AFCEA

THE ARMED FORCES COMMUNICATIONS
AND ELECTRONICS ASSOCIATION

Command & Control (C^2) Systems Acquisition Study

Final Report

1 September 1982

BLANK PAGES IN THIS DOCUMENT WERE NOT FILMED

ARMED FORCES COMMUNICATIONS AND ELECTRONICS ASSOCIATION

Publishers of SIGNAL - Official Journal of the AFCEA
One Skyline Place
5205 Leesburg Pike, Falls Church, Virginia 22041
Area Code (703) 820-5028

November 8, 1982

Hon Richard D. DeLauer
Undersecretary of Defense
Research & Engineering
Room 3E1006, The Pentagon
Washington, DC 20301

Dear Dr. DeLauer:

I am pleased to forward the attached Final Report of AFCEA's Command & Control (C2) System Acquisition Study.

You were briefed on highlights of the study on April 28, 1982 by the Study Chairman, Bob O'Donohue of The BDM Corporation. As you will recall, a key finding is that the evolutionary acquisition (EA) approach is appropriate for most C2 systems which augment the decision-making and decision-executing activities of operational commanders and their staffs. Also, we provided you recommended changes to DoDI 5000.2 and a Draft Implementation Memorandum for C2 system acquisition. We strongly believe that the unique nature of command and control mandates that separate acquisition policy be required for C2 systems.

The enclosed Final Report amplifies the summary briefed to you. In addition to changes required in DoD's acquisition policies and practices, the report included discussions about when to follow EA, types of EA strategies, steps necessary in the EA approach, changes required in the relationships among the participants in the C2 system acquisition process, and the importance of using architecture and development practices designed to accommodate change and insertion of new technology. The Final Report should be especially useful to managers and staff officers who are involved in the acquisition of C2 systems. AFCEA plans wide distribution to C3I leaders in DoD and industry.

We deeply appreciate the splendid support the Study Team received from John C. Cittadino and Richard G. Howe of your office throughout the study. They focused our efforts in the right directions and gave invaluable assistance in arranging meetings with key C2 players.

As you know well, completion of a study is only the beginning of the actions necessary to improve C2 system acquisition and operational capabilities. AFCEA will help any way it can with the follow-on actions.

Sincerely,

Jon L. Boyes

Dr. Jon L. Boyes
Vice Admiral, USN (Ret)
President

JAN 1 [illegible] 1983

A

JLB/dc
Enc.

AFCEA
ARMED FORCES COMMUNICATIONS & ELECTRONICS ASSOCIATION
COMMAND & CONTROL (C²) SYSTEM ACQUISITION STUDY
FINAL REPORT — 1 SEPTEMBER 1982

EXECUTIVE SUMMARY

The March 1980 version of DoD Instruction 5000.2, "Major System Acquisition Procedures," notes that the characteristics of certain types of command and control (C^2) systems are sufficiently different from weapon systems that these types should be acquired in most cases via an evolutionary approach involving special management procedures, rather than by the traditional approach. The specific C^2 system types that most require such an evolutionary acquisition (EA)[1/] strategy are those which involve or augment the decision-making and decision-executing activities of operational commanders and/or their staffs, including those which constitute automated management information or intelligence information/exploitation and management/force planning and control aids of some type (hereafter referred to as C^2 systems).

These types of systems: (1) have numerous complex and changing external and internal interfaces, often of an inter-Service or multinational nature; (2) involve operational requirements, user acceptance criteria, and measures of worth which cannot adequately be specified in advance, and which are highly dependent on the specific doctrine, procedures, threat, geographic constraints, mission scenarios, and management approaches of specific mission users, and hence are subject to relatively frequent change; and (3) are software-dominated, with the software highly interactive with the cognitive processes of specific (or classes of) mission commanders and their staffs at multiple organizational levels. (As a result, communications systems and sensors normally would be excluded).

1/ Evolutionary acquisition is a system acquisition strategy in which only a basic or "core" capability is acquired initially and fielded quickly, based on a short need statement that includes a representative description of the overall capability needed and the architectural framework within which evolution will occur. Subsequent increments or "blocks" are defined sequentially, based on continuing feedback provided from lessons learned in operational usage, concurrent evaluation of adequacy of hardware/software configuration, and judgments of improvements or increased capabilities that can result from application of new technology, where feasible.

A Study Team, consisting of representatives of member companies of the Armed Forces Communications and Electronics Association (AFCEA), all of whom were experienced in C^2 system acquisition, test and support, reviewed the acquisition process normally followed for these types of C^2 systems and the degree to which the EA policy for C^2 systems has been implemented. Visits were made and discussions were held with over 200 representatives of the significant participants in the C^2 system acquisition field, including OSD and the various DoD Component headquarters, users/user surrogates, providers and independent testers. An extensive literature search was conducted (including literature on numerous commercially-developed decision support systems), and over a dozen pertinent DoD programs were examined, including interactions with past and present program personnel, to identify "lessons learned." During the course of the study, the Study Team interacted with a DoD Advisory Group, consisting of senior representatives of the Services, OJCS and DCA. The Study results also were reviewed by a Senior Review Group appointed by the AFCEA Board of Directors and has the endorsement of AFCEA.

The Study Team found that the application of EA within DoD is spotty, partly because the concept of evolutionary acquisition is not well understood, and partly because of resistance to the special management procedures and changes in organizational relationships which are required. In this regard, the Team distinctly found that EA will not work on a "business-as-usual" basis. Many organizations and personnel involved with C^2 systems requirements determination/validation, PPBS (DoD's Planning, Programming, and Budgeting System), procurement, the "ilities," development, test, and support have not as yet fully recognized that evolutionary acquisition is a different, but highly necessary, strategy for these types of C^2 systems. Until they do, the Study Team believes that the application of EA will continue to be inhibited. Among the most prominent deficiencies found was widespread lack of recognition of: (1) the need for continuous, cooperative interaction among the users, developers, testers, and supporters in a C^2 program, rather than the more "arms length" approach ordinarily used in the acquisition of other types of systems, and (2) the fact that the "requirements process" for these kinds of systems cannot be implemented on paper alone, but requires feedback from testing of the "core" capability, (and subsequent increments) as an integral part of (NOT separate from) an evolving "requirements process."

Synthesis of the study data affirmed that there is a much higher probability that useful improvements in C^2 capability will be fielded sooner and more often if EA is used and that EA should be required and facilitated for these types of C^2 systems (and that it may well be desirable in other C^3I and even weapon system programs at times). There have been too many expensive failures in C^2 programs acquired via the "traditional" approach to do otherwise.

Finally, but most importantly, the Study Team determined that for EA to achieve its full potential for accommodating change in a manner responsive to user needs and without deleterious effects to system reliability, performance and costs, it must proceed within two concurrent types of architectural frameworks. The first addresses the operational theatre or mission-related military functions and tasks (e.g., detection, fusion, allocation) which a commander and his staff use to discharge their responsibilities. These functions and tasks are performed and supported by a number of systems including a C^2 system(s). This collection of systems or "system of systems" and organizational entities (too often treated in isolation) must be addressed architecturally as a totality with particular attention paid to inter-system interfaces, especially inter-Service and multi-National wartime interfaces.

Corresponding with the operational "system of systems" is a hardware/software infrastructure which also requires the interconnection and interoperability of a number of systems. This second type of architecture addresses the design and implementation of specific C^2 system hardware and software which provide system functions and capabilities (e.g., data base management, networking, display) necessary to support the military functions and tasks. At present, the International Standards Organization (ISO) open system interconnect (OSI) model, which has been implemented partially, is the most promising and most widely accepted approach.

These architectural frameworks which structure their respective "system of systems," must be modularly designed with sufficient built-in flexibility to facilitate growth and allow for the insertion of new technology with minimum negative impact on the existing system.

Specific major recommendations for actions DoD should take to facilitate successful acquisitions of these types of C^2 systems are:

(1) Change DoD (and corresponding Service and Agency) policy and procedures which fail to encourage, or which inhibit, the use or effectiveness of EA, and develop guidelines to facilitate its use.

- Revise DoDI 5000.2 (or issue a separate directive) and related DoD Component documentation to mandate EA as policy, and clarify its use for these types of C^2 systems, and require justification if an alternative acquisition strategy is proposed.

- Establish an intra-DoD task force to prepare an informal guide that amplifies this EA policy, including recognition of the fact that EA is not a single approach but a strategy encompassing a spectrum of possible approaches.

- Establish a significant program to educate all participants in the C^2 system acquisition process on the merits and basic tenets of evolutionary acquisition.

- Make those changes in OSD and DoD Component PPBS policy and practices which are needed to recognize the evolving nature of C^2 systems.

- Change the current approach to the requirements determination/validation process for these types of C^2 systems by abbreviating and expediting it, as a means of recognizing both that this process is continuous for these types of C^2 systems, and that feedback from testing them in the user's environment is the primary means both for refining and amplifying requirements for them and for evolving to the needed capability.

- Revise OSD and DoD Component procurement policies to reflect the special needs of these types of C^2 systems, including the manner in which solicitation of bids/proposals and contracting is accomplished.

- Give program offices the flexibility they need to adapt to EA.

(2) Alter the roles of participants in the C^2 system acquisition process as defined in OSD and DoD Component policy/regulation. Particularly, strengthen and assure continuous real user involvement.

- Recognize the real user in such policy--that is, the commander and his staff who have an operational wartime mission.

- Provide such real users with the resources (people, tools, funds, and facilities) needed to facilitate their increased, continuous participation throughout the entire requirements determination and acquisition processes.

- Change the current approach to testing and evaluating these types of C^2 systems by providing for (1) joint user/tester/developer T&E, in the user's environment, of a rapidly-fielded initial ("core") and subsequent incremental capabilities, and (2) joint user/tester determination of operational utility.

- Recognize the changed nature of the logistics and training functions under EA, which involves earlier integration of new technology, more flexibility for accommodating changes and field support of evolving systems.

(3) Assure the use of system architectures and development techniques that can accommodate growth, change, continuous user and developer learning, and the insertion of new technology with a minimum of redesign and impact on existing systems.

- Require DoD-wide employment of a layered systems interconnect reference model, such as the internationally-accepted ISO model, as the basis for developing interface and protocol standards.

- Expedite efforts to establish theater and other operational mission architectures within which individual C^2 systems can evolve.

- Enforce the use of proven development practices which can accommodate change (rather than maximizing initial system designs), such as: wider use of already-developed system software; High-Order-language programming; transportable application software; and rigorous software quality assurance;

- Use centralized configuration and integration management to assure interoperability;

- Establish and use permanent system design/support facilities, jointly operated by users, developers, supporters, and testers.

Based on the Study Team's analysis and data synthesis, AFCEA believes that if these recommendations are implemented, the following benefits will accrue to DoD:

- Measurably increased C^2 capability in the hands of users, achieved far sooner than if DoD waited for a one-time "total" solution, due to the incremental, user-oriented development approach.

- Greater user satisfaction with, and more rapid assimilation of, systems resulting from the evolutionary C^2 system acquisition process, as a result of his close and continuous coupling with the acquisition, and the smaller, more-frequently-fielded increments that he will receive.

- Reduced Government risk and exposure, since each increment is limited.

- Easier technology insertion, and hence reduced obsolescence of materiel in the field, due to an architecture and approach to design aimed at readily accommodating change.

- Longer useful life of C^2 systems, also resulting from an architecture that readily accommodates change.

Appendix A to this report is AFCEA's recommended change to the DoDI 5000.2 section on C^2 system acquisition. AFCEA strongly believes that the unique nature of command and control mandates that a separate acquisition policy be required for C^2 systems. Appendix B is a proposed memorandum for Under Secretary of Defense Research & Engineering signature, implementing the results of this study.

AFCEA has been most pleased with this opportunity to support DoD and will be glad to help further in implementing its recommendations, as desired by USDR&E.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

APPENDICES

CHAPTER I

INTRODUCTION

CHAPTER I

INTRODUCTION

A. BACKGROUND AND OBJECTIVES

Problems being encountered with the acquisition of command and control (C^2) systems over the years are well known and have been documented in numerous studies. There are many examples of cost growth, program delays, equipment deemed obsolete by the time it is fielded and general user dissatisfaction with systems when they finally are fielded. The Army's Tactical Operations System/Operable Segment (TOS/OS) program, the original version of the Navy's Tactical Flag Command Center (TFCC) and the Air Force Tactical Air Control Center Automation (TACC AUTO) program are but three examples that evidenced these problems, and which were cancelled as a result.

The 1978 Defense Science Board study of command and control system management[1/] came to several conclusions about these problems, one of which was that C^2 systems had "special characteristics," the most important of which is:

> "the need for adaptability to user (described as "Service unit commanders, CINCs, or the National Command Authority") needs, and for their evolutionary change over time."

The DSB also observed that:

> "a very large fraction of (C^2 system) development cost is in software rather than hardware" and therefore "acquisition procedures based on hardware have little *a priori* applicability to command and control systems."

1/ "Report of the Defense Science Board Task Force on Command and Control Systems Management," Dr. S. J. Buchsbaum, Chairman, U.S. DoD, July 78.

The DSB Chairman, in forwarding the report to the Secretary of Defense, recommended that DoD policy directives be issued that:

> "brings the using Commands very deeply into the development of their command and control systems and emphasizes the evolutionary character of command and control systems."

An Appendix was added to the DSB report proposing a DoD Directive 5000.XX, dedicated to acquisition of support systems for command and control.

This DSB recommendation on the acquisition of C^2 systems gained momentum as a result of the interest of the U. S. Air Force Electronic Systems Division and other major C^2 acquisition organizations and was reflected in a separate section (Section 13) in the March 1980 revision of DoD Instruction 5000.2, which took the Defense Science Board's position and made it into policy. The 5000.2 policy verified that certain kinds of C^2 systems were different from weapon systems and should therefore be acquired under an evolutionary strategy and other special management procedures, rather than the "traditional" approach.

Later in 1980, the Armed Forces Communications & Electronics Association's (AFCEA's) Board of Directors determined, at their annual meeting, that it was time for the association to assume a more significant role in resolving key C^3I issues by providing direct assistance to the Office of the Secretary of Defense and other senior military decision-makers. After due consideration of a number of alternative topics, it appeared that the most helpful work that AFCEA could do would be to review the command and control system acquisition policy as described in DoDI 5000.2 for validity, identify what impediments there were, if any, to implementation of this policy, and suggest ways the policy could be better articulated, so that program managers, HQ staffs and other organizations associated with C^2 system acquisition would understand how best to implement the policy. The study also would determine what systems within the rubric of "command and control" the policy best applied to, and which ones it didn't apply to. In short, the intent was to provide guidance on how program managers could find their way through the many "bureaucratic snares" that awaited them in their attempt to solve the problem of fielding new systems in a timely fashion, and make recommendations for changes in C^2 system acquisition policies and practices to facilitate C^2 system acquisition.

Stated simply, the study's objective was to gain answers to the following questions:

- Is evolutionary acquisition being employed and with what success?
- What impediments exist to implementation of evolutionary acquisition policy?
- What actions are required to improve C^2 system acquisitions?

Before AFCEA could initiate its study effort, there was a change of Administration. Study initiation was delayed until April 1981, when senior officials from AFCEA Headquarters met with Dr. Richard DeLauer and Dr. James Wade, the new Under Secretary of Defense for Research & Engineering (USDR&E) and his Principal Deputy, to determine whether they felt an AFCEA study of command and control system acquisition was of interest. Drs. DeLauer and Wade affirmed their interest in this topic and pledged to help implement AFCEA's recommendations when the study was complete.

The process of nominating and selecting Study Team members was completed in early June 1981. An initial agenda was then formulated and OSD help in obtaining pertinent background briefings was requested. The first Study Team meeting was held in July 1981, followed by a number of other meetings, analyses and visits with participants in the C^2 system acquisition process, culminating in a summary presentation and this report.

The remainder of this chapter describes Study Team composition, study approach, definitions of key terms used in the study and summarizes our information gathering and research. Chapter II summarizes the Study's Major Conclusions and Recommendations. Chapter III discusses the C^2 system acquisition process, the promise of evolutionary acquisition (EA) for C² system acquisition, impediments to the practice of EA and other issues related to the Team's first major recommendation--to mandate and facilitate EA. Chapter IV discusses the participants in the C² system acquisition process, emphasizing the critical role of the user in the case of C² systems. This chapter amplifies the issues related to the Team's second major recommendation--to alter the roles and relationships of these participants in order to improve the practice of EA. Finally, Chapter V discusses the third major recommendation and other issues related to the use of appropriate system architecture and development practices to enable successful EA.

Appendix A to this report is the AFCEA Study Team's proposed revision to the paragraph in DoDI 5000.2 that deals with command and control system acquisition. This appendix implements the conclusions and recommendations of this study as a proposed policy statement.

Appendix B is a draft memorandum for USDR&E signature, implementing the recommendations of this study.

Appendix C includes the revision to DoDI 5000.2 C² system acquisition policy currently (April 1982) under consideration for incorporation in DoDI 5000.2. AFCEA has reviewed this version, and for reasons given in Appendix C, believes that if it were to be approved as written, it would be a step backwards from the March 1980 version of 5000.2 currently in effect, let alone what AFCEA believes should be issued as policy for C² system acquisition. Appendix C also gives AFCEA's comments on the April 1982 "For Coordination" revision to DoDI 5000.2, regarding C² system acquisition policy. As will be developed later, AFCEA believes that the unique nature of command and control mandates that separate acquisition policy be required for C² systems.

Appendix D includes Section 13 of DoDI 5000.2 of March 1980. This is the original statement of C² system acquisition policy which presumably is still in effect until a revision to DoDI 5000.2 is approved.

Appendix E summarizes representative C^2 system programs reviewed by the Study Team.

Appendix F is a list of abbreviations and terms used in the report.

Appendix G provides amplifying material on C^2 system evaluation.

Appendix H provides further material on architecture.

Appendix I depicts the organizational elements involved in the acquisition process.

B. STUDY TEAM COMPOSITION

In formulating the Study Team, the objective was to have the Team represent a microcosm of the AFCEA Industry Membership. Therefore, people were selected from a range of company sizes, varying from large to small, and encompassed hardware, systems and professional services firms in the C^2 systems business.

