-requisite for the application of this methodology is the production of complete and static requirement specifications. Since the next generation avionic system must be designed through the Top-Down design methodology, complete and static requirement specifications are crucial for its achievement. These specifications are not produced by industry, but by government and military personnel. These persons must therefore produce complete and static requirement specifications if the next generation avionic system is to be achieved.

The second class of demands deals with the solution to the "integration problem". Chapter 3, Section 3.1.4, stated that the "integration problem" has the effect of making the integration phase of avionic development programs an unacceptably complicated, lengthy, and costly phase. The next generation avionic system will become a reality if this "integration problem" is solved. Section 3.1.4 also stated that the "integration problem" has two aspects. The first one is that of developing new avionic systems while the second aspect is that of developing updates/upgrades of existing systems.

Chapter 3, Section 3.3.4, outlined that the integration problems encountered in the design of new avionic systems arise because, on the one hand, the existing integration approach is not entirely suitable for the highly complex systems currently used, and, on the other hand, there is no single clear-cut, well-defined integration approach accepted by industry as a whole. This means that these integration problems can be solved by the following two actions. First, implement an integration approach that will satisfy all the requirements of present and future avionic systems and, secondly, implement a standardized integration approach acceptable to industry as a whole. Chapter 4, Section 4.2, concluded that the only approach satisfying these requirements is the Total Integration approach.

As for the integration problems of developing updates/upgrades of existing systems, there are two possible ways of solving them. The trivial solution is to not do updates/upgrades. However, Chapter 3, Section 3.1.3, determined that updates/upgrades of avionic systems are inevitable. Therefore, this leads to the second, and only viable, solution: a standardization program. Chapter 5, Section 5.2.3, concluded that proper standardization allows the addition of new capabilities without integration problems. Thus, the application of a suitable standardization program will alleviate the integration problems of developing updates/upgrades. Therefore, the "integration problem" is solved by the application of the Total Integration approach and by the implementation of suitable standardization.

The Total Integration approach must be supported by appropriate implementation mechanisms. Chapter 5, Section 5.2.1, analyzed the problem of implementing the Total Integration approach and concluded that the Top-Down design methodology is the proper implementation mechanism for the Total Integration approach.

This implementation scheme implies changes to the avionic system development program structure. Traditionally, the system integration phase of an avionics development program is implemented concurrently with system testing, after the avionic system has been designed and developed. However, the application of the Total Integration approach through Top-Down design, requires that the integration phase be moved into the design phase. Such a change can only be realized if the avionic system development program structure is changed to accommodate system integration as a part of the initial design phase.

The third class of demands concerns the need for simulation/integration lab facilities to support all aspects of avionic development programs.

Chapter 5, Section 5.3.2, examined the issue of developing new avionic systems. It was concluded that these systems can only be developed through the use of system simulation and that this system simulation can only be achieved through the use of a simulation/integration lab facility. Section 5.3.2 also analyzed the development of updates/upgrades of totally integrated systems in terms of the support

tools required for their achievement. As was the case with new avionic systems, system simulation is required, and this simulation can only be cost-effectively achieved through the use of a simulation/integration lab. Finally, Section 5.3.2 looked at the problem of implementing a system that is significantly different than the previous one. The high cost of avionic development programs, coupled with the pre-defined, unchangeable organizational roles of a PMO, results in a strong resistance to changes on the part of government and industry. Thus, significant changes have a tendency to occur over several incremental steps. Since the FIHA avionic system is significantly different from current systems, and the next generation is only one incremental step away, the acceptance of changes must be accelerated. A demonstration of the required system changes will accelerate their acceptance, and the most cost-effective way to achieve such a demonstration is through the use of a simulation/integration lab. If such a lab facility is not used for demonstration purposes, the implementation of the FIHA system will occur over several incremental steps and thus will not be achieved before approximately year 2010.