The selection process began with requests from the AFCEA Chairman to over 100 of the AFCEA member companies (those stating background in C^2), to nominate candidates with suitable backgrounds. A selection committee, consisting of John Cittadino and George Salton of OUSDR&E/C^3I; MG Robert Edge, USAF (Ret), a member of the AFCEA Executive Committee; BG Kirby Lamar, USA (Ret), the AFCEA focal point for the study; and Robert O'Donohue of The BDM Corporation, the Study Chairman, selected the Study Team's members from over 60 candidates that were nominated.

Figure I-1 illustrates the composition of the AFCEA C^2 System Acquisition Study Team and its organization into subteams. The people on the Study Team came from top management, or were people who had been (or are) program managers and/or those who had hardware and software development, systems engineering, and/or analytic experience or who had been involved in testing or system support. Some Team members came from many years in military service such as MG "Hank" Stelling, USAF (Ret.) of Rockwell International, or had been senior DoD civilians, such as Dr. Norman Waks of MITRE and Bob O'Donohue of BDM. LTG John Cushman, USA (Ret.), provided helpful advice from a user's standpoint, as did Dr. David Alberts (MITRE) regarding technology and architectural issues, as well as in helping integrate the overall study.

During the course of the study, the Team interacted with a DoD Advisory Group, consisting of senior representatives of the Services, OJCS and DCA. Also, the Study Team briefing and report was reviewed by a Senior Review Group appointed by the AFCEA Board of Directors and has the endorsement of AFCEA.

Dr. DeLauer appointed John Cittadino, Director, Theater & Tactical C^2, OUSDR&E, as the Study Team's liaison with OSD, assisted by Richard Howe of that office.

When the Study Team was first organized, it defined a series of issues related to C^2 system acquisition which fell under three general categories: (1) those relating to the participants in the C^2 system acquisition process ("players"), (2) those relating to technology and architectural questions, and (3) those relating to the C^2 system acquisition process itself. Subteams were defined, as shown in Figure I-1, to address these issues. In addition, over a dozen basic DoD C^2 system programs were determined to be appropriate for study, and two-person teams assigned to research, analyze, and provide "lessons learned" from these programs ("cases"). Case data was then analyzed and synthesized by a "Cases" subteam. The literature also was reviewed to determine the experience and lessons that could be learned from commercial as well as military decision support system developments. Finally, a "Steering Committee," consisting of the subteam leaders, Kirby Lamar, and the Study Chairman, was created to provide overall study coordination.

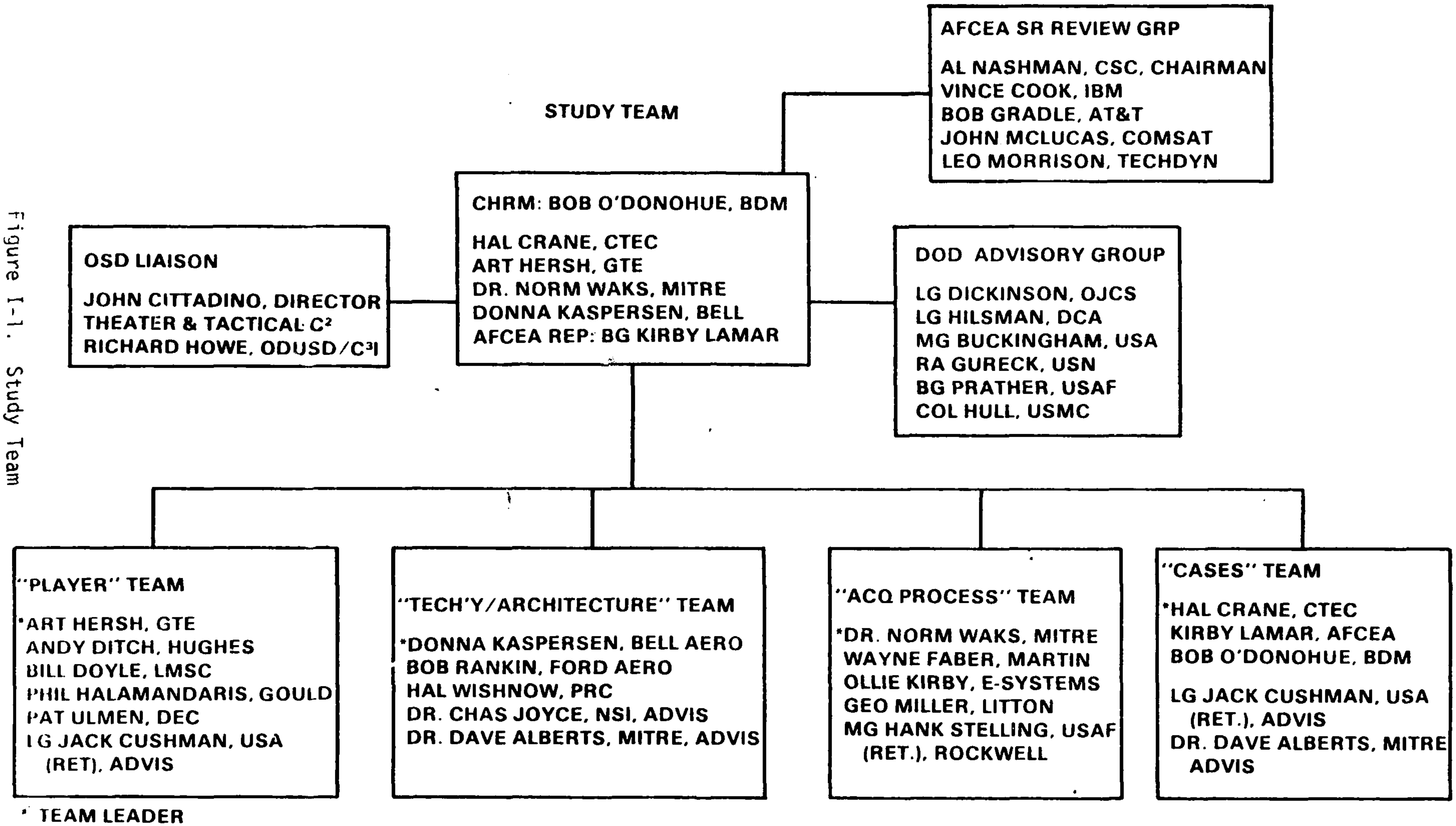


Figure I-1. Study Team

In interactions with the "players" in the process, the Study Team included as many of the overall Study Team membership as could be mustered for any one meeting, given their schedules.

C. STUDY APPROACH

The Study Team took a straightforward approach to organizing and executing the study. (Figure I-2).

- SCOPING & IDENTIFICATION OF KEY CONCEPTS & ISSUES DEFINITIONS
- INFORMATION GATHERING/RESEARCH
- CASE STUDIES
- ISSUE PAPERS
- ANALYSIS/SYNTHESIS
- BRIEFING
- REPORT
- DODI 5000.2 CHANGES

Figure I-2. Study Approach

First, the Team defined the scope of the study and set about agreeing on what at first were thought to be relatively-simple (but which didn't turn out to be) definitions. The definitions of "command and control system," "provider," "user," and "evolutionary acquisition" are discussed later in more detail.

The Team then embarked on a major information-gathering effort, including a literature search, a series of meetings with key government and industry personnel (totaling over 200 people) and an analysis of a series of case studies, (selected jointly with OSD and the Services) to derive "lessons learned." As will be discussed later, during this information-gathering effort, the Study Team met with a major cross-section of all participants in the C^2 system acquisition process to obtain informed judgments on past experiences and present practices in C^2 system acquisition.

Finally, members of the Study Team were divided into subteams to investigate a series of issues related to the study (discussed in Chapters III, IV and V), to perform case studies, to prepare a summary briefing and to write this report. In particular, they analyzed and synthesized all of the data and formulated the conclusions and recommendations of this report.

D. DEFINITIONS

This section discusses some basic terms used in the study. These include "command and control system," "provider," "user," and "Evolutionary Acquisition (EA)."

1. Command and Control (C^2) System

a. JCS Pub 1 Definition, Contrasted With Study Team's Meaning

The Team does not intend to revise the definition of a command and control system given in JSC Pub 1. The JCS Pub 1 definition has served DoD well over the years; but, in recent years, the community has tended to characterize what is defined as "C^2" in JCS Pub 1 by such terms as C^3I, C^3I^2, C^4I, etc.

The JCS Pub 1 definition of "C^2 system" is:

"The facilities, equipment, communications, procedures, and personnel essential to a commander for planning, directing, and controlling operations of assigned forces pursuant to the missions assigned."

Note that this definition excludes the Commander.

For the purposes of this study, the Study Team focused on what has been characterized in the literature as Decision Support Systems; i.e., under the Team's definition, C^2 systems are:

> "Those systems that augment the decision-making and decision-executing processes of operational commanders and their staffs. The central, essential ingredient in any command & control system is the commander or decision maker himself."

In the Study Team's view, those systems most appropriate to the evolutionary acquisition approach called for in DoDI 5000.2 are those that augment the decision processes of the commander, including those which constitute automated management information or intelligence information/exploitation and management/force planning and control aids.

Note that in such C^2 systems, the operational military commander is not merely the <u>user</u> of a C^2 system. He is very much a part, if not the dominant element, of the C^2 system. Thus, two significant differences between the JCS Pub 1 definition of a C^2 system and the type of system on which the study focused are:

- JCS Pub 1 really defines a "C^3I" system, to use currently-popular Pentagon terminology.
- The commander is <u>not</u> an explicit part of the C^2 system in the JCS Pub 1 definition.

For the remainder of this report, when a C^2 system is noted, it means the Study Team's definition, <u>not</u> the JCS Pub 1 definition.

At this point, it must be noted that restricting the scope of the investigation to decision support systems does not mean that the Team advocates not considering other C^3I systems for evolutionary acquisition, nor do we wish to imply that other aspects of C^3I (e.g., communications, sensors, electronic warfare) are less important. Rather the reduced scope is meant to emphasize that the use of evolutionary acquisition is crucial to success in decision support (C^2) system development.

b. Unique Characteristics of C^2 Systems

Command and control systems have several characteristics that demand an evolutionary acquisition (EA) strategy and related special management procedures. (EA is described on p. I-15).

C^2 systems are "mind extenders," not "muscle" or "sense" extenders--that is, C^2 systems provide decision support. Man (the commander and his staff) must therefore be considered to be an integral part of a C^2 system, and the "line item" system hardware and software acquired by the provider considered to be incomplete without the commander and his staff.

Any C^2 system designed today not only is "software intensive," but the software is highly interactive with the cognitive processes of the commander and his staff, as opposed to "weapon system" software, which is more "process oriented"--a fire control algorithm, a navigation algorithm, a flight control solution.

Also, the worth ("goodness" or "badness") of weapon system software is more objectively measurable in terms of its contribution to mission accomplishment. The contribution of automated decision aids to mission accomplishment is much more difficult to measure.

As developed in more detail in Chapter III, the main reason for using evolutionary acquisition is that detail requirements of C^2 systems are difficult, if not impossible, to articulate and quantify in advance:

- Little really is known about how an individual commander's cognitive processes work and, therefore, how these processes should best be supported by automation. Issues include:

  - How much information properly can be processed?

  - In what form should the information be presented for maximum comprehension and retention?

  - What should be the nature of computation aids which can be used to aggregate/simplify data?

- C^2 systems have more complex interfaces that change more rapidly and require more effort at interoperability.

- A more severe "cultural" (or language) barrier exists between users and providers of C^2 systems than exists between users and providers of ships, tanks, missiles, airplanes--for weapon systems, the user can relate to the meaning of a more-maneuverable fighter plane, or a more-accurate or longer-range air-to-air missile, and can visualize the potential impact on mission performance more easily. Trying to understand, for example, what distributed microprocessor technology might mean to his ability to command and control is substantially more difficult, unless the user has had meaningful past experience with automated decision aids. The sophistication level of senior commanders in the uses and benefits of ADP technology to enhance their ability to command and control should increase in the future, as today's junior officers, many of whom have been exposed to computers in college or early in their military careers, and some of whom are buying and using personal computers, move to senior ranks.

C^2 system requirements change with changes in: threat to forces commanded (and to the C^2 system itself), geography of the theater or type of forces commanded, doctrine, rules of engagement, scenario, battle situation, status of systems being controlled, and especially as commanders and/or their terms of reference change.

But continuous service must be provided: Like the Bell system, C^2 system operation cannot be interrupted as upgrades are introduced.

Finally, the man-machine interfaces are more interactive and complex.

2. Provider

The "provider" is a term the Team coined to mean the combination of the developer or acquirer (including contractor(s)) plus the supporter/logistician/trainer, and the software programming team which supports the system during its operational use. The purpose of this grouping is to avoid the classic separation between developer and supporter (such as between AFSC & AFLC) that exists in DoD system acquisition, because, as developed later, a C^2 system acquisition program should not attempt a total one-time system definition and execution, as does a traditional acquisition.

3. User

The real users of C^2 systems are the commanders (and their staffs) who have a wartime mission and are responsible for the outcome of their missions (i.e., a user who "faces" a potential enemy daily--e.g., theater forces in Europe or Korea, the RDJTF, deployed fleet forces, SAC wings, NORAD). All others are surrogate users; that is, representatives of the real user(s).

We believe participation of the real user is crucial for at least three reasons: (1) the real user has a wartime mission and is responsible for failure, (2) the real user and his staff do the mission "thinking," and (3) the real user automatically considers the multi-Service and multi-national imperatives.

The Study Team recognizes that the degree of real user sophistication and ability to understand the impact/utility of ADP (automated data processing) on his ability to command and control, varies as a function of whether the user has had experience with or presently has automated decision aids.

Additionally, as developed later, the Study Team believes that the definition and development of systems that are going to be deployed worldwide--for example, Tactical Flag Command Center, planned to be deployed aboard fleet capital ships; and systems like TACFIRE, TSQ-73, and

SIGMA/maneuver control system (so-called "tactical" systems)--require the active participation of a "user surrogate" in system definition/development, coupled with a representative real (i.e., "lead") user. This "lead user"/real user team is required because all the users cannot be involved in the development.

It is further recognized that only at Hq-level occurs the requirement to reconcile: (1) total stated needs and (2) total resource constraints. This cannot occur at field commands. Thus, the Hq-level user surrogate has a role in C² system acquisition, but not a dominant role. (See Chapter IV). As developed later, the Study Team found that for higher-echelon C² systems--in the case of the Army, corps and above; in the case of the Air Force, wing and above or certainly at the air force level; and especially at the CINC level at the unified and specified commands--the real user has not been well represented by surrogates. The problem with leaving requirements definition to the Hq-level user surrogate is that a single Service does not objectively serve the real user well, when the real user has a joint or multi-national wartime mission--and most do. Therefore, the Study Team strongly believes that a lead (representative) real user should be assigned to work with a user surrogate(s) in the definition & design of these systems. This will insure that the concerns felt by a real user will be blended with a surrogate's longer-term perspective and need to satisfy resource constraints. The user surrogate(s) also acts to insure that the views of other real users are factored into the process.

4. Evolutionary Acquisition

a. Basic Definition

Evolutionary acquisition is a concept substantially different from the traditional method of system acquisition. The Study Team found that no generally-accepted definition of evolutionary acquisition exists. The definition of evolutionary acquisition derived by the Team is shown below. It is a carefully worded definition, derived by the Study Team over many hours of discussion.

> "Evolutionary acquisition is a system acquisition strategy in which only a basic or "core" capability is acquired initially and fielded quickly, based on a short need statement that includes a representative description of the overall capability needed and the architectural framework within which evolution will occur. Subsequent increments or "blocks" are defined sequentially, based on continuing feedback provided from lessons learned in operational usage, concurrent evaluation of adequacy of hardware/software configuration, and judgments of improvements or increased capabilities that can result from application of new technology, where feasible."

This definition bears examination. First of all, it talks about "basic or core" capability. How does one define "core" capability? The "core" is the minimum capability, that, if deployed, would provide an operational user with a measurable incremental increase in his capabilities to perform his wartime mission. Thus, definition of the "core" must involve the user.

Who is in the best position to determine the value of this incremental increase? In the Team's view, it is the operational user himself. Or, for systems that have multi-user, world-wide deployment requirements, a user surrogate, <u>coupled</u> with a representative real (i.e., "lead") user.

The definition also shows that EA is based on "a short need statement that includes a representative description of the overall capability needed and the architectural framework within which evolution will occur." This is in marked contrast to the traditional requirements process, because it involves a shortened initial "requirements process," and the system description is qualitative, merely providing an overall framework of desired functional characteristics within which to get the program <u>started</u>. Such description can be initiated by the user or by a joint user/provider effort and should focus at a functional or capability level, not a hardware/software level.

The key concept here is that, under EA, the requirements process cannot be implemented in paper alone, but requires feedback from user testing of the "core" capability (and subsequent increments), as an integral part of (NOT separate from) an evolving "requirements process," when applied to C^2 systems. Neither the operational requirement nor measures of operational worth can be specified adequately in advance. They, themselves, must evolve as well.

The "requirements process" never ends for C^2 systems! This is not a "minus," as one might conclude, if thinking in terms of the traditional method of acquisition. This is a "plus," since the evolutionary approach limits risk and keeps from overextending, which ample past evidence (TOS/OS, original TFCC, TACC AUTO) shows will result in failure.

Another concept fundamental to evolutionary acquisition is to keep the increments of effort relatively small. ("Build a little, test a little.") This has the added advantage of minimizing overall exposure and risk. Also, since much of the evolution is in software, this approach is particularly valuable for C^2 systems, as contrasted with more hardware-rich systems.

Finally, just as there are many alternative means of implementing (or tailoring) a "traditional" acquisition strategy, there is no one single strategy for implementing evolutionary acquisition. This, too, should be tailored to the C^2 system being acquired and to the architecture within which it will evolve.

b. Considerations Underlying the Definition

Some discussion of how the definition of EA was derived is in order. Section 13 of the March 1980 version of 5000.2 (Appendix D) stresses the so-called "evolutionary approach" to the acquisition of "Command and Control Systems." One of the major descriptors of such systems that it lists is that "their operational characteristics are largely determined by the users in an evolutionary process." (emphasis added) More significant here, perhaps, the March 1980 version of 5000.2 mandates that "the design and testing of such systems should, in most cases, be accomplished in an evolutionary manner." (emphasis added) In view of this stress, the AFCEA Study Team in reviewing the acquisition of C^2 systems with the objective of improving the approaches being taken to such acquisition, was asked to give particular attention to the promise and problems of the evolutionary approach (discussed in Chapter III).

There arose within the Study Team, the question of what the EA approach is, in contrast to the more "normal," or presumably "revolutionary," approach that might be taken to the acquisition of C^2 systems. Trying to answer this question in a strict fashion at the beginning of the study was determined by the Team not to be a useful activity, because of the range of disagreement about the topic. On the one extreme, some participants expressed the belief that EA has a precise meaning; on the other extreme, others evidently believed that most DoD materiel is acquired in an evolutionary manner in one way or another, in the sense of regularly being upgraded over time. They felt that truly "revolutionary" jumps are a rarity, in fact. And so one of the important purposes of the case studies and command visits by the Study Team was to learn what people meant, with what results, when they said they were using the evolutionary approach to C^2 acquisition.

Part of the reason for the disagreement over what "evolutionary" means was found to lie in the two different meanings of the term "revolutionary" that are simultaneously in current use. One meaning is the "new start," in which an existing capability to do some military job essentially is entirely replaced with something else in a single, specified effort (e.g., F-15 fighter aircraft to replace the F-4). The other meaning of "revolutionary" relates to a far-reaching degree of technological advance being undertaken in a single effort (e.g., NASA Project APOLLO). Neither meaning is completely satisfactory for defining EA by contrast.

By the end of the study, it was agreed that an acquisition approach would be considered evolutionary for purposes of the study only if it had at least the three attributes shown in Figure I-3, in contrast to those of the more "traditional" (or "normal") approach.

Evolutionary	"Traditional"
• The requirement which gives rise to it is quite difficult, if not impossible, to specify, and must, therefore, itself evolve.	• The requirement is sufficiently specifiable at the beginning of a program to allow for trying to satisfy it all at one time, even if the acquisition proceeds on an incremental basis for other reasons.
• It is a "design-and-tryout," or adaptive approach, in which succeeding blocks of work after the first cannot adequately be specified until feedback from some user is received on the usefulness and needed modifications to prior blocks.	• Real users do not have to be heavily involved in the development effort once their requirements are officially approved.
• Only a real user can determine the ultimate worth of what is being acquired, because he cannot sufficiently specify measures of worth in advance for others to perform this task.	• Operational utility evaluation can be accomplished largely by others, such as independent test organizations.

Figure I-3. Key EA Attributes vs Traditional Approach

EA also has the flavor of the biological meaning of evolution, since it involves "survival of the fittest." That is, under EA, only those increments survive (and in a form) which satisfy a real need on the part of some real user.

In sum, while "evolutionary" acquisition appeared at first glance merely to be the opposite of a "revolutionary" acquisition, it turned out to be much more than that. It is, in fact, a relatively new concept of heuristic (or "learn-as-you-go") design, on the part of both user and developer, which of necessity is applicable to C^2 systems because of their very nature. In theory, however, EA is applicable to any DoD system acquisitions which have the above three attributes.

c. Steps in Evolutionary Acquisition

While there is no one single strategy for implementing EA, the scenario presented below was developed by the Team to: (1) illustrate the key elements and attributes of EA, and (2) given a situation in which the requirements are difficult to ascertain and properly articulate (see 1.D.1.b Unique Characteristics of C^2 Systems), to serve as the Team's recommended design and acquisition model.

One of the most difficult aspects of C^2 systems is the development of a set of system requirements that truly reflects the needs and desires of the user, as well as being understood by the provider. As developed later in this report, the Team has concluded that a tool should be provided to serve as a catalyst to: (1) enhance the understanding of technology and its implications on the part of the users; (2) enhance the understanding of operational problems and needs on the part of the Provider, and (3) facilitate communications between them.

As described in Chapter V, the Study Team recommends a Rapid Requirements Definition Capability (RRDC)[1/] which serves to provide a tangible embodiment of design concepts to users for investigation and testing.

Having iterated through alternative system concepts and assessed their potential operational impact, technical risk, and feasibility utilizing the RRDC, a system concept, architecture and design for a "core" should emerge.

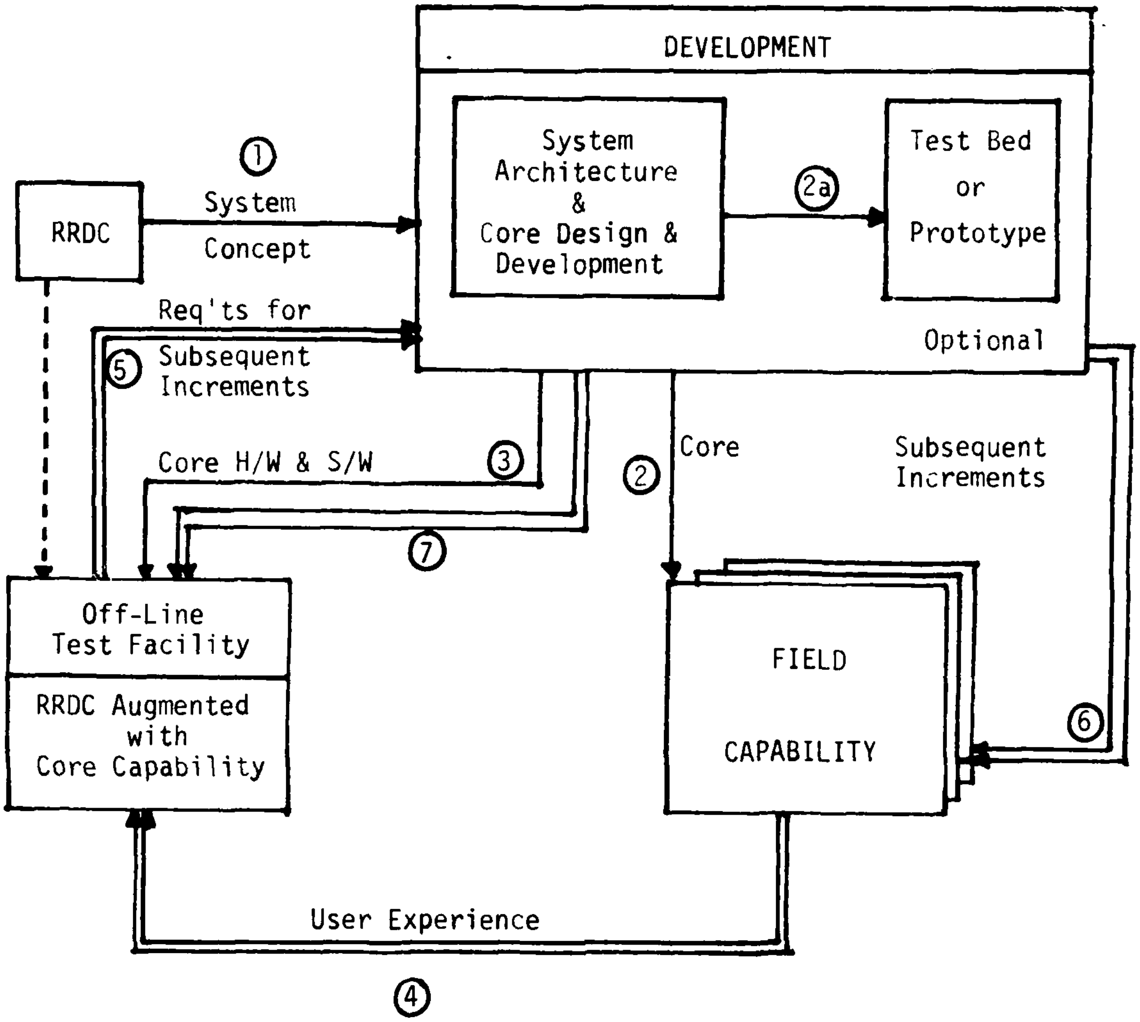
As explained above, this "core" is more than just an initial set of capabilities rushed to the field. It embodies the architectural attributes of the system as it will evolve, and has been implemented in a way to facilitate this evolutionary change.

After the "core" system has been defined, there may or may not be a test bed or prototype fielded for further design/development testing before actual building of the "core." This will depend, in part, upon whether this is a one-of-a-kind system.

The development and fielding of the "core" sets the stage for system evolution. User experience with the "core" (plus experimentation with doctrine, tactics, and corresponding system utilization) would serve to be a major source of feedback to defining the capabilities to be incorporated in subsequent increments.

As also described in Chapter V, the Team envisions the RRDC being augmented with the actual capabilities of the "core" to provide an off-line capability for actual testing of proposed hardware or software changes to the "core." This seven-step process is diagramatically depicted in the figure below.

[1/] The RRDC should not be confused with a test bed or prototype--these later facilities are normally thought of as tentative or trial implementations of a system concept or design, while an RRDC is intended as a tool to help develop a system concept.



1.	Use of RRDC to Develop System Concept & Core Requirements	4.	User Feedback suggests possible changes to be explored in RRDC
2.	Development of Core	5.	Design for Subsequent Increments Developed
2a.	Core-Based Test Bed (Optional)	6.	Subsequent Increments Developed
3.	Augmentation of RRDC to serve as Off-line Test Facility	7.	Off-line Facility Updated

Figure I-4. Steps in Evolutionary Acquisition

d. Evolutionary Acquisition Compared With P^3I

The Study Team found much confusion in various policy and field groups visited about the difference between "Evolutionary Acquisition" (EA) and "Pre-Planned Product Improvement" (P^3I). Most either considered them to be quite similar, if not identical, or EA merely to be a sub-set of P^3I. The Study Team found a number of similarities and differences between the two approaches. The similarities are as follows:

(1) Both are incremental approaches, where it is planned to implement regular upgrades from the beginning of a program.

(2) In both cases, initial and subsequent design efforts are deliberately approached in such a way that the planned upgradings can be accomplished more easily, i.e., design is focused initially as much on changeability as on system optimization. In the case of C^2 systems, this might be done by providing extra through-put capacity and/or memory and taking a modular approach to system design.

(3) Both ordinarily involve initially striving for something less than either the system or technological states-of-the-art would permit, particularly something less than the most far-reaching states, or "revolutionary" leaps, would permit.

In view of these similarities, one might ask what differences there are between the "evolutionary approach" and P^3I." In answer, several possible differences might be noted where C^2 systems are concerned:

(1) The evolutionary approach usually is adopted as a strategy because it has to be, i.e., because: (a) it is so difficult to state requirements adequately at the beginning of a true C^2 program, (b) such requirements are expected to change frequently over the life of the program, or (c) users cannot specify accept-

ability criteria adequately in advance due to the subjective nature of these criteria. This leads to a "design-and-tryout" approach having to be taken both to defining the need and to the approach to satisfying the need. In contrast, the P^3I strategy may be adopted for any one of a number of reasons, even when it doesn't have to be--even, for example, when a requirement can be stated adequately and its achievement can be measured objectively.

(2) An overall program to which the evolutionary approach is being taken may involve little to no advanced development (6.2/6.3A) of any type: for example, when the user upgrades his C^2 capability through using existing commercial or military materiel to build a "prototype" of some type. In fact, this is the mode preferred by policy (Section 13 of DoDI 5000.2 of March 1980; Appendix D hereto). In contrast, a P^3I program ordinarily does involve advanced forms of development-- significant amounts of such development, in fact. Indeed, P^3I is a strategy that has come to the fore in recent times as a means of dealing with just such uncertain advances, because, among other principal reasons, the development period involved in taking a very large or "revolutionary" jump towards the limits of art, each time a new program starts, has been taking so long and been so risky, that U.S. readiness is being threatened.

(3) While it is highly desirable that users be constantly knowledgeable about P^3I programs--indeed, play a continuous, if reactive role in the acquisition of <u>any</u> DoD system--the P^3I approach <u>per se</u> does not require that the user accept any significant responsibility at any stage of the acquisition cycle. In contrast, strong real user/lead user participation in and influence over the acquisition is a major aspect of the EA of C^2 systems, as previously indicated. EA requires a larger role and heavier continuing involvement of the user in terms of:

- Planning and design initiative (e.g., CAFMS).

- Relative responsibility for program results.

- Management control of the program as it progresses (e.g., determination of operational utility).

In fact, the fundamental need for continuing, close interaction among all the participants in the C^2 system acquisition process--especially the provider, user and independent tester--is basic to EA, whereas it is not basic under P^3I.

(4) Finally, EA differs from P^3I in several other respects:

- EA demands an accelerated and abbreviated "requirements process" and "procurement process" leading to contractor selection. This is necessary to enable rapid fielding of a "core" and subsequent increments so that evolution can occur based on feedback from test in the user's environment.

- Different PPBS/budgeting approach arising from less initial detail on the ultimate total program.

- Differences in Program Office staffing. A traditional acquisition is "front-end heavy" in engineers and analysts and "back-end heavy" in specialists in producibility, testing and ILS. Under EA, there is essentially a continuous need for all these skills but in a more "level-of-effort" fashion.

In sum, while the two approaches are incremental and have a number of similarities in form, they differ significantly in front-end specification and implementation. They are distinctly different concepts.

E. INFORMATION GATHERING & RESEARCH

1. Visits & Interviews

Figure I-5 lists the various organizations with which the Team interacted in this study. It did not meet with members of the Congress or their staffs because it believed that DoD and industry issues and viewpoints should be resolved first.

HQ	USER & USER SURROGATES	PROVIDERS	TESTERS
OSD/ACQ POLICY	TRADOC	CECOM	OSD, ODT&E
OSD/C^3I	OPNAV	NAVELEX	OTEA
OJCS/C^3S	CINCLANT	ESD	OPTEVFOR
DCA	II MAF	HQ AFSC	AFTEC
HQ, DA	TAC/TAFIG	JTFPMO	
OPNAV	ADCOM	DIR, DCA	
HQ, USAF	USAFE		
HQ, USMC			

Figure I-5. Visits & Interviews

a. Headquarters

The Study Team began its information gathering effort with various Hq people in acquisition policy, as well as command & control positions in OSD and the MIL DEPs, Hq DCA, and OJCS/C^3S. Representatives also met personally with LTG Dickinson (Director of OJCS/C^3S) and LTG Hilsman (Director, DCA) during the study. In general, Hq people not regularly associated with C^2 systems acquisition were skeptical of the assertion that "C^2 systems were different and so should be acquired differently."

b. Providers

The Team also made a number of field trips. The first field trip was made to CECOM, NAVELEX and ESD (meeting with both Navy and USMC C^2 system project offices at NAVELEX) to interview the functional and project organizations that are involved with C^2 system acquisition, as well as the various project offices related to the case studies that the Team was reviewing, plus some other project offices that were suggested to it.

Because of his strong personal interest in the topic, representatives of the Team had a special session with General Marsh (Cmdr, AFSC) and his top staff and also met with representatives of the Joint Tactical Fusion Project Management Office, which is the program taken as the Team's joint case.

While it is difficult to generalize interactions with "providers," in the main the Team sensed a belief in and a desire by C^2 system project offices to implement the evolutionary approach, but to do so, they all were being forced to go to extraordinary lengths to "negotiate a truce" with "functional" organizations (e.g., requirements writers/validators (usually at Hq), budgeting (PPBS) (again, usually at Hq), contracting, "ilities" and legal people, independent testers, supporters), who, in general, were reluctant to permit any deviation from the "traditional" approach, even though OMB Circular A-109 & DoD Directive 5000.1 encourage "tailored" acquisition strategies.

The "closer" (organizationally) the Study Team got to Service Hq, OSD and Congress, the more concern was expressed about the ability to get those responsible for budget (PPBS) approval to accept the need for budget approval without a detailed "Requirement" supported by exhaustive cost/effectiveness analysis.

c. Users and User Surrogates

The Team also visited a number of users and user surrogates. In the latter regard, the general view at TRADOC, for example, was that they well represented Army "tactical-level" users, but they did accept the notion of working with a "lead user" in multi-user programs and recognized that they did not well represent users at echelons above corps.

Members of the Team also met with VADM Gordon Nagler (OP-094), the Director of Navy Command and Control, who generally is accepted as the "user" for all C^3 systems within the Navy, even though technically a user surrogate. There is a view, within parts of the Navy, that, within the Navy, only OPNAV can define C^2 system requirements. While the Study Team recognizes the role of the Hq-level user surrogate in adjudicating competing "needs" with available resources, the Study Team's view, as supported by ample case evidence, is that continuous real (or lead) user involvement is a (perhaps the) key factor in successful C^2 (decision support) system definition and implementation.

One significant "real" Navy user with whom members of the Team interacted was CINCLANT, ADM Train, and the CINCLANT staff. (He also is SACLANT, CINCLANTFLT and COMWESTLANT). ADM Train's attitude was that ADM Nagler was the best qualified to define Navy command & control system requirements. ADM Train felt that since Adm Nagler had been a Battle Group Commander himself, he understood the problem of the real afloat user. In ADM Train's opinion, the Fleet and Battle Group Commanders themselves did not need to be involved, day-to-day, with the acquisition of Navy C^2 systems--ADM Nagler and his staff were capable of defining Navy C^2 system requirements, and OPTEVFOR (Navy Operational Test & Evaluation Force) was capable of testing and evaluating Navy C^2 systems and determining their readiness for fleet use. Therefore, ADM Train did not see a significant need for fleet resources to be employed in the evaluation and test of C^2 systems, beyond those already employed in support of OPTEVFOR activities. (This appeared, to the Study Team, to be more a CINCLANTFLT view than a CINCLANT or especially, SACLANT view).