Constraints

Standardization is a major issue in the development of the proposed avionic system. Chapter 5 points out that it is required for the implementation of a Hierarchical architecture, for the solution to the integration problem, and for the proper application of the Top-Down design

methodology. However, there are constraints regarding such standardization.

Chapter 3, Section 3.1, examined avionic system requirements and it was concluded that there is a requirement for continually increasing mission capabilities. It was also observed that such a requirement has traditionally been fulfilled by the application of new technology products, as it should be. However, such new technology insertion can only be effected if the standardization is as high level as possible, since low level standardization prevents the insertion of new technology.

Chapter 5, Section 5.2.3, stated that the solution to the integration problems resulting from the development of updates/upgrades to avionic systems, is to apply a maximum of standardization, as low level as possible, to such systems. However, it has just been observed that technology insertion requires standardization to be as high level as possible. Therefore a compromise, regarding the level of standardization to be applied, must be reached. Such a compromise implies that there is a limit to the amount of standardization to be applied to avionic systems.

Conclusions

The previous discussion serves to outline the characteristics and development mechanisms of the next generation avionic system, and establishes the demands and constraints imposed on its implementation. As a result of this, several conclusions can be drawn.

The first conclusion concerns the achievability of a hierarchical architecture. Chapter 5, Section 5.1, outlined several areas of concern regarding the implementation of such an architecture. The first factor to consider is the information transfer system. The requirement for high bandwidth buses (up to 60 MHz and more) implemented via fiber optics should be met. What is less certain is the development and approval of suitable contention protocols. Although this has the impact of restricting the use of Hierarchical architectures in the areas of size and functions [39], it does not render it unachievable.

The requirements for standardization (data bus, ISA, HOL) have a high probability of being met. DoD is actively working in these areas and appropriate MIL-STDs and DoD instructions have already been issued. The two-level maintenance concept is also being actively pursued. The Future Fighter Avionics Demonstration (FFAD) program, a near term implementation of portions of a Pave Pillar architecture, is aiming for a late 1987 test flight date with the following objectives [37]:

- Develop and demonstrate an advanced architecture based on common modular hardware.
- Demonstrate a reduced intermediate shop role for maintaining avionics, and
- Propose the types of specifications and standards necessary to procure modular based subsystems in a competitive environment.

This is very encouraging and even though there are always several years delay between a prototype demonstration and full-scale, industry-wide acceptance, it is anticipated that this concept can be implemented before the end of the 1990s.

The area least likely to be developed in time is that of artificial intelligence. This is a new technology for the commercial and scientific communities and is even newer to the avionics world. As with any new technology, its potentials, although tremendous, are somewhat shaded by the huge difficulties of its implementation. The major difficulty is that it requires extensive research and development, which implies significant immediate investments in terms of time and money. However, the advantages are worth the efforts, and AI technology can be expected to progress steadily. The drawback is that the rate of progression will not allow the implementation of the full capabilities of AI and Expert Systems (ES) technology by the year 2000. Chapter 4, Section 4.1.3, stated that a Hierarchical architecture requires the use of artificial intelligence (i.e. ES) to allow the operator to become a system manager. The restriction in available ES capabilities will translate into reduced performance for the Hierarchical architecture.

The conclusion to this discussion is that the Hierarchical architecture is achievable but will not reach its full potential for the next generation avionic system.

The next conclusion concerns the possibility of implementing the Total Integration approach by the end of the 1990s. Chapter 5, Section 5.2, determined that the achievability of the Total Integration approach is dependent on three factors. The first one is the application of high level standardization. Efforts in this area have been initiated and are progressing satisfactorily such that the required standardization should be in place by the early 1990s.

The second factor is the existence of a multileveled, multiplexed ITS with a contention protocol. Such an ITS is not presently in existence but efforts have been initiated towards its development such that it has a high probability of being implemented by the mid-1990s. Finally, the last factor considered is the possibility of performing Top-Down design. Such a methodology already exists, therefore it will be possible to perform Top-Down design for the next generation of avionic systems.