In this regard, in an earlier meeting, RADM Blount, the Commander of the Navy Operational Test and Evaluation Force, indicated that he drew quite heavily, as did the other independent testers, on user assets--fleet people, in the case of the Navy--to perform operational tests, and therefore got strong "user" inputs as a result of what he felt was adequate user participation in the testing by those means. (In the Study Team's view, this reflects a tactical C^2 viewpoint and not a CINC (joint and/or multi-national) viewpoint and is the essence of the gut issue--whether the Services can adequately develop C^2 systems to serve joint and/or multi-national users). The Study Team recognizes that shipboard systems are not adapted as easily as shore-based systems. Equipment changes usually require schedules to be meshed with yard ship-alteration or outfitting times, which occur several years apart. These limitations also apply, but to a lesser degree, to airborne systems. Also, the requirement to standardize Service-wide (or DoD-wide) inhibits responsive change to factors such as those discussed in Section D.1.b (p I-11-12), but may be necessary because of programmatic considerations.

Members of the Team met with Hq USMC and FMFLANT/II MAF representatives and found a favorable response to the notion that C^2 system acquisition should be evolutionary.

The Team went to Hq Tactical Air Command and met with members of the TAC staff, as well as the TAFIG (Tactical Air Force Interoperability Group). Representatives of the Team met with Aerospace Defense Command people, regarding the 427M/Cheyenne Mountain Complex case that they were analyzing, as well as obtaining views regarding the efficacy of the combined program office for the 427M/Cheyenne Mountain Complex (CMC) Upgrade. Some of the Team also met with representatives of Hq USAFE. In general, the Air Force appears to have thought about evolutionary acquisition of C^2 systems a great deal and has taken the initiative to provide it on several programs, even to the extent of user-led programs, such as CAFMS (Computer-Aided Force Management System), and combined user/provider programs, such as the present CMC Upgrade.

d. Independent Testers

Some of the Team also met with representatives of the OSD Director of Test & Evaluation and the Army, Navy and Air Force independent testers. The Team did not meet with the Marine independent tester, mainly due to time availability.

Representatives of the Team met with Charles Watt and Don Wood from the Office of the Director of Defense Test and Evaluation in the Office of the Secretary of Defense. Mr. Watt, the Deputy Director for Strategic, Naval and C^3 System Test, is sympathetic to the notion of heavy user involvement, both in the acquisition and the test of C^2 systems, but appears reluctant to reduce the "arms length" relationship between provider and tester. Don Wood, the cognizant staff person for tactical C^2 systems tests, expressed frustration at the inability of users to define requirements specifically enough so that operational effectiveness test criteria could be defined.

Representatives of the Study Team met with the Commander of the Army Operational Test & Evaluation Agency (OTEA), MG Kirwan, and Dr. Dickenson and other members of General Kirwan's staff, including the OTEA Project Manager for SIGMA/Maneuver Control System (MCS) testing. There is a good appreciation at OTEA of the complexity involved in testing C^2 systems, and a recognition that C^2 systems are indeed different from other systems, coupled with a degree of frustration over vagueness of "requirements" for C^2 systems. It is very difficult for the independent tester to define objective measures of operational effectiveness against which to perform independent test & evaluation (IT&E) of C^2 systems.

The Team had an excellent meeting with RADM Blount, Commander Operational Test & Evaluation Force and his staff at Norfolk. The thrust of his comments is related earlier.

Mr. O'Donohue, the Study Chairman, met with MG Whitlatch, Commander of AFTEC, and also had several phone conversations with members of his command & control systems test staff.

Some members of the Study Team went into this study with a point of view that the independent tester was an obstacle to progress, from the standpoint of timely fielding of new systems. Meetings with senior independent testers modified these views. The Team found increasing awareness, within parts of the IT&E community, of the complexity and difficulty involved with C^2 system IT&E--especially with respect to operational effectiveness determination--coupled with a willingness to allow the user to play a more significant role in the evaluation of C^2 systems than normally has been allowed. The Team also sensed an increasing appreciation, on the part of senior independent testers, of the need for closer coupling between testers, users and providers in testing C^2 systems, and using these tests as part of the evolving requirements process. However, the "testers" were concerned that higher authorities (e.g., ASARCs, DSARCs) are accustomed to receiving test data in a more precise and quantitative manner than appears feasible for C^2 system operational utility determination. As discussed later, more progress needs to be made in making the traditional approach to T&E more responsive to the unique nature of C^2 systems.

Colonel Alan Salisbury, the SIGMA/MCS Program Manager, whom the Study Team visited during its trip to CECOM, briefed the approach he had taken to acquisition of the SIGMA/MCS program to date. He expended a great deal of effort in "negotiating" arrangements with the IT&E community and other members of the materiel acquisition community to provide a different approach to the SIGMA acquisition. The first portion of SIGMA, the so-called TCS/TCTs, have been deployed--17 or so of them--to VII Corps in Europe, and currently are in use at various VII Corps echelons and commands. The VII Corps is the so-called "lead corps" in the development of SIGMA, working with TRADOC as the user surrogate for all the other potential using corps and commands of the SIGMA/MCS. COL Salisbury also has worked out an arrangement with the IT&E community, where they take on a role more of providing resources to support the test and evaluation and "observe" on a "non-interference" basis, the T&E in essence being performed by the VII Corps itself. This approach (user-led T&E) has resulted in some frustration, in the IT&E community, concerning their ability to provide meaningful IT&E data to higher-level decision-makers.

Many of the real users with whom the Study Team met expressed concern about being "resource poor," especially in technically-sophisticated people. Real users say they have all they can do to perform their day-to-day operational mission. They do not have any resources left over to enable them to participate to the degree that the Study Team would advocate in the acquisition and evaluation of C^2 systems. It should be noted that C^2 systems of the type covered in this study comprise only a small part of the totality of DoD systems. The Study Team makes some recommendations later in this report about providing resources to the user to facilitate his participation in EA, but the unsophisticated user needs help from the provider and the tester to be efficient in the way he goes about acquiring and evaluating C^2 systems. The tester specifically can help the user avoid being wasteful in the way he goes about the T&E. He can help design the test and data acquisition/analysis system, participate in specifying the test measures of effectiveness, etc. As was said earlier, the increasingly sympathetic nature of the independent tester to this kind of approach was most heartening.

2. Case Analysis

Figure I-6 lists the basic DoD programs selected for case study, all of which, with one possible exception, fit the Team's definition of C^2 systems. As stated earlier, the Team selected a cross-section of C^2 system programs from each military department, as well as a joint program. Some of these programs were at least partially evolutionary, some were not; some were big and some were small, both in dollars and in quantity of systems acquired. The Team tried to select cases from a spectrum of "failures" to "successes," and to review systems that were used at various echelons of users and types of users. The programs selected were taken both from a list that the Team compiled and from suggestions it got from OUSDR&E/C^3I and the military departments.

JOINT	ARMY	DEPT OF NAVY	AIR FORCE
BETA JTFP ASAS ENSCE	TOS/SIGMA TACFIRE PLRS/JTIDS HYBRID	OUTLAW SHARK TFCC MIFASS	OASIS 427M/CMC TACC AUTO CAFMS CONST WATCH EIFEL

Figure I-6. Case Studies

The joint program we reviewed was the Army/Air Force/DARPA BETA (Battlefield Exploitation & Target Acquisition) testbed and its current nomenclature, the Joint Tactical Fusion Program (JTFP). The Army Initial All-Source Analysis System (ASAS) and the Air Force Enemy Situation Correlation Element (ENSCE) are planned to derive from the JTFP. An evolutionary approach is planned for these JTFP systems.

The Army programs reviewed included the cancelled Tactical Operations System/Operable Segment (and earlier versions, including TOS/Europe), plus the SIGMA/Maneuver Control System, a program planned to be evolutionary, that addresses part of the TOS mission need; TACFIRE, a traditional C^2 system development (that started out as a Total Package Procurement) to provide automated artillery fire direction control; and the PLRS/JTIDS Hybrid (Position Location Reporting System/Joint Tactical Information Distribution System), a system that only partially meets the Study Team's definition of C^2 (PLRS provides FRIENDSIT). All of these are multi-user systems.

Department of the Navy C² system programs studied included the OUTLAW SHARK/TFCC (Tactical Flag Command Center) program, an over-the-horizon targeting system; and the Marine Corps MIFASS (Marine Integrated Fire & Aerial Support System), the latter a major traditional C² system development that was preceded by a significant testbed effort to define requirements.

Air Force programs studied included OASIS (Operational Application of Special Intelligence Systems), a one-of-a-kind system for USAFE (U.S. Air Forces Europe); original 427M and the later Cheyenne Mountain Complex Upgrade, an evolutionary upgrade of CINCNORAD's C² system, now a combined user/provider program; and several attempts to automate all or portions of the Tactical Air Control Center: (TACC AUTO), a "total" attempt based on a "total" requirement; CAFMS (Computer-Aided Force Management System) a user-led acquisition of a reduced version of TACC AUTO, using parts of the TACC AUTO software design, for application by the 9th and 12th (CONUS) Tactical Air Forces; EIFEL, an adaptation of the Luftwaffe's automation of FRG-operated ATOCs (Allied Tactical Operations Center) to ATOC Sembach, a U.S.-led ATOC for NATO RSI (Rationalization/Standardization/ Interoperability); and the TACC automation part of Constant Watch (the PACAF-led program to upgrade the Korean Air Intelligence System).

Figure I-7 lists the major categories of information gathered in doing the case studies, the main purpose being to derive "lessons learned."

- GENERAL DATA
- REQUIREMENTS DEFINITION
- ACQUISITION STRATEGY
- PROGRAM REVIEW MECHANISMS
- TESTING APPROACH
- TECHNOLOGY EMPLOYED
- CONTRACTOR/GOVERNMENT PERFORMANCE
- LESSONS LEARNED

Figure I-7. Case Data Collection

As a separate effort involving cases, the Study Team reviewed literature that summarized conclusions drawn from experience on over 20 decision-support systems developed for commercial use--the business/industry "equivalent" of a C^2 system.

The analysis and conclusions drawn from these case studies are discussed in subsequent chapters.

CHAPTER II

MAJOR CONCLUSIONS AND RECOMMENDATIONS

CHAPTER II

MAJOR CONCLUSIONS & RECOMMENDATIONS

A. SOURCES OF CONCLUSIONS & RECOMMENDATIONS

A significant source of data was the Study Team's own experience. Many of its members had had experience with one or more of the C^2 systems selected for case study, plus other similar C^2 systems. All had prior/ current experience in C^2 system acquisition, some both with government and industry.

Second, through the Team's own resources and those of organizations and individuals with whom it met, the Team was able to compile a collection of pertinent reports, memos, briefings, etc.

Third, the Team met with over 200 individuals in the organizations of the various participants ("players") in the C^2 system acquisition process--Hq staffs, users/user surrogates, providers and independent testers.

Finally, the Study Team compiled and analyzed a stratified sample of case histories of pertinent C^2 system developments, in order to derive "lessons learned" from past and current experience.

The combination of these data and experiences were analyzed and synthesized to form the basis of the Study Team's conclusions and recommendations.

B. MAJOR CONCLUSIONS

Figure II-1 outlines the five major conclusions of the Study Team:

- EVOLUTIONARY ACQUISITION GIVES A MUCH HIGHER PROBABILITY THAT A USEFUL MILITARY CAPABILITY WILL BE FIELDED EARLIER
- ALTHOUGH EVOLUTIONARY ACQUISITION IS POLICY FOR C² SYSTEMS, ITS APPLICATION IS SPOTTY AND IT IS NOT WELL DEFINED OR UNDERSTOOD
- EVOLUTIONARY ACQUISITION WILL NOT WORK ON A "BUSINESS-AS-USUAL" BASIS, YET ACQUISITION SUPPORT COMMUNITIES (E.G., REQTS VALIDATION, BUDGETING, CONTRACTS, "ILITIES," TEST) DISCOURAGE APPROACHES DEVIANT FROM THE TRADITIONAL APPROACH
- SUCCESSFUL EVOLUTIONARY ACQUISITION REQUIRES CONTINUOUS INTERACTION AMONG USERS, PROVIDERS & TESTERS AND A MORE INFLUENTIAL ROLE BY THE REAL USER
- A POTENTIAL FOR CHAOS EXISTS IF C² SYSTEM ACQUISITION PROCEEDS WITHOUT AN ARCHITECTURAL FRAMEWORK, INCLUDING FLEXIBILITY TO FACILITATE GROWTH

Figure II-1. Major Conclusions

First, the Study Team concluded that evolutionary acquisition (EA) gives a much higher probability that a useful increment in military capability will be fielded sooner and more often. In addition to data gathered from its literature search and its extensive meetings with experienced C² personnel in various DoD organizations, the Study Team found strong supporting evidence from its case analysis. Programs such as CAFMS, OASIS, TFCC/Outlaw Shark, and the current Cheyenne Mountain Complex Upgrade, which were incremental in nature and have a higher measure of continuing user involvement throughout, were found to be more "successful" than programs that followed the traditional approach, such as TOS/Operable Segment, Original TFCC, TACC AUTO and original 427M--all failures. The Study Team found similar results in the literature on similar commercial systems. (See Chapter III.)

Second, although evolutionary acquisition has been official policy for C^2 systems since early 1980, and was encouraged by OSD for several years before that, the Study Team found application of EA to date has been spotty. More importantly, the concept is not well defined or understood, and this misunderstanding exists among all of the participants in the process, from Program Managers to Hq staffs, acquisition support people, testers, users, user surrogates, logisticians. The March 1980 DoDI 5000.2 C^2 system acquisition policy is being interpreted to "allow" rather than require EA, and is not supported by understandable application guidelines. (See Chapter III.)

Third, the Study Team concluded that evolutionary acquisition will not work on a "business-as-usual" basis. With few exceptions, C^2 system acquisition generally is being practiced on a "business-as-usual" basis (i.e., traditional, serial weapon system-oriented acquisition methods)--especially in the requirements/program approval process, in budgeting and contracting, and in T&E. In the absence of clear direction to the contrary, this "business-as-usual" bureaucratic inertia will not easily be overcome and is forcing both program advocates and program managers to take extraordinary actions to enable them to pursue evolutionary acquisition, even though it is mandated in DoDI 5000.2. (See Chapter III.)

Fourth, the Study Team concluded that successful implementation of evolutionary acquisition requires continuous interaction among users, providers and testers. The present relatively serial and "arms-length" relationship among the real user, the provider, and the independent tester inhibits the effective use of evolutionary acquisition. Even though the policy was pronounced over two years ago, the classic relationship still remains. The provider is dominant in development. The independent tester dominates the test with few program exceptions. (With some exceptions) there is little, if any, continuous participation by real users. This must be changed to a situation where the real user (or combination of lead user and user surrogate for multi-user systems) is dominant in the acquisition process of C^2 systems, with significant support from providers and independent testers. A "combined" program office, comprised of users, providers and independent testers, is one promising approach. (See Chapter IV.)

The most significant problem is insufficient continuing real user participation and influence throughout the acquisition process. In most programs the Study Team reviewed that encountered problems, there was little real user participation and influence, especially in the initial definition of "core" capability and the feedback of user test data to the provider in near-real time. The Team found a general attitude, especially in user surrogate organizations, that quarterly or periodic (e.g., at SDR, PDR, CDR) user or user surrogate participation is adequate to enable the acquisition of C^2 systems. It is the unanimous view of the Study Team that the real user (or lead user plus surrogate) <u>must</u> be involved continuously with the process of evolving the system, its requirements, its design, its testing, resource allocation, etc. (See Chapter IV.)

Finally, and perhaps most significantly, the Study Team concludes that there is a <u>potential for chaos</u> if C^2 system acquisition is allowed to proceed without a carefully-conceived and structured architectural framework that provides flexibility to facilitate orderly change and incremental growth without adverse affects on reliability, performance and cost. Without such a well-conceived architectural framework, designed to accommodate change, the user could be left with an initial "core" that rapidly could become obsolete.

The Study Team, therefore, recommends the development of a framework which encompasses two types of architectures. The first addresses the operational theater or mission-related military functions and tasks (e.g., detection, fusion, allocation) which a commander and his staff use to discharge their responsibilities. These functions and tasks are performed and supported by a number of systems, including a C^2 system(s). The collection of systems, or "system of systems" and organizational entities (too often treated in isolation) must be addressed architecturally as a totality, with particular attention paid to inter-system interfaces. The need for C^2 system interoperability with co-deployed systems varies with the theater operation plan and special mission assignments. Therefore, measures are needed to insure that individual users or lead users are kept fully aware of potential applications of other users of the system. Strong user/provider interaction is required here.

Corresponding to the operational "system of systems" is a hardware/software infrastructure which also requires the interconnection and interoperability of a number of systems. This second type of architecture addresses the design and implementation of specific C^2 system hardware and software which provide system functions and capabilities (e.g., data base management, networking, display). The individual C^2 system designs must support the theater/mission interconnections needed to perform the military functions and tasks.

A layered, open system interconnect model to enable establishment of interconnect and protocol standards is most critical to successful implementation of this second architecture. The compatibility of the C^2 system interfaces can only be assured by developing a structure for interface standardization. At present, the ISO (International Standards Organization) open system interconnect model, which has been implemented partially, is the most promising and most widely-accepted approach. The Study Team was advised by John Cittadino of OUSDR&E that DoD has committed to NATO that DoD systems will employ the ISO model in systems requiring NATO interoperability.

These architectural frameworks, which structure their respective "system of systems," must be modularly designed with sufficient built-in flexibility to facilitate growth and allow for the insertion of new technology with minimum negative impact on the existing system.

Chapter V discusses these architectural conclusions in more detail.

C. MAJOR RECOMMENDATIONS

Figure II-2 lists the major recommendations arising from the Study Group's conclusions:

1.	MANDATE & FACILITATE USE OF EVOLUTIONARY ACQUISITION
2.	ALTER ROLES & RELATIONSHIPS AMONG USERS, PROVIDERS AND TESTERS TO ENABLE CONTINUOUS INTERACTION
3.	USE IMPROVED SYSTEM ARCHITECTURES & DEVELOPMENT PRACTICES, DESIGNED TO ACCOMMODATE CHANGE

Figure II-2. Major Recommendations

The first recommendation arises from the first three major conclusions. The second major recommendation arises from the fourth conclusion; and the third recommendation arises from the final major conclusion.

First, evolutionary acquisition must be mandated and facilitated as the primary C^2 system acquisition strategy. Chapter III expands on this in more detail.

Second, one of the most significant actions that must be taken to facilitate evolutionary acquisition is to alter the roles and relationships among users, providers, and testers to enable them to interact continuously, rather than relying on the more "arms length," serial approach used in traditional weapon system acquisition. (See Chapter IV.)

Finally, importantly, for evolutionary acquisition to achieve its full potential as the adaptive design and usage approach that it is, C^2 system acquisition must proceed within an architectural framework and employ development practices that are designed to facilitate growth and the insertion of new technology with minimum redesign and negative impact on the existing system. (See Chapter V.)

Chapters III, IV and V elaborate on these major recommendations and discuss related underlying issues.

CHAPTER III

THE COMMAND AND CONTROL (C^2) SYSTEM ACQUISITION PROCESS AND EVOLUTIONARY ACQUISITION

CHAPTER III
THE COMMAND AND CONTROL (C^2) SYSTEM ACQUISITION PROCESS AND EVOLUTIONARY ACQUISITION

A. INTRODUCTION

This is a report on "command" and "control"--management functions being performed in a military context at all organizational levels. JCS Publication 1 defines command and control as:

> "The exercise of authority and direction by a properly designated commander over assigned forces in the accomplishment of his mission."

Beginning with the Korean War, DoD leadership increasingly has realized that operational military events of all types have entered a stage where they can be expected to take place too fast, to be too complex, and to be too numerous for the human in the loop to deal with unaided. The resulting increased attempts to automate C^2 systems since the Korean War brought into question the appropriateness of the use of normal DoD acquisition methods for acquiring such automated capability. A significant reason for this questioning stems from an increasing realization within DoD that the materiel that provides this automation--data processors, displays, and data links--are not the central ingredient of what is referred to as a "Command and Control (C^2) System," as defined in the first DoD acquisition policy directly on the topic: Section 13 of DoD Instruction 5000.2 dated 3/19/80 (Appendix D). Rather (and unique to C^2 systems among DoD's materiel acquisitions), a human being called a "commander" (with his staff and pertinent doctrine and procedures) is this central ingredient. Providing automated aids to help this commander perform what JCS Publication 1 refers to as "command and control functions" does not change the fact that the commander (and his staff) are the central ingredient of a C^2 system.

Indeed, this official JCS definition of "command and control" makes no mention of automation. It merely goes on from the statement quoted above to conclude its definition of command and control by stating:

> "Command and control functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures which are employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of his mission." (Note: Not merely in making decisions.)

Thus, the driving factor about C^2 systems, among all the systems DoD acquires, is that an operational military commander is not merely the user of the C^2 system. He and/or his staff are the dominant element of it.

Automation provides the means of enlarging a commander's capability to command and control resources in today's complex, high-speed military world. A number of other automated DoD materiel items also are intended to control something, such as the control element of a communications system and embedded control elements of individual weapons, and these other items may even have a number of the major characteristics of "Command and Control Systems" described in DoDI 5000.2's Section 13 (particularly the involvement of much software). But the difference is that the other systems are not involved in human control, but in physical control. In fact, their very purpose is to eliminate the human from the loop as much as possible, not further his role in it.

The Study Team's efforts in reviewing the acquisition process for "Command and Control Systems" thus was focused on those C^3I systems which involve a high degree of man-machine symbiosis, management orientation, and complex interoperability relationships, in contrast to those which are relatively impersonal, are discrete in scope, and involve relatively simple process control. Said differently, the study focused on those C^3I systems for which users should assume a significant responsibility in the acquisition process. Besides playing their traditional roles in determining requirements for the system and in assessing its operational acceptability and worth after development, these users are such an important part of the system that they possess a significant part of the knowledge base needed to design it.

B. PROBLEMS IN DOD'S CURRENT ACQUISITION OF C^2 CAPABILITY

As noted in Chapter I, the Study Team was purposely selected to have diverse backgrounds. Despite these diverse backgrounds, the Study Team came to early agreement that something is very wrong with the way DoD has been approaching the acquisition of its C^2 capability, whatever the DoD Component involved and whatever the intended operational application of the capability. It didn't seem to matter whether an individual Study Team member had been involved in a program to provide unified and specified commanders or the national command authority (NCA) with the integrated capability needed to help them carry out their far-flung responsibilities, or a program to help tactical battlefield commanders to retain control of their forces and weapons in real time, and to exploit available intelligence information in a timely way. All brought to the discussion table a common experience: that ever since DoD first attempted to automate its C^2 functions with the SAGE Continental Air Defense System in the 1950s, automated C^2 capability regularly was costing far more than intended, was entering the inventory far later than expected (if at all), and too often was a disappointment to real users with real needs.

Clearly, the message of the past twenty years of DoD attempts to acquire automated C^2 systems via the traditional DoD acquisition approach (and variations thereof) was that the traditional approach will not work, except under certain distinct program circumstances (to be discussed later). The problems with using the traditional approach to acquire C^2 systems have become worse recently, as laborious, formal, sequential steps have been introduced and enforced--steps aimed more at assuring higher-level control of the process than achieving timely results.

Such a sequential process was deemed necessary because the costliness of many major weapon system programs threatened the consequent "contract" DoD enters into with the Congress regarding the desirability of their acquisition as compared with other things that DoD (and the Nation) could do with the same resources. At the present stage of knowledge of how to apply automation to military materiel, the process by which acceptable results actually are achieved in acquiring C² systems is fragile and heuristic at best. This process, therefore, cannot afford the "excess baggage" imposed by the traditional acquisition approach whether for good reason or not. DoD simply must do <u>something</u> to alter its approach fundamentally to gain such increasingly needed--indeed, high priority--automated aids for accomplishing C².

The basic message this report hopes to establish--not only from the experience of the Study Team, but also from the literature, the numerous and varied visits and interviews conducted, and from the carefully stratified list of cases examined--is that C² systems cannot be acquired successfully via the traditional approach, wherein a detailed total system requirement and resulting total system definition is established "up front," followed by development of the "total" solution.

As will be developed subsequently in this chapter, the Study Team has concluded, from all its data gathering and experience, that the evolutionary approach to acquiring C² systems--which Section 13 of DoD Instruction 5000.2 encourages so strongly "in most cases"--is the most promising approach to such acquisition.

To those who are skeptical about this evolutionary approach, the Study Team stresses that <u>the choice between the traditional approach and the evolutionary approach is not a choice between something acceptable or something better</u>. It is a choice between something unacceptable and something that offers hope, <u>if</u> (and only if) DoD makes the effort to understand EA's ramifications and to adapt it fully. Therefore, to these skeptics, the Study Team poses the question: "What other alternative <u>is</u> there, and what evidence do you have that the alternative can result in successful C² system acquisition?"

1. The Sources of Data on Current Problems

The Study Team collected a voluminous amount of written and oral material to supplement its own knowledge. In a sense, its major contribution through the Study was to provide an experienced and highly interested "set of legs" to help validate the numerous problems in DoD's current acquisition of C^2 capability. It found many senior people in the C^2 community already knew about these problems by the time Section 13 of DoDI 5000.2 was promulgated, and hence realized the need to try something quite different. Some of the more serious problems will be listed in the remainder of this section--problems in the acquisition of C^2 systems in general and problems which are being caused in particular by the application of the traditional approach to the acquisition of such C^2 systems.

What follows in this Section (B) is divided, for discussion purposes, into the basic sources of data explored: the literature, the visits and interviews, and finally the case data gathered. It is presented in this way, in spite of the significant overlap of the findings on problems among all three sources, because each source tended to illuminate certain types of problems more than others. The Section then closes by discussing a problem that DoD has in general with systems acquisition, but which is so magnified in the case of C^2 systems that it warranted separate discussion: the problem of evaluating a C^2 system as it passes through the various stages of its acquisition life cycle.

2. Problems Derived from the Literature

The literature examined was essentially of three types:

- The theoretical or academic literature which underpins the broad information technology field of which C^2 is a part,
- Official and semi-official material (including DoD-funded studies) developed by DoD or its individual members (military and civilian),

- General practitioner literature published in periodicals or individually by such practitioners.

In order not to imply a reliance on any one document for any finding, no sources will be quoted in what follows. Rather, some of the major problem findings derived from literature of all three types will be listed.

a. <u>Uniqueness of C² Systems</u>

The most significant finding from the literature in particular was that C² systems are quite different from other types of DoD systems, and this difference is in kind, not merely in degree along some dimension. That is, C² systems are unique, because the personnel and procedural aspects of a C² system require complete integration of the human element into system design criteria, something not required of any other kind of system.

This finding was brought out especially in the academic literature regarding what are broadly called "decision-support systems (DSS)." But it also appeared in the writings of a number of individual DoD people experienced in C² acquisition, particularly at the Air Force's Electronic Systems Division (ESD), the Army's Training and Doctrine Command (TRADOC), and its Communications and Electronics Command (CECOM). The problem is that this fact of C² system uniqueness is either not well-known in DoD or its full implications are being resisted for a variety of reasons.

b. <u>Procurement and Support Problems</u>

A second major current C² acquisition problem derived in particular from the literature is the difficulty being experienced in trying to support adequately, and procure efficiently (in terms of commonality), systems which are often one or only a few-of-a-kind. These are systems that: (1) have a rapidly evolving technology, (2) have to be tailored to the styles of individual commanders, and (3) above all whose operational essence and costs are dominated by an intangible called "software." And this is not simple software, but software whose very purpose is to emulate cognitive human processes.

Adding to these support and procurement problems is the problem of trying to provide for the interchangeability of trained maintenance and operations personnel that is so vital to a military unit at war. The ability to achieve this needed interchangeability is compounded by conditions in which automated equipment can vary greatly from installation to installation--especially in an environment in which machines can be owned, and their software designed, by individual personnel in the field, going with them when they move or otherwise become unavailable [e.g., personal computers owned by military people and used by them to help them do their (presently unautomated) jobs better].

c. Software Configuration Control

While it is recognized that highly centralized configuration control of application software is a development necessity, it is not adequately appreciated either in higher-level policy or at working levels that this control must be restricted to the functional requirements of the system and the control of the product after it is built. Flexibility in the development of systems under EA must be recognized in configuration control. Each increment must make use of the experience gained through previous development. Configuration control must provide a framework for this introduction of controlled changes based upon prior experience yet retain the rigors necessary for multiple fielded systems.

d. Meaning of Architecture

A serious and pervasive lack of understanding was found to exist regarding what the term "architecture" means when applied to C^2

systems. Acceptance and understanding of the fact that architecture can and should exist simultaneously at multiple levels in a C² program was further lacking. Such multiple levels range from the architecture of multi-system mission capabilities, spanning numerous user organizations both vertically and horizontally, down through individual military information system architectures to computer architectures (and families thereof), and even to instruction set architectures.

Part of the problem stems from the terminology being used with so many different meanings in the current literature. But this finding was identified by the Study Team not so much from what it found in the literature as from what it did not find when it sought valid reference data to satisfy the pervasive confusion it found on the subject, as well as on the related subjects of network and data communications architectures and models throughout DoD. To deal with this problem, the Study Team set up a separate sub-Team to provide the needed material as a major topic within what is now Chapter V of this report.

e. Designing for Change

Finally, since C² systems are subject to rapid change for a variety of reasons, the literature was found to give stress to the peculiar need for designing specifically for this very change, i.e., maximizing the ability of the design to be changed over time, in contrast to trying to optimize the original design of the system. It is so important that "design for change" be considerably more accepted than it is in the case of C² system acquisition that Chapter V deals with this problem as a major topic also.

3. Problems Derived From Interviews & Visits

Chapter I outlined the extensive number of visits and interviews conducted by the Study Team throughout DoD, in addition to those which were an inherent part of the case gathering (to be discussed in Section C). These visits and interviews, together with an analysis of the cases gathered, the literature search, and the integrated experience of the Study Team itself, formed the basis for the conclusions of this report. But there were some findings and conclusions about DoD current problems in the acquisition of C^2 capability that are particularly attributable to these oral discussions and the related briefings that were provided to the Study Team.

a. Participant Roles and Cultures

Acquisition of DoD systems involves a number of different functional groups. Principal among these are what have come to be known as developers, users, independent testers, trainers, and logisticians. As might be expected, over the years these groups have developed unique cultures of their own, which are as different as the well-known differences among the DoD Components themselves, and which are just as fiercely preserved. This is an acceptable situation for producing DoD's needed materiel and operational capability, as long as each of the cultural groups works more or less serially, independent of each other (i.e., user to developer to independent tester, back to user, and then to both logistician and trainer) and/or participates throughout most of the acquisition in a specialized way in an essentially off-line, but highly structured, fashion (as do logisticians and trainers in good part). But the acquisition of C^2 systems requires continuous, less-formal, and highly-flexible relationships among all the participants in the acquisition, as the C^2 program experience discussed in the next section of this chapter amply demonstrates. And so these cultural differences were found to be raising serious barriers of communication, as well as "turf" issues.

The outstanding problem uncovered here was the understandable concern of the developer for the loss, or considerable blurring, of his traditional role in system acquisition as a consequence of the substantial need for real users to play a considerably greater role in the acquisition of C^2 systems (to be discussed at length in Chapter IV). Hence the strong drive of developers to retain the <u>status quo</u>, even for C^2 systems. This "turf" concern went even to the extreme of developers worrying that if program funding was made more subject to the control of users, rather than remaining largely in the hands of developers as it now is, that developers would be excluded entirely from C^2 acquisition activities. And this in spite of the considerable reluctance found among some real users to play the more influential role in C^2 acquisition that is necessary, and their ready recognition of the specialized skills and the knowledge of the technology available from developers.

Other pertinent problems noted here, especially in the case of C^2 systems, were:

- The formality and serial nature of the relationship between the Air Force's Systems Command and its Logistics Command
- The insistence by the Navy that a combination of Navy Hq (OPNAV) and the independent tester (OPTEVFOR) are adequate to represent the real user's (the fleet's) viewpoint
- The Army's only recently emerging recognition of the early role needed to be played by trainers and logisticians in assuring the adequacy of programs from operability and maintainability points of view, and
- While the Study Team found increasing awareness, within the independent test and evaluation community, of the complexity and difficulty involved in C^2 system operational effectiveness testing, coupled with a willingness to allow the user to play a more significant role in the operational effectiveness evaluation of C^2 systems than normally has been the case, widespread belief was found (except in the Navy) that the so-called "independent" tester is the fiercest guardian of his prerogatives of all.

b. Business-As-Usual

A second significant problem area in C^2 system acquisition detected by the Study Team from the visits and interviews in particular was attitudinal in nature. People involved in materiel programs, who are outside of the technical "mainstream" of developing, testing, and producing the system needed (e.g., those involved in requirements determination and validation, PPBS, procurement, and program management activities such as planning and control), were found almost universally to believe that their work should be done the same way regardless of the type of system being acquired. That is, they see no reason to conduct their business in other than the usual way, simply because a C^2 system is being acquired. This attitude was found to extend even to the level of those who are responsible for making acquisition policy. A massive wall of resistance thus exists to making the changes in acquisition approach required to satisfy the particular acquisition needs of C^2 systems (or any other deviant program type, for that matter). These actions partly are deliberate. But the Study Team believes it also to be largely a matter of indifference and inertia, and hence of education.

The effect of this "business-as-usual" attitude will be discussed later in this chapter as regards the serious impediment it can be to enabling DoD to exploit the promise of an evolutionary acquisition (EA) strategy.

This attitude appeared to be the most rigid (worst) in the Army, as compared, for example, to the relatively more participative and experimental attitude found in the Air Force about adapting the attributes of what has come to be called evolutionary acquisition. However, "business-as-usual" inertia was found to be prevalent as a problem throughout DoD, especially as one gets away from the program offices who have to deliver the needed capability. This attitude also impinges on the issue (discussed in detail in Chapter IV) of the validity of using a surrogate for the real user on a C^2 system acquisition. The Study Team found that the real user has not been represented very well in C^2 acquisitions by such surrogates, especially in higher-organizational-level systems (echelons above Corps and equivalent echelons in the other Services--those most impacted by multi-Service and multi-national wartime considerations).

In sum, a major problem found by the Study Team was that C^2 system program management offices have to go to extraordinary lengths to get their jobs done, because they have to "negotiate truces" with each of the various functional groups outside the program office, in order for program managers to be able to operate in the way their judgment and experience with C^2 system acquisition dictates.

c. Joint/Multi-National Users

A final major finding of concern with current C^2 system acquisition that the Study Team derived largely from interviews and visits (because it is primarily an organizational problem), is the inadequate attention being given by the Services to the development of C^2 capability to serve joint and/or multi-national users. While a number of the more-important reasons for this situation are beyond the scope of this acquisition study, there were at least two things about the acquisition of C^2 systems that the Team found contributed to the problem:

1) Impact of Prior User Experience with ADP

The degree of user sophistication and understanding of the impact/utility of ADP (Automated Data Processing) on his ability to command and control varies as a function of whether the user presently has automated decision aids. The more he has, the more he can identify valid needs and other uses of value for it. But joint commanders have relatively few such aids, and multi-national commanders even less, because they do not have the funds to grow in this area. The Services (or Nations), who are their "executive agents" for acquisition purposes, understandably focus their attention, as well as their funds, on their own tactical C^2 needs.

2) Mission Needs vs System Solutions

The Services, even when willing, are largely organized to act as purchasing agents in their acquisition activities. They focus on buying things, i.e., on obtaining systems at the hardware/software level (such as radar systems and communication systems) not mission systems like defending the Continental U.S. against air and missile attack or locating and killing enemy tanks in which the C^2 capability involved needs to interface with weapons, platforms, and other C^3I systems to do some job. It is these missions around which the job of joint and multi-national commanders tend to revolve, in contrast to their subordinate, single Service commands and other lower echelons.1/

4. Problems Derived From Case Explorations

As indicated in Chapter I, case studies of a number of DoD C^2 systems were a very important source for the conclusions and recommendations of the Study Team. These cases are summarized in Appendix E. They were selected to provide a cross-section of C^2 system programs from each of the Services, as well as a joint program. The cases were intended to cover many echelons of users, from lower-level multiple tactical units to the highest levels of command; many types of users; many functional applications, ranging from weapon or platform control systems (such as TACFIRE) to top-level strategic force management systems (such as 427M); a wide range of quantity and dollar investment; and a spectrum of program outcomes from "failures" to "successes." Figure III-1 lists some of the general characteristics of the systems studied:

1/ A former commander of a major Service materiel command pointedly indicated, in response to a question, that the Services want to continue to focus on this hardware/software level of buying, because they fear loss of cost and program control if they allow the focus of DoD's acquisition efforts to be on what he designated as the "vague" mission system level.