The final point to be made is that truly total integration may not be achievable regardless of the factors previously discussed. This is because total integration requires the identification of "all" the interactions within the system. Today's systems have grown so complex that the actual identification of "all" the interactions is less than certain let alone the achievement of suitable trade-offs for each identified interaction. Even if one assumed that there existed an ideal design methodology which would allow the design for all interactions on a global optimization criteria, the following realities would prevail, [3]: the sys-

tem would be open-ended; no one is that smart, and, if someone was, nobody would believe him; and, there would be no practical way to implement the system. In the light of all this, the conclusion to be drawn is that although truly total integration may not be possible, system designers can work toward achieving as much integration as possible remembering that "the system engineering process is a mechanism for reaching closer to an ideal of total integration" [3.p6].

At this point, it is worth looking at the integration problem again. It was previously established that the solution to the "integration problem" is two-fold. The first part of the solution is to implement a Total Integration approach. The second part of the solution is to implement a suitable standardization program to alleviate the problems of an update/upgrade implementation. These problems can be ultimately eliminated through the application of total standardization. However, it has been established that there is a limit to the amount of standardization that can successfully be applied to next generation avionic systems. The effect of this limit is that the "integration problem" cannot be eliminated completely, unless updates/upgrades are eliminated. With updates/upgrades, the integration problem can only be minimized proportionally to the amount of standardization applied to avionic systems.

The next point to consider is the possibility of designing the next generation of avionic systems through the Top-Down design methodology. Chapter 5, Section 5.3, stipulated that the achievability of Top-Down

design is dependent on three factors. The first one considered is that of the requirements specification. All that is needed in this area is for the personnel involved with these specifications to be aware of their importance and of the tools available for their write-up. This should not be a big problem but it is crucial to the effective implementation of the design methodology.

The second factor is that of changing the avionic system development program structure. Two basic changes are required. The first change is to include the integration approach as a part of the design phase and the second is to include the standardization program proposed in Chapter 5, Section 5.3.4. The changes to the development program structure are rather straightforward but their implementation will require careful thought and planning. However, given the proper resources, it should be possible.

The final factor is that of the support tools. It was shown that the development of the proposed avionic system will require extensive use of a simulation /integration laboratory. This is a major endeavor in terms of the cost and resources involved. However, a few such facilities already exist; it is now a matter of using them appropriately. Where these facilities do not exist, they will have to be built. Although this represents a significant investment, it is nevertheless required. To neglect this investment is to make the larger investment in an entire weapon system, a wasted effort.

The conclusion to be drawn is that the Top-Down design methodology is achievable. The major problem for its accomplishment in time for the next generation avionics is the availability of a simulation/integration lab. If one has to be built, it should be started immediately to be ready for the next generation avionic system development.

A final point to be analyzed is the effect of the acceptance of changes on the achievability of the FIHA avionic system as a whole. This system is significantly different from the current, state-of-the-art systems. Chapter 5, Section 5.3.1, discussed the impact that the huge investments of avionic development programs have on the acceptance of changes. It was concluded that this acceptance will likely occur in several intermediate steps. This realization generated the requirement to accelerate the acceptance of changes as discussed in Section 5.3.2 where it was concluded that a demonstration of the changes will accelerate their acceptance. The point to be made here is that the investment is so big that such a demonstration will not totally overcome the resistance to changes which is part of the human nature. The effect of the remaining resistance to changes will be to delay the full implementation of the FIHA system beyond the year 2000.

The next 10 to 15 years should see the appearance of weapon systems with partial implementations of the FIHA avionic system. What must be remembered is that the operational requirements currently being established for the next generation avionic system can be fully met only by

the full implementation of the FIHA avionic system.