SYSTEMS	QTY	FUNCTIONS	ECHELON(S)
JOINT			
-BETA/JTFP/ASAS ENSCE	3/3/20's	ANAL/FUSION TGT NOMIN	CORPS/ATOC DIV
ARMY			
-TOS/SIGMA	50's	FORCE MGT DECIS AIDS	CORPS, DIV BDE
-TACFIRE	50's	AUTO ARTY FIRE DIR & PLANNING	BN, DIV CORPS
-PLRS/JTIDS HYBRID	*30's 1000's	ICNI	ALL
NAVY			
-TFCC/OUTLAW SHARK	20's	FORCE MGT	TASK FORCE
-MIFASS	10-20	AUTO AIR & GRD FORCE MGT	MAW, MAF
AIR FORCE			
-OASIS	1	SPEC INT MGT TENCAP INTER-FACE	USAFE (EAC)
-427M + MTN COMPLEX	1	I&W + DEF MGT	NORAD (CINC)
-TACC AUTO	2		QUASI CINC
-CAFMS	2	TACTICAL BATTLE MANAGEMENT	
-CW	1		
-EIFEL	1		

* Master Control Units/User Units

Figure III-1. Some Characteristics of the C^2 System Acquisition Cases Studied

This stratified group of cases were then arrayed on a scale of relative "success" in terms of two criteria which the Study Team deemed as the sine qua non goals of the acquisition of a decision-support system like a command and control system:

- Whether useful capability was (or promises to be) put in the hands of the system's user(s) more quickly and more often as a result of the approach being taken; and
- Whether this user was satisfied with the system when he got it, in terms of agreeing that his capability to perform his command and control functions has been enhanced, how readily he can operate and maintain the system (including upgrading its application software over time); and the reliability and availability of the system under adverse environmental/military conditions in the field or at sea.

a. C^2 Systems Acquired the Traditional Way Were Failures

The single most significant DoD C^2 system acquisition problem identified through drawing up this "success" array was that the traditional DoD approach to C^2 system acquisition is the least likely way to achieve a successful result in terms of these two criteria.

While, understandably, all types of detailed arguments can be and were raised with the Study Team about whose fault it was and why any particular program got into trouble, it was compelling to note from the array that all of the cases that would have to be labelled either "failures" or "tending towards failure" (at least in their original approaches) involved the traditional approach to acquisition. This finding held for the programs of all four Services and the joint program examined--an observation which at least the C^2 communities of these four Services and DoD Hq have determined to be a major "lesson learned" of the past decade, judging from the revised approach each is now taking to current versions of these programs and to more current C^2 programs.

This lesson was most vividly conveyed in the analysis of attempts by all three Services to develop and field automated tactical operations centers.

- Army efforts toward a Tactical Operation System (TOS) date back to the early 1960s--and still have not resulted in a fielded capability. One Army approach (the Europe TOS) generally followed an evolutionary approach and achieved initial success, but was then moved from Europe to CONUS (III Corps) and lost support. Another approach following the traditional acquisition route (TOS/Operable Segment) was terminated. GAO estimated that at least $93M was spent on the unsuccessful TOS efforts.

- An Air Force program for Tactical Air Control Center Automation (TACC AUTO), based on a 1967 ROC, also mostly followed a traditional acquisition strategy. Although the system was judged a conditional success after testing, the serious problems encountered caused Congress to stop the program. About $80M was spent on the development. The ROC remains unfulfilled in the field.

- In the early 1970s, the Navy also followed a traditional acquisition approach in its initial program for a Tactical Flag Command Center (TFCC). After lengthy analytical studies and preparation of a detailed full system specification, a contract was awarded. At the completion of the design phase, estimated cost had tripled, the schedule had extended, and there was disagreement about the system capabilities. The program was then terminated, for "affordability" reasons. The Navy then took advantage of an evolutionary development which was in progress, Outlaw Shark, to form the "core" of continuing evolutionary acquisition of the TFCC.

Other cases examined also revealed serious difficulties when the traditional acquisition approach was followed for decision support types of C^2 systems. The Army's TACFIRE, although eventually fielded, experienced long delays and large cost growth, which were largely due to problems in developing complex software in a large step function rather than in small increments. The first systems did not reach the field until 13 years after award of the total package procurement contract, and only after strong involvement by user surrogates, the Army Field Artillery School at Ft. Sill, who, at the program manager's request, interacted intensively both at the contractor's plant and with the program office.

MIFASS, a Marine Corps automated tactical data system, used a nonoperational test bed in developing initial system requirements over a five- year period, prior to engineering development via the traditional method. That traditional approach will require at least 12 years from beginning of the test bed phase until completion of engineering development. Due to problems with MIFASS development, the Marines appear to be transitioning to an evolutionary approach for MIFASS.

A strategic system that followed the traditional DoD acquisition approach, the original 427M/Cheyenne Mountain Complex system also experienced serious problems. The system was fielded late and initially was not satisfactory to the user.

In addition to the difficulties cited above, the case studies identified other drawbacks from following the traditional acquisition process. Failure to use an architecture facilitating change and growth led to delays and additional cost in 427M, as well as in changing from TACC AUTO to CAFMS. Insufficient continuing real user influence has caused, or may be expected to cause, serious problems in TOS-OS, MIFASS, the original 427M, TACC AUTO and BFTA.

b. <u>Failure to Admit Uniqueness of C^2 Systems</u>

Although the traditional acquisition approach has consistently led to failure to field C^2 systems (three of them were cancelled, a fourth was threatened to be cancelled for lack of completion, and a fifth was subject to extensive high-level review), there is surprisingly little appreciation in DoD, outside of its C^2 community, of the underlying reason for the poor results achieved. Part of this is due to the natural reluctance of people to think about the need to change their approach in any significant way in an activity as complex as the acquisition of technologically-advanced DoD materiel. But there is decided evidence that part of the problem is due also to the pointed unwillingness of a number of people in influential positions in DoD to admit the Study's major finding from the literature (Section B.1 above) and from the experience of many of the people interviewed: that C^2 systems, like the automated information systems now being increasingly acquired for various military management purposes, are something different for acquisition purposes, something relatively unique.

This reluctance to recognize to the uniqueness of C^2 systems has been aggravated in recent times by the fact that the 1980 OSD policy on the matter is a bit too embrasive, both in terms of its six criteria and in its presumed breadth of application by type of C^3I system (both to be discussed in Section C.3 of this Chapter).

c. Other Lessons

Other important C^2 system acquisition problems, or negative "lessons learned," from the case studies conducted were as follows:

- Those programs over which users did not have a significant influence (not merely participate in) throughout the acquisition cycle tended towards the "failure" side of the scale. Two of the cases, in fact, dramatically illustrated the importance of this lesson by markedly shifting on this one variable of user influence during their acquisition. In each case, the results were materially better, albeit not entirely satisfactory from a lead-time or other points of view, during the period when a user was heavily influential in the program.

- The same holds for those programs in which an architectural approach was not followed which allowed for ready change and ease of technology insertion.

- Rigidly sticking to original, approved program goals or requirements in the face of both program events and the difficulty of stating requirements for C^2 systems in the first place (without being either too vague or too detailed) was found to be a significant cause of program delays and extra costs in some programs.

- Programs positioned on the "failure" side of the spectrum tended to be those which did not employ flexible "test beds" of some type on their requirements, development, and/or test processes--"test beds" running all the way from permanent system design/support facilities to *ad hoc* capabilities which merely demonstrate the feasibility and value in an operational environment of various commercial configurations.
- C^2 systems have interoperability needs which often go beyond the capability of its acquirers (developers and users) to provide for, in terms of adequate external interface control.

5. Inability to Evaluate

a. The General Problem

One of the major problems DoD has in the acquisition of systems in general is evaluating adequately what is essentially a collection of physically inter-related pieces of hardware and software of various types, called a "system," in terms of the collection's contribution to the accomplishment of the operational military mission which gave rise to its acquisition in the first place. That is, DoD acquires what it calls "ship systems"; but what it is really after is either the ability to transport men and materiel by sea safely and speedily, or the ability to project fire power as far forward as possible from platforms at sea that are made as invulnerable as possible. Likewise, DoD acquires "aircraft systems," but its real goal is to accomplish the same as it does under the second naval mission named (project fire power) or to help protect friendly ground combat forces.

The general evaluation problem for DoD is that it can measure relatively satisfactorily the speed, range, capacity, and other physical performance characteristics of the various parts of such systems. DoD can even measure to some degree how well various parts of systems work together physically to perform various operational military functions (such as detecting the enemy or communicating with friendly forces). However, DoD cannot adequately determine how much: (a) various components of the system, (b) increments of physical performance characteristics, or even (c) the whole hardware/software system, contribute to some overall military mission, as compared to other possible hardware/software systems or their parts. This is especially the case during the various earlier stages of a system's acquisition, such as the conceptual and advanced development stages.

b. Worse For C^2 Systems

The Study Team found ample evidence in the various sources of data it examined, as well as from its own directly pertinent experience, to support the finding that this general DoD evaluation problem is considerably magnified in the case of command and control systems of the type on which it focused. There are two basic reasons for this:

- The commander and/or his staff and related doctrine and procedures are the dominant element of such systems. This means having to evaluate for their contribution to the mission a part of a system whose attributes lend themselves least to measurement--the people part. It also means that the contribution of all of the other parts of the system must be measured in terms of their contribution to complex human processes like cognition and interpersonal communication.

- The evaluation criteria used in such cases must be highly subjective, because, in the final analysis, while data processing capability can be measured in more-or-less objective physical terms, information, by its very nature, can be measured only in terms of the degree to which it informs particular persons or groups--a highly subjective activity.

c. Traditional Approach Even Worse

These problems in C^2 system evaluation are independent of acquisition strategy used. However, when the traditional acquisition strategy was employed, C^2 system evaluation was found to become even more difficult because:

- It involved an attempt to state firm requirements for such systems when such was not really possible. This resulted too often in requirements statements which were either too vague to provide helpful evaluation criteria, or conversely which were too specific/detailed, and hence provided false, unjustified, or undesirable criteria.

- The traditional approach fixed these questionable requirements statements for long periods without relief unless and until a formal and lengthy requirements change process was undertaken.

As a consequence, the Study Team decided to end this section on problems it found in DoD's current acquisition of C^2 capability by devoting a separate appendix (Appendix G) to this very important problem of evaluation, particularly as it applies to C^2 system acquisition.

d. EA Facilities Evaluation

As a bridge to the next section (C) of this Report, which covers the promise of evolutionary acquisition (EA) for dealing with some of the problems described in this section, the Team also affirms that EA deals directly with the two problems of the traditional approach just described. Specifically, EA:

- Recognizes the continuously-evolving nature of the "requirements process" for C^2 systems, as well as for their acquisition process;

- Provides that feedback from test and evaluation (T&E) of small increments of capability in the user's environment will be the basis for defining and refining "requirements," thus making T&E a part of the requirements process itself;

- Deals with the expected rapid change of C^2 systems by encouraging a major focus of the system's design to be on accommodating such change within some flexible overall architecture. (See Chapter V for a discussion of "Designing for Change.")

e. Evaluation Hazards of EA

Finally, because evaluation focuses on operational military missions, it should be recognized that such missions and the missions of individual using commands are not the same thing (although they may be for a given user-commander). This difference is particularly important to keep in mind in an evolutionary acquisition, because the greater role of the user-commander in such acquisition could cause a focusing on organizational missions at the expense of tactical missions.

Another possible hazard, for multi-user systems, is the possible "bias" introduced by using a representative "lead" real user to interact with the provider, with the interests of the other users "protected" only by a user surrogate, such as TRADOC, OPNAV, or Hq TAC. The Study Team believes the benefits of a properly selected and motivated "lead" user far outweigh this hazard.

C. THE PROMISE OF EA FOR RESOLVING C^2 ACQUISITION PROBLEMS

1. Case Data Summary Results

As indicated earlier, the Study Team reviewed a number of C^2 system programs to determine the acquisition approach that was followed in each case, the results that were achieved or anticipated, and lessons that could be learned from the conduct of this stratified list of programs. Then, as discussed in Section B of this chapter in terms of problem cases, the programs were arrayed on a scale of relative "success" on the basis of essentially two basic criteria:

- Whether useful capability was (or promises to be) put in the hands of the system's user(s) more quickly and more often under the approach taken; ari

- Whether this user was satisfied with the system when he got it, in terms of enhancement of his capability to perform his command and control functions, his ability to operate and maintain the system (including upgrading its application software over time), and the system's reliability and availability under adverse environmental/military conditions in the field or at sea.

The complete "success" array is shown in Figure III-2, taking into account the fact that various programs studied are in various stages of completion. (Some cases represent restructuring of what were about 10 basic programs):

FAILURES	TENDING LESS SUCCESSFUL		NEUTRAL		TENDING SUCCESSFUL	SUCCESSES
TOS-OS	BETA				ASAS	TOS-EUROPE
ORIG TFCC	MIFASS	⟵	TACFIRE	⟶	ENSCE	OS-TFCC
TACC AUTO	ORIG 427M	(BEFORE USER)		(AFTER USER)	SIGMA	OASIS
					EIFEL	CURRENT 427M/CMC-U
					CONS WATCH	CAFMS

Figure III-2. Program Success Array

Cases judged as failures or tending to be less successful were discussed in Section B.4, with the principal lesson learned from this review being that taking the traditional DoD approach to the acquisition of a C^2 system is the least likely way to achieve a successful result in terms of the two basic success criteria given above. Conversely, the lesson derived from studying the more successful cases, which are in various stages of completeness, is that applying variants of what is described in Chapter I as the Evolutionary Acquisition (EA) approach to C^2 programs shows much promise in terms of meeting the two success criteria. In fact, in contrast to the less successful cases, all of the more successful cases applied the EA approach in varying degrees. This can be seen in Figure III-3, in which all of the cases are categorized on the basis of the degree to which they were deemed to be taking the EA approach in terms of five attributes of EA:

(1) They are architected to accommodate readily to growth, change and insertion of new technology

(2) Their requirements process is being abbreviated and expedited

(3) They involve an initial "core" capability deployed for test in the user environment followed by subsequent discrete increments, rather than being one total program

(4) The program is incremental and the increments are relatively small

(5) Feedback from user operational testing is the basis of both revising prior increments and designing/specifying requirements for future ones.

Based on these attributes, a score of five in Figure III-3 means that a full EA approach is being taken to the program, a score of 3 or 4 that the program is more EA than not, and a score of 0 through 2 that the program is (or was) more traditional than not. Some programs are labelled "planned" because they are in an early stage; hence no definite results could be put down for them as yet.

C^2 Program	Degree EA Used	Strong Real User Influence on Development	Result	
			Initial Capability Sooner?	User Satisfaction on Receipt?
*TOS-Europe	4	yes	yes	yes
TOS-OS	0	no	no-cancelled	no
*SIGMA/MCS	4(P)	yes (lead user)	planned	yes
TACFIRE	1	yes (surrogate), but late	no	eventually
Orig TFCC	0	some	no-cancelled	no
*OUT SHK/TFCC	3-4	more	yes	generally yes
MIFASS	1	some (non-ops test bed)	late	?
*OASIS	4	yes	yes	yes
Orig 427M/CMC	1-2	some	late	no
*Later 427/CMC	4	yes	yes	yes
TACC AUTO	1	some	no-cancelled	no
*CAFMS	4	yes	yes	generally yes
*CONSTANT WATCH	3-4	yes	yes	yes
*EIFEL	3-4	yes	yes	yes
BETA	2	no (but Strg Comm)	no	?
ASAS	3(P)	more planned	?	?
ENSCE	4(P)	more planned	?	?
*Denotes programs deemed more evolutionary than not. P = Planned.				

Figure III-3. Case Data Summary[1/]

1/ PLRS/JTIDS Hybrid (PJH) was one of the cases studied. However, it was judged that only the friendly situation element of that system falls under the Study Team's definition of C^2; and so PJH was not studied further.

While it is still early from an experience point of view, the favorable results being observed or projected as a result of the application of EA are not really surprising. This is because EA was not applied as a full-blown, specific new technique by DoD. Rather, as the case gathering amply illustrated, it reflects the fact that as each DoD Component experienced the negative results described in Section B, they began to experiment pragmatically with the adaption to pertinent programs of various <u>attributes</u> of what later collectively came to be called "evolutionary acquisition", such as the five listed above (p III-24).

Thus the Study Team's case gathering results reflect a snapshot in time in which past unacceptable results in the acquisition of various C^2 programs (usually early-generation programs for the particular DoD Component), has led to marked revision in the approach being taken to meeting the still-existing need which gave rise to these early-generation programs in the first place. It has also led these Components to begin consciously to apply this fundamentally new heuristic or adaptive design approach to some of their later generations of C^2 programs from the beginning.

More specifically, the three outstanding findings from a review of the more "successful" cases--all of which took or are proceeding on the basis of an EA approach as an acquisition strategy--are as follows:

(1) First (and foremost) is the need for heavy and continuous real user involvement and influence from the beginning of the acquisition, in the sense of an approach in which the user has significant influence over the design of the system. All of the "success" cases illustrated this finding. And one additional program which did not use the EA approach (TACFIRE) improved eventually when a knowledgeable user surrogate (Artillery School, Ft. Sill) was introduced into the program. Conversely, one program which started out at a real user's site with success (TOS-EUROPE) went negative when this condition changed.