REFERENCES

- [1] AVERILL, G.L., "An Approach To System Integration (BASIC)", Proceedings of the 2nd Digital Avionics Systems Conference, 1977, pp 137-145.
- [2] BEUM, C.O., LEVIN, E., "Distributed Avionics Information Systems", Proceedings of the National Aerospace and Electronics Conference, 1974, pp 78-84.
- [3] BOTHA, D.G., "Integrated Avionics: Concepts and Concerns", Proceedings of the 4th Digital Avionics Systems Conference, 1981, pp 1-9.
- [4] BOWEN, B.A., BROWN, W.R., VLSI Systems Design for Digital Signal Processing, Volume 2, Prentice Hall, 1985.
- [5] BOWEN, B.A., et al., "Analysis and Evaluation of VLSI/VHSI Technology on High Performance Digital Signal Processors - Final Report", prepared under Contract 2SU80-00333 by Microprocessor Systems Development Laboratory, Carleton University, Ottawa, 1982.
- [6] BRAMMER, K., WEIMANN, A., "Functional Versus Communication Structures in Modern Avionic Systems, AGARD Conference Proceedings No 303, 1981, pp 8.1-8.11.
- [7] BROWN, W.R., "Very High Speed Integrated Circuits for the U.S. Military", Department of Systems Engineering and Computing Science, Carleton University, Ottawa, 1979, unpublished.
- [8] BURNS, R.C., ROSENKOETTER, E.A., "Advanced System Integration Demonstration (ASID) for the 1990s; Proceedings of the National Avionics and Electronics Conference, 1983, pp 651-658.

- [9] BUSSEL, J.F., "A Concept of Digital Avionics", AGARD Conference Proceedings No 43, 1968, pp 283-295.
- [10] CALLAWAY, A.A., "Trends in Digital Data Processing and System Architecture", AGARD Conference Proceedings No 272, 1979, pp 3.1-3.5.
- [11] CALLAWAY, A.A., "The Influence of Digital Techniques on System Architecture", Paper Presented at the Royal Aeronautical Society Symposium on "Digital Avionics - Promise and Practice", March 1980.
- [12] CRENSHAW, J.H., "Federated vs Integrated Computer Systems", AGARD Conference Proceedings No 68, 1970, pp 4.1-4.14.
- [13] CROSSGROVE, W.A., SMITH, L.A., "Distributed Systems: The Next Integration Method, Proceedings of the 2nd Digital Avionics Systems Conference, 1977, pp 45-53.
- [14] DICKEY, F.J., POHLMANN, L.D., "Avionics as an Expert Systems Domain", Proceedings of the National Aerospace and Electronics Conference, 1984, pp 163-168.
- [15] DUKE, P.A., "A Practical Approach to the Design of a New Avionic System", AGARD Conference Proceedings No 343, 1983, pp 25.1-25.11.
- [16] EDWARDS, J.A., HUBANS, F., Jr., "A Hierarchical Network For Avionic Systems", Proceedings of the National Aerospace and Electronics Conference, 1978, pp 129-136.
- [17] ENGLAND, G.R., "System Architecture: Key to Future Avionics Capabilities", AGARD Conference Proceedings No 343, 1983, pp 1.1-1.6.
- [18] GANGL, E.C., "Air Force Findings and Recommendations on Digital Aircraft Avionics", Proceedings of the National Aerospace and Electronics Conference, 1974, pp 72-77.