(2) The second is that fielding a "core" capability first, and then providing increments of capability based on reaction to earlier ones (rather than trying immediately to develop and produce a total "final" overall configuration of the needed capability), breaks requirements and technical opportunities down into "bite-sized" increments that are both more manageable and more assimilable for all of a program's participants. Further, the deployment of such an early "core" capability assures closer real user involvement from that point on in the acquisition cycle (if not earlier). Finally, the cases provided clear evidence that new capabilities can be gotten into the field in substantially less time under EA, even though part of this lead-time gain can be attributed, so far, to adaptations of some of the work done in earlier attempts to satisfy the same need. The value of this early "core-feedback-subsequent-relatively-small-increments" aspect of the EA approach was particularly found to be illustrated by cases such as TOS-EUROPE, SIGMA, OASIS, CAFMS, EIFEL, and the current CMC upgrade.

(3) Third, it was found that EA's promise to field useful capability earlier and more often results, in part, from the encouragement it provides to use already-developed commercial or military hardware and related operational (or system) software. When coupled with a flexible architectural framework designed to facilitate growth and readily allow for the insertion of new technology, EA facilitates redesign with minimum negative impact on the existing system. Thus, it was found, under EA, users are less likely to feel that they have to "go for broke" each time in their requirements statements, asking for capabilities that force developers to stretch the state-of-the-art, because under EA it will not take as long to achieve an operationally meaningful increase in C^2 capability, and each increment can be more easily assimilated by the user.

Regarding the notion of quickly fielding a "core" capability, TOS/Europe was available for a 7th Army exercise in three years. OASIS, CONSTANT WATCH and EIFEL became (or are expected to become) operational about five years after the programs were launched. CAFMS took only two years from program start to acceptance for deployment (CAFMS software top-level design was derived from TACC AUTO.) SIGMA has taken two years from program initiation to limited employment in VII Corps of TCS/TCT (again, capitalizing on elements of the cancelled TOS/OS program).

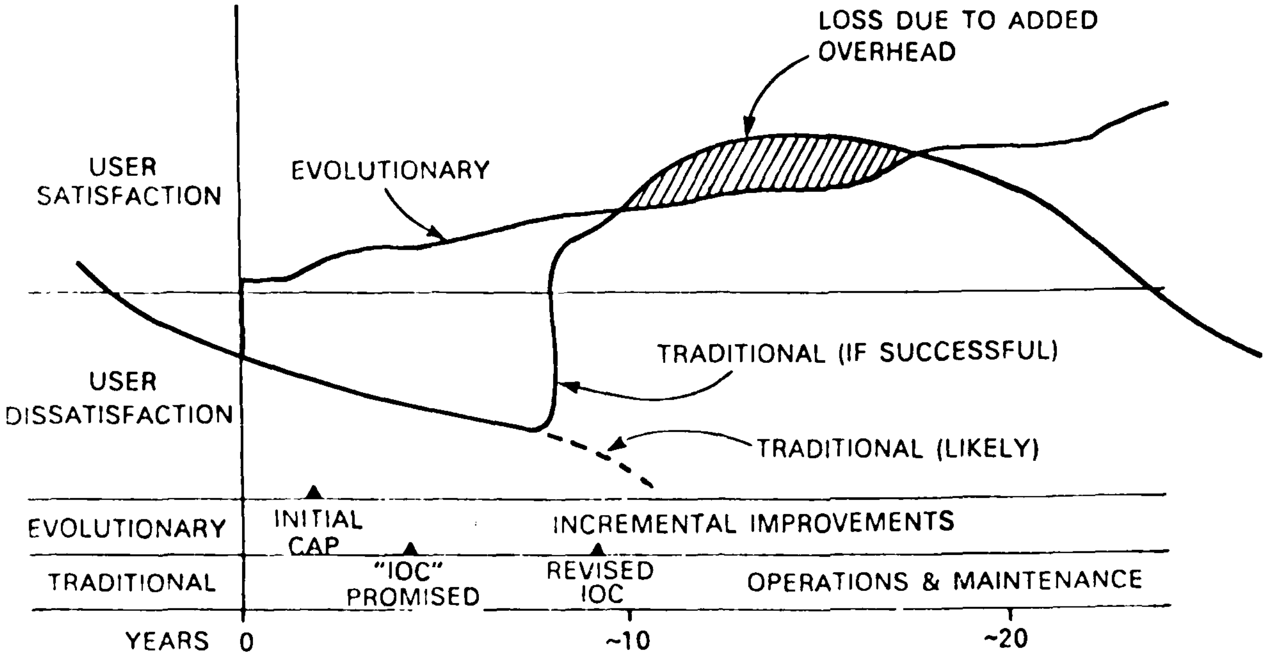
In sum, the cases provided clear evidence that new capabilities can be gotten into the field in a substantially lesser time under EA, even though part of this lead-time gain admittedly can be attributed so far to adaptations of some of the ork done in earlier attempts to satisfy the same need, or to use of other already-developed hardware or software. In this regard, it is the Team's view that, whenever feasible, new C^2 systems <u>should</u> employ available hardware and software as one way of fielding a useful initial increment in operational capability in 2-5 years <u>rather</u> than the 10-14 year average required to field a "total solution" acquired in the traditional way.

Combining good and bad lessons from the case studies provides convincing evidence that DoD can reap valuable benefits by applying evolutionary acquisition as it strives to improve C^2 system capabilities.

2. Why EA Increases Probability of C^2 Program Success

a. Conceptual Comparison of EA Versus the Traditional Approach

The answer to the question of why the case findings turned out as they did, and in general why EA looks so promising, if practiced properly, can first be viewed conceptually. Figure III-4 is an attempt to compare conceptually how user satisfaction is believed to change during the life cycle of a program when it follows the evolutionary approach as contrasted with the traditional approach. While the comparison is highly generalized, it does convey important insights. (The curves should not be deemed to be precise, but merely conceptual.)



*FROM "MANAGEMENT OF C^3 IN DOD" DR D ALBERTS, DR S STARR, SEP '81

Figure III-4. Conceptual Life Cycle Comparisons

Programs are born primarily out of dissatisfaction -- either dissatisfaction with the capabilities being offered by the current system (judged against either some need or what is perceived to be feasible), or dissatisfaction based upon projected future needs or technological growth that is foreseen. Programs begin slowly, first by developing an idea or concept, then by coalescing the necessary support to initiate programmatic advocacy. A program cannot get started without some measure of user dissatisfaction ("years = 0" point). Starting at the "years = 0" point, in the TRADITIONAL acquisition life cycle, a system is designed and developed in a number of sequential and time-consuming steps, beginning with preparation, coordination, and validation of requirements for a "total" solution, through necessary budget and program approvals, and serially through the Conceptual, Validation, Development, Production and Deployment/Use/Support phases of the acquisition cycle. Throughout this period, which can span many years, the real user usually is kept at "arm's length" to the new program, and his satisfaction with his current system steadily decreases (TRADITIONAL curve). At some point in time, however, the development community gives birth to an initial operational capability (IOC), at which time the user's satisfaction starts to rise sharply, even though this IOC often is achieved much later than promised during the original program advocacy (the TRADITIONAL (IF SUCCESSFUL) curve).

Often in a traditional acquisition, system design has not kept pace with the changes which have taken place over the five to ten years since the user community first was involved in the articulation of system needs -- changes in the threat, potential war scenarios, available technology, etc. And so the system delivered is not acceptable to the user as being capable of meeting his current requirements as he sees them. These systems are deemed "failures" by the user, and his dissatisfaction with his current capability thus continues to go down (the TRADITIONAL (LIKELY) curve). As indicated earlier, most of the C^2 systems reviewed by the Study Team that followed a TRADITIONAL acquisition strategy fell into this category.

Even for traditional programs that are deemed "successful" (i.e., those following the TRADITIONAL (IF SUCCESSFUL) curve), a significant period of adjustment is required after IOC, in order to allow the user time to get to know the system, and the developer in turn to respond to articulated needs for modification and change. At this point in the life cycle of the system, even though user satisfaction is still increasing, if the system design did not take place in an architectural framework that can accommodate change, there eventually comes a time when changes no longer can be made satisfactorily. At this point user satisfaction starts once again to decrease, even though the capabilities of the system have not necessarily diminished--indeed, they may even have improved since IOC (except when the software has been patched and repatched to the extent that it becomes both undocumented and "unreliable"). And the "cycle" then repeats itself. This anomalous situation usually represents a divergence between user aspirations and system capability. And user aspirations are driven, among other things, by his perceived need and by the increased expectations that he has with respect to "available" technology.

In any case, since the traditional acquisition approach usually has not been based on an architectural framework which can accommodate change, user dissatisfaction will continue to grow at this point until the system is replaced. Experience with TRADITIONAL design and acquisition approaches thus leads to the conclusion that there is, within every program's life cycle, at best only a short period in the "middle" of the life cycle (no more than a half) in which the user is reasonably satisfied with his capability to do his C^2 job. During the remainder, there is considerable dissatisfaction.

However, compare the attributes of the TRADITIONAL acquisition approach to what has become known as an EVOLUTIONARY acquisition approach with respect to user satisfaction. In this approach (EVOLUTIONARY curve), the aim is to get the real user (or lead user) intimately involved in the design of the system and in the test and evaluation of (alternate) system concepts right from the beginning--and keep him involved. This user involvement, as opposed to the usual "arms-length" relationship between the user and the provider in the traditional approach, will, it is believed, account for an immediate increase in user satisfaction (the vertical jump in the EVOLUTIONARY curve at "years = 0"). Since, too, an evolutionary acquisition aims at providing a fieldable capability in the near term, it can also be expected that user satisfaction will continue to increase, to the extent that visible progress is being made on providing him with an initial capability. Finally, since the user has been involved in a closely-coupled, interactive process with the provider, and since the IOC time of the initial capability is relatively short, it also can be expected that the initially-fielded capability will largely meet user needs and, hence, not be rejected or disdained upon delivery.

One of the cornerstones of the evolutionary approach is the provision of a system architecture which is designed to accommodate change. Given such successfully-implemented architecture and an evolutionary philosophy, the system can be expected to continue to grow and be enhanced readily in a series of relatively small and closely spaced (in time) increments. Thus, the user is not asked to wait a long time (given technical and budgetary feasibility) to see the implementation of the improvements which are being derived from the learning on his and the provider's part which occurs with use and experience. In fact, a properly-designed architecture should be able to transcend several generations of subsystem hardware, as well as users, before its replacement is required.

There is one possible rebuttal. The "LOSS DUE TO ADDED OVERHEAD" part of Figure III-4 indicates that although some capability is being fielded sooner and more often under the EVOLUTIONARY approach, one could expect that if the TRADITIONAL approach were fully successful, at some point in time more total capability would have been fielded more efficiently than if the total program were decomposed into small pieces/increments, thus providing greater user satisfaction for a period. However, the Study Team believes that the overall length of the period of user satisfaction even in this totally successful traditional case will be much shorter than under the EVOLUTIONARY approach. In fact, there is in EA at least the possibility of continuous user satisfaction with his C^2 capability, as the curve indicates.

b. Specific Contributions of EA

Turning from the conceptual to the more specific, as previously indicated, the single most significant contribution of EA to the acquisition of C^2 systems is likely to be its ability to deal adaptively and iteratively with the inherently evolving needs that characterize the requirements determination process for a C^2 system. In fact, there is really no other way in the usual case. For such systems, the user's ability to state what he needs is a distinct function of his actual experience with the highly technical, automated capability that he has at any given time to aid his command and control functions. He recognizes that his ultimate goal is not to replace the physical things that help to comprise such capability but to up-grade his ability to perform his command and control functions over time. Thus he and other members of the acquisition team are engaged essentially in an interactive, "design-and-try out" process, in which stating what is wanted is accepted as being as much in a state of constant revision as is the means proposed for satisfying it.

A closely related contribution that can be expected of EA derives from the speed with which such C^2 requirements ordinarily change, as a result of both:

(1) The unusual rate of growth of commercial technology in the information system field and the compelling opportunity for increased military C^2 efficiency that this provides. Several generations of new information technology are being made available within the length of time of the acquisition life cycle of a single C^2 system program, if the traditional approach is followed . This new technology was found by the Study Team to be eagerly sought after, even by uninitiated users, since it would make their job so much easier and productive.

- More immediate and more nearly continuous user satisfaction with what he is getting, due both to his close and continuous coupling with the acquisition effort and the greater ease with which he can assimilate the smaller increments of capability involved.

- The reduced Government risk of program failure and less financial exposure involved in both proceeding in small increments and in focusing on available commercial or military materiel as much as possible, rather than on taking one large revolutionary jump towards the limits of the art each time a program starts. (See the discussion in Chapter I of the difference between EA and P^3I in this regard.)

- System architectures that readily can accommodate change and which facilitate easier technology insertion, and which, hence, can be expected to reduce C^2 system obsolescence, extend useful system life, and allow for upgrading the existing overall C^2 capability of a commander with minimum disruption.

d. A Commercial Comparison

It may be instructive in closing this section on why EA increases the probability of C^2 program success to take note, from the Study Team's review of the literature, of the findings of a study of non-military decision support systems (DSS) by an international task force of the International Institute for Applied Systems Analysis (IIASA) which included consideration of about 30 cases.1/

That study resulted in major findings that are quite similar to those evidenced in the military case studies. It concluded, for example, that:

- DSS are often difficult or impossible to define;

- A short feedback loop is required between the designer-developer and the user, with frequent repetition of a single development cycle;

1/ Fick, G. and Sprague, R. H. Jr., "Decision Support Systems: Issues and Challenges," Proceedings of an International Task Force Meeting, June 23-25, 1980, Pergamon Press.

- Development should be started initially with the identification of a small, critical subproblem or set of decisions;

- The resulting system should be used and evaluated for a short period of time before development goes on, and

- This evaluation should be used to guide the next cycle of analysis, changes, additions, and deletions that expand or redirect the system's capabilities.

The referenced IIASA report included a paper by Professor Peter G. W. Keen of MIT's Sloan School of Management. His research on computer-based "decision support systems" (DSS), has arrived at essentially the same process as evolutionary acquisition. He used the concept of "adaptive design," which states that "the final system must evolve through usage and learning." Keen's "adaptive design" emphasizes:

- Starting with a prototype that provides something concrete for the user to react to and experiment with. The prototype is a real system, not a mockup or experiment. It provides a basis for learning-by-using.

- Paying careful attention to the user-DSS dialog, the encouragement of user learning, the evolution of the system, flexibility in the DSS, and responsive service by the system builders. In essence, the system design must be "user friendly."

- First, building the initial system ("Version 0") [our "core"]; then extending and improving it in response to the user's reactions; finally, creating the stable, documented, and reproducible system product.

In his concept of "adaptive design," Professor Keen has captured the essence of what the Study Team means by "evolutionary acquisition."

While not surprising that the main findings in the military case studies herein closely parallel the conclusions of a diverse scientific group studying non-military decision support systems, it does strengthen the Study Team's belief that the lessons learned from the military case studies are sound and form the basis of the prescription for increasing the likelihood of success in C^2 system acquisition.

3. Types of C^2 Systems Most Suitable for EA

a. Criteria for Application of EA

1) Current "Criteria" in Section 13, DoDI 5000.2, March 1980

All of the "criteria" listed in Section 13 of DoDI 5000.2 of March 1980 (see Appendix D) are valid characteristics of C^2 systems. They become criteria for the application of EA only when related to making some decision, however. And the decision to be made in applying Section 13 is when to resort to "special management procedures" in the acquisition of a particular class of military materiel, principally when to take an evolutionary approach to such acquisition, with related concomitants such as: (1) markedly increased user involvement with and influence over the acquisition, (2) the elimination of counter-productive official phase distinctions in the early part of a C^2 program, and (3) use of flexible T&E supporting facilities (in some cases called "test beds").

In theory, the criteria listed in Section 13 on when to adopt such special procedures apply to any DoD program. And some of the tenets of Section 13 should be considered for optional application in programs other than C^2 systems. However, the section is focused on "Command and Control Systems", to highlight the fact that this class of system usually ("in most cases") benefits from such an approach.

What then should the criteria in Section 13 be? This question can perhaps best be answered by examining the six so-called "criteria" currently listed in Section 13. These are:

- A rapidly-evolving technological base
- Multiple requirements for internal and external interfaces
- Reliance on ADP hardware and related software

- Acquired in small numbers, in some cases only one of a kind
- Their operational characteristics are largely determined by users in an evolutionary process
- Commercial equipment exists that can emulate the function.

The Study Team concluded several things about this list.

a) <u>Three of the Existing (March 1980 5000.2) "Criteria" Are Suspect</u>

First, the Team doubts whether three of these so-called "criteria" in fact do, or should, affect the decision whether to take the evolutionary acquisition (EA) approach. These three, in decreasing order of doubt, are: "acquired in small numbers" (the most dubious), "commercial equipment exists that can emulate the function", and "reliance on ADP hardware and related software".

Regarding "acquired in small numbers", the fact that C^2 systems often are acquired in small numbers does make it difficult to justify the expense of developing a prototype, but this fact does not make the desirability of such prototype, as a normal acquisition step, any less valid. The Study Team therefore concluded that the "small numbers" characteristic of C^2 systems should be taken into account in their acquisition not so much as a <u>cause</u> for resorting to EA but as an important shaper of how the EA is conducted as regards ensuring the supportability and procurability of the C^2 system being acquired.

Likewise, the fact that commercial equipment can do part of the job can save military development effort. But savings through the use of commercial equipment should be, and is, sought in <u>other</u> military materiel programs where possible, not just C^2 programs. Thus whether "commercial equipment exists that can emulate the function" is not considered to be a criterion for deciding when to use EA.