- [19] GANGL, E.C., "Digital System Integration Techniques, The Key to Affordable Avionics", Proceedings of the 2nd Digital Avionics Systems Conference, 1977, pp 82-88.
- [20] GIFFORD, C.A., "A Military Standard for Multiplex Data Bus", Proceedings of the National Aerospace and Electronics Conference, 1974, pp 85-88.
- [21] GREGORY, J.G., "System Architecture for Aircraft and Electrical Systems", Proceedings of the National Aerospace and Electronics Conference, 1973, pp 30-36.
- [22] HALKO, C., JADICO, T., "Basic Avionic System Integration Concept", Proceedings of the 3rd Digital Avionic Systems Conference, 1981, pp 21-27.
- [23] HAROLD, G.M., "Trends in Avionics Systems", Proceedings of the National Aerospace and Electronics Conference, 1969, pp 7-12.
- [24] HAYES, J.P., Computer Architecture and Organization, McGraw-Hill, 1978.
- [25] KLOS, L., "Pave Pillar Avionics - An Architecture for the Future", Proceedings of the National Aerospace and Electronics Conference, 1984, pp 156-162.
- [26] LEEPER, K.R., "The Missing Link For Advanced Avionics Systems Executives", Proceedings of the 4th Digital Avionics Systems Conference, 1983, pp 2.6.1-2.6.6.
- [27] MARTIN, J.T., "The Effect of Increasingly More Complex Aircraft and Avionics on the Method of System Design", AGARD Conference Proceedings No 303, 1981, pp 3.1-3.5.
- [28] MASON, R.C., et al., "DAIS: Design and Performance", Proceedings of the 2nd Digital Avionics Systems Conference, 1977, pp 129-136.

- [29] McCUEN, J.W., "Next Generation Aircraft Will Require Hierarchical/Multilevel Information Transfer Systems", AGARD Conference Proceedings No 303, 1981, pp 31.1-31.8.
- [30] MORGAN, L.F., "Challenge in Software", Proceedings of the Electronics and Aerospace Systems Conference, 1976, pp 111.A-111.H.
- [31] MORGAN, L.F., "Advanced Avionic Architectures for the 1980's: A Software View", Proceedings of the 3rd Digital Avionic Systems Conference, 1979, pp 13-18.
- [32] NAGLAK, L.A., NORR, W.M., "The Application of Structured Design and Distributed Techniques to Avionics Information Processing Architectures", AGARD Conference Proceedings No 251, 1978, pp 16.1-16.16.
- [33] OGI, S.K., et al., "Distributed Digital Processing Architectures for Military Avionics", Proceedings of the National Aerospace and Electronics Conference, 1983, pp 25.3.1-25.3.6.
- [34] OLDFIELD, D., SALMON, L.T.J., Dr., "A Future System Design Technique Based on Functional Decomposition, Supported by Quantifiable Design Aims and Guidelines, for Minimum Maintenance Costs", AGARD Conference Proceedings No 343, pp 5.1-5.6.
- [35] PRICE, R.C., "Integration Plus Federation", Proceedings of the National Aerospace and Electronics Conference, 1969, pp 3-6.
- [36] SCARPINO, F.A., WEBER, J.G., "A DAIS Architecture Utilizing Microprocessors", Proceedings of the National Aerospace and Electronics Conference, 1976, pp 82-88.
- [37] SCOTT, D.M., "Future Fighter Avionics Demonstration Program", Proceedings of the National Aerospace and Electronics Conference, 1984, pp 1343-1347.
- [38] SMYTH, R.K., Dr., "State of the Art for Digital Avionics & Control", AGARD Conference Proceedings No 272, 1979, pp 1.1-1.20.

- [39] WHITTING, J.H., "Military Aircraft Avionics in the 1980's", Proceedings of the Seminar on Standardization in Military Avionic System Architecture, 1980, pp 14-20.

- [40] WILENT, C.E., "Directions in Avionic Data Distribution Systems", Proceedings of the 5th Digital Avionic Systems Conference, 1983, pp 12.6.1-12.6.6.

- [41] WEISSMAN, S., "Avionics Designer's View of the System Integration Problem", AGARD Conference Proceedings No 62, 1969, pp 18.1-18.10.

- [42] ZEMPOLICH, B.A., "A Tutorial on Distributed Processing in Aircraft/Avionics Applications", AGARD Conference Proceedings No 303, 1981, pp 4.1-4.11.

- [43] ZEMPOLICH, B.A., "Integrated Digital Avionic Systems - Promise and Threats", Astronautics and Aeronautics, October 1983, pp 46-53.

END

1	4	0	4	8	6
---	---	---	---	---	---

FIN