Finally, the fact that modern C^2 systems rely on ADP hardware and related (support) software introduces an entirely new technological element for the acquisition process to contend with, and therefore leads to the need to review each step of this process in a critical, innovative way. This does raise the possibility of having to resort to "special management procedures". But it is <u>not</u> the machines and their support software <u>per se</u> which cause this need, because they can be acquired readily in a relatively normal, albeit tailored, procurement manner. Rather, it is the need to acquire them in such a way that they serve to enhance the ability of a commander to perform the functions of commanding and controlling that calls for special management procedures. And this last is largely a function of the system's architecture, on the hardware side, and its <u>applications</u> software, not its support software.

b) <u>Software Dominance Alone is not a Sufficient Criterion</u>

This leads to the second conclusion: that the (applications) software-dominance criterion frequently used to justify the use of "special management procedures" (such as EA) in the acquisition of C^3I systems in general can be misleading if the <u>role</u> of such software in the system is not kept strictly in mind. If the purpose of such software is to <u>aid</u> the <u>person</u> in the system to perform human functions (e.g., commanding and controlling) better, then special management procedures <u>are</u> appropriate. If, however, its purpose is to <u>reduce</u> the role of the human in the system as much as possible (as it is, for example, in an automated control element of a communications system or by ADP embedded in a weapon system), then they are not necessarily needed (albeit might well be considered desirable in a particular case).

c) The Dominant Criterion

Above all, the Team believes that the dominant criterion for taking an evolutionary approach to the acquisition of a true C^2 system is the fact that "their operational characteristics are largely determined by users in an evolutionary process". That is, the strong need exists in such cases for some user to play a significant-to-dominant, iterative role throughout the acquisition of the system, both because neither he nor anybody else can adequately specify in advance what is needed and because it is his particular and shifting operational needs, style of management, and geographical and resource constraints that are to be embodied in the system's application software. The Study Team considers all other criteria to be secondary to this one.

2) The Appropriate Criteria for Application of EA

In view of the foregoing, the Study Team recommends that the criteria for using EA and related "special management procedures" in the acquisition of C^2 systems should be keyed to only those few characteristics of these systems which distinguish them from systems to which conventional acquisition approaches can be applied. Principal among these few are:

- The need for the system's operational characteristics and value to be largely determined by users in an evolutionary manner,
- The need for the acquisition ordinarily to take into account an unusually high number of complex internal and external interfaces at multiple organizational levels,
- The orientation of the system's application software towards facilitating the role of the human in the system in a "brain-aggrandizing" or "mind-extending" (decision-support) way.

The more of these three characteristics a program has, the more it must follow the EA approach.

3) Guidance to Program Managers

Besides formal criteria, guidance should be provided to program managers on when to take an evolutionary approach to the acquisition of C^2 systems. This guidance can be stated as a set of minimum program conditions which, unless satisfied, ordinarily call for EA to be applied (and conversely helps to determine when it need not be applied, even in the case of C^2 systems). Stated as a rule, this might read something like: "C^2 systems shall be acquired in an evolutionary manner unless all of the following conditions are satisfied:

- The requirements are definite.
- The user is satisfied with the completeness of the requirements specification.
- Requirement changes are not expected to be rapid or extensive during the useful life of the system.
- The user can specify acceptance (quantitative operational utility) criteria for the system which others can be expected to apply objectively to measure operational mission performance.
- The user's role can be minor during development.
- There is an insignificant amount (relative to total program size) of man/machine interfaces and new software development involved in the program, the latter of a type which is highly interactive with the decision process."

b. The User as an Acquirer

Section 13 of the March 1980 DoDI 5000.2 mandates that "the design and testing of (command and control) systems should, in most cases, be accomplished in an evolutionary manner". But, as recommended above, this mandate should be limited to just those C^2 systems which have the three basic characteristics given above (Section C.3.a.2)). One might ask: "What types or categories of C^2 programs have these characteristics and hence call for an evolutionary approach to acquisition (EA)?" or "Which classes of programs satisfy the limited number of criteria that the Study Team feels should be in the policy statement?"

The Study Team's answer is: Those which are a highly people-oriented information, decision, or management/force planning and control aid to a given commander and/or his staff in the performance of military functions. In these cases, the commander is himself the central element of the system being acquired and hence he (and/or his staff) needs to influence its acquisition to a degree far more than he would as an ordinary user. In such cases, he needs to be an acquirer, as well as a user.

A user's role as an acquirer refers to at least three things:

- The degree of initiative he (and/or his staff) is expected to take in the acquisition, in terms of such things as development planning and design;
- The degree of management control he (and/or his staff) is to exercise over the program at various stages, in terms of such things as program direction, rate of planned progress, and resource allocation, and
- The degree of direct responsibility he (and/or his staff) bears for program results.

Not all C^3I systems are "command and control" systems, but the Team concluded further that classes of such C^2 systems can be distinguished in terms of this degree of needed user involvement in the acquisition. These classes range all the way from the type in which some commander plays only an over-ride or other judgmental role in an otherwise highly-automated C^2 system, like an air defense system, to top-level force management systems of the strategic planning type in which the system's functions must be performed largely by men rather than by machines.

Figure III-5 illustrates conceptually this spectrum of degrees of needed user influence over the acquisition of a C^2 system in relation to that retained by others in the process (developers, testers, and logisticians). This figure shows where various classes of C^2 systems lie, in a range bounded by "Other DoD Systems" on the one extreme, to a type of system on the other which a military commander, like any other manager, may automate, but which is independent of his strictly military functions--a so-called "Administrative Decision Support System."[1/] These are not simply "housekeeping" systems, such as those used for keeping personnel records, providing inventory control, or doing payrolls. Rather, they are decision aids to high-level functional managers in the financial, human resources, and "corporate" planning areas, to use examples that are as applicable in a military environment as in a commercial environment. They form an upper boundary because the approach of such managers can be so different (higher-level management is an "art") that this type of system is considered to require the most user influence over its acquisition.[2/]

1/ Some may call these "Automated Management Information Systems," but we use the term Administrative Decision Support System because MIS tends to have a very broad meaning to include many administrative routine systems, as well as systems which need extensive user influence in the design.

2/ Note that while this figure portrays a "zero-sum" relationship among the participants in a C^2 system acquisition, in terms of the specific areas of influence listed, it does not intend to imply anything about proportionate or absolute numbers of user people involved or any other resources allocated to the program for any period. Nor does it imply anything about control of the funds involved. Developers, testers, and logisticians may well dominate in these respects at pertinent stages of the acquisition cycle, either directly or through assigning people and providing specialized assets to other participants in the effort. Also, although the curve is "thin" and, thus appears to be precise, the reader should recognize the curve is only conceptual, to illustrate a needed trend in user influence.

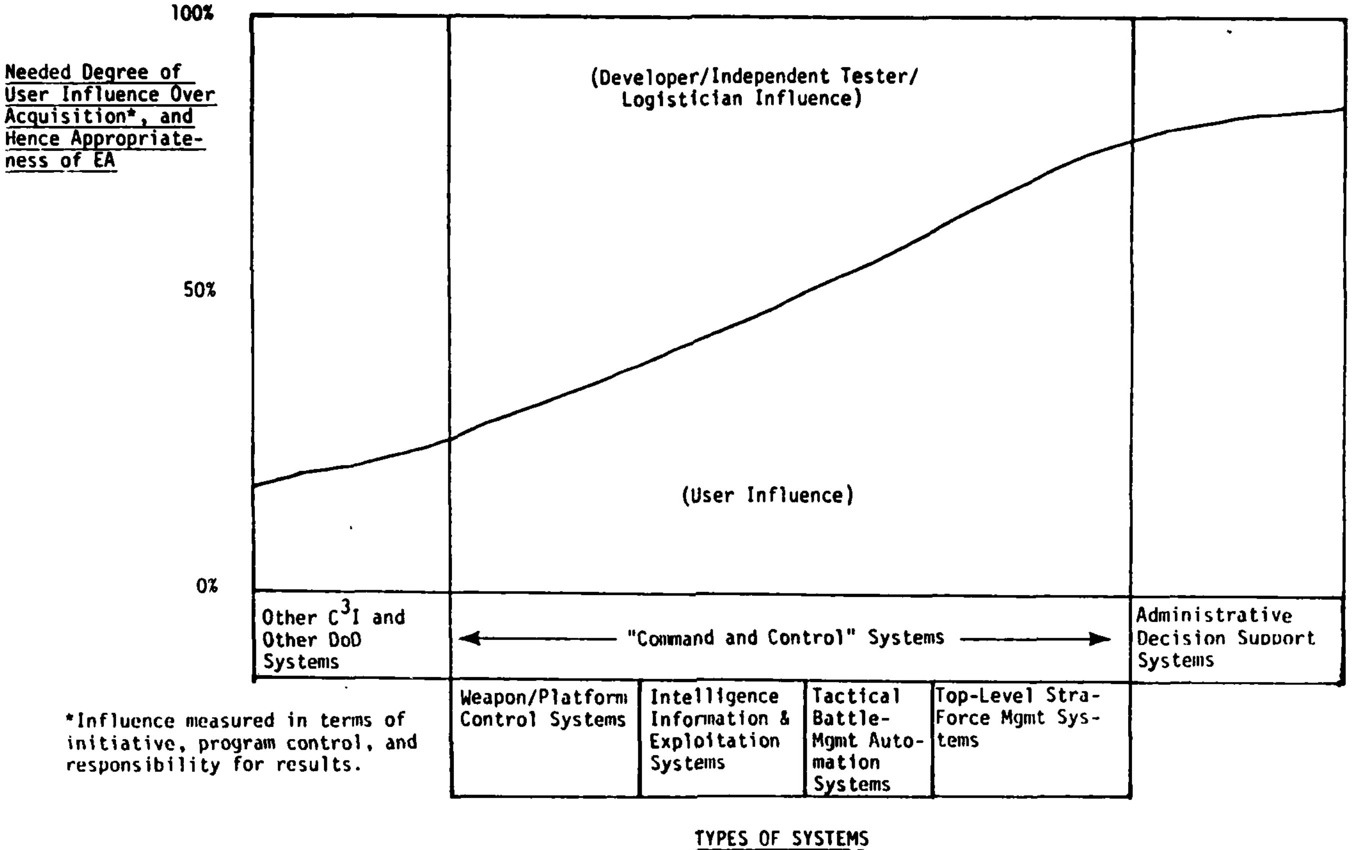


Figure III-5. Degree of Needed User Influence Varies with System Type

As Figure III-5 illustrates, examples of the increasing order of needed user involvement as an acquirer of various possible types of systems in the "command and control" category are as follows:

- Automated systems for controlling weapons and platforms (e.g., those used in air defense)
- Intelligence information and exploitation systems
- Tactical battle-management automation programs
- Top-level strategic force management systems of the NCA, the unified and specified commands, and the principal operating elements of both.

For these classes of C^2 programs, the burden-of-proof should be on program management to justify explicitly why they are not using EA and related DoDI 5000.2 Section 13 procedures--in those cases in which they elect not to.

Examples of the "other" types of C^3I systems listed on the left side of the figure are as follows:

- ADP embedded in weapons, platforms, or in communications control elements
- Common-user communications systems
- Data links
- Sensor systems of the stand-alone radar type or used in process control applications such as fire control, flight control, and navigation
- Electronic warfare and counter-C^3 systems.

In these other types of C^3I programs, and even in certain non-C^3I programs EA may well be desirable at times, in whole or in part. This desirabilit will occur when they satisfy one or more of the three criteria fo application of EA listed earlier. But EA should be elective in these cases, merely one of a list of possible acquisition strategies that a program manager may choose to adapt to his or her program.

4. No Single Optimum EA Strategy

a. EA as a Spectrum of Acquisition Strategies

There is no single, specific approach to evolutionary acquisition. Rather, EA is a broad acquisition strategy encompassing a spectrum of possible approaches. The approach taken should be tailored to the particular circumstances of a program, and EA not used when program circumstances do not satisfy the criteria discussed in Section C.3.a.2) on page III-41.

Because EA can be described procedurally in terms of an initial or "core" capability and subsequent increments or blocks of effort (all based on user feedback), there is a tendency to think of EA as a single specific approach. However, EA can vary in a number of important aspects from program to program. EA can, in fact, cover cases ranging from small, user-dominated upgrades of existing C^2 capability with items that he adapts (with the advice of the provider) from essentially off-the-shelf material, to large increments, involving significant amounts of development, in which both users and providers are heavily involved in an iterative way.

Figure III-6 illustrates this range of possible variation, for just two dimensions:

(1) Increment size (three possibilities are presented: small, moderate, and large) and

(2) The relative dominance of the user and provider, in terms of initiative, program control, and degree of responsibility for results (three possibilities are presented).

While Figure III-6 shows only three cases, theoretically there are nine possibilities for the two variables (size, dominance) presented. (Actually, there are many more, as one varies both dimensions more precisely with the actual needs and phases of a particular program.) There are also other possible variables that could impact the specific EA strategy chosen, such as:

- The number of timely interfaces with other programs that are required,
- The rate of commercial development of the particular technology involved in the program
- The organizational levels at which the program's results are intended to be used.

Case / Parameter	CASE 1	CASE 2	CASE 3
Increment Size	• Small	• Modest	• Large
User/Provider Relationship	• User Dominant • Provider Involved	• Provider Dominant • User Involved	• Both User & Provider Heavily Involved in and Iterative Way
Impact on Ops Concept	• Essentially within current ops concept	• May involve adjusting ops concept	• Involves major change in, or new, ops concept
Program Focus	• Focuses on: - People, procedures - Tactics - Use of existing resources	• Involves experimentation with new equipment or with modified ops concept	• Calls for Major developments of equipment
Technology	• Adequate technology is available	• Involves some development but mostly application of technology	• New technology has to be developed
Development Locale	• Can develop with facilities and people on site	• Must do development in laboratory with off-site people	• Requires industry and/or special development facilities
T&E Focus	• Can test and evaluate at user site	• Requires development laboratory type testing plus user T&E	• Requires elaborate test program, incl T&E in user environment
Outcome	• "What you test is what you get"	• Militarized version may be required	• Involves a big procurement to replace a prototype

Source: Adapted from non-published paper by W. Melahn, MITRE Corp., Dec 1979

Figure III-6. Examples of Alternative EA Strategies

Finally, the appropriate EA strategy could vary from increment to increment in the continuous upgrading of a needed C^2 capability.

b. Representative Types of Programs

Looking at Figure III-6 on a case-by-case basis, Case 1 might represent a situation where the user has some inherent capability to modify software and to make other small changes to his C^2 system. If so, the approach might be to give this user the authority and funds needed to develop small increments of improved capability, as well as to fix mistakes and make the minor modifications necessary to maintain compatible interfaces for his C^2 system with its neighbors. Such activity should be under the control of the user, in order to achieve quick responses and to assure that the priorities of the improvements undertaken are responsive to the user's needs as the commander perceives them. However, a designated provider needs to be coupled closely enough to this user activity to assure that any changes that are being made to a user's overall C^2 system for which the provider has major responsibility take account of user changes that are being made locally, and vice versa. This clearly requires a different working relationship between user and provider than exists under the traditional "turn-key" approach, in which the user states all his requirements to the provider "up front", and some years later the provider delivers the product to him that is intended to meet these requirements. Case 1 is thus a tailored form of EA.

Case 2, in contrast, illustrates the notion that in addition to an on-site EA effort managed largely by a user (Case 1), and a major EA acquisition effort managed largely by a provider with heavy user involvement (Case 3), an intermediate means of providing evolutionary improvements to C^2 systems is available. This way (Case 2) involves a strategy for moving things quickly from development laboratories to the field. Capability obtained in this way will still tend to be relatively moderate in scope, because it ordinarily results either from periodic products of a long-term development program or is a deliberate, short-lead-time development, accomplished in response to a specific user problem. Case 2 is another tailored form of EA.

This second representative variation of the EA strategy requires a means to get a user directly involved with the laboratory, in order to determine that the product or products will be acceptable as they periodically come from the laboratory (or can be made acceptable with reasonable adjustments). Also required is a means to carry the results of the mutual effort directly into later program stages quickly, rather than viewing each increment as separate new developments, with each started from scratch.

Case 3, at first glance, might look like a traditional major system acquisition, rather than a type of EA. But in contrast, for example, to the Airborne Launch Control Center for the M-X missile (which may be able to be sufficiently well-defined so that it can be acquired in a traditional manner), Case 3 is typified by the way the new Space Defense Operations Center, also a major acquisition, is being handled. SPADOC acquisition is being conducted in an evolutionary way, because the system is both: (1) linked closely to other systems that are changing at this time and (2) has requirements that are not precisely definable as yet. Therefore, an evolutionary approach, with incremental delivery of blocks of the system, has been chosen as the acquisition strategy. Two contractor teams have been selected to compete during the design phase. At the end of this phase, which will last one year, a single design will be selected and the winning team will be put on contract to produce Block A of the SPADOC capability and to design Block B. Subsequently, a contract will be written for the production of Block B and the design of Block C, etc. So long as things go well, the plan is to continue with the same contractor; but there will be no long-term contract to this contractor to satisfy a requirement for the final overall system desired.

Finally, regarding Figure III-6, it should be noted that all three cases illustrated may occur at different times in the development of a given C^2 capability, as it is continuously upgraded.

c. Policy Aspects

Since EA is really a whole spectrum of possible approaches, the Study Team recommends that strong policy stress be placed on the need for creative tailoring when EA is adopted as an acquisition strategy. Otherwise, EA faces the danger of becoming just another acquisition fad which will inevitably get discarded when enough acquisition practitioners find that it does not fit their particular needs.

Use of EA in any form should not be forced when program circumstances do not warrant it, just because the system being acquired is a C^2 system. Circumstances where EA would not be appropriate are discussed in more detail later (Section D.1.e.). In these circumstances, policy should provide that the use of more traditional acquisition approaches, such as a one-step acquisition based on a design specification or a performance specification, or some combination of these traditional approaches and EA, may be used.

In this latter regard, the problem of defining an acceptable "core" capability, for example, may well call for a two-phased program in which there is first a combined user/provider activity, aimed at gaining enough information to proceed with a specific "core" effort at an acceptable risk level. The program might then become an EA effort for subsequent increments.

The point is that program circumstances can dictate the choice of an acquisition strategy, just as much as the type of system. Both must therefore be carefully considered, and the approach chosen tailored to the specific case. The basic policy mandate of DoD Directive 5000.1 that "The acquisition strategy developed for each major system acquisition shall consider the unique circumstances of individual programs" hold as much for "command and control" systems as it does for any other kind of system. As developed in the next section (D) of this chapter, the problem is that participants in the process generally oppose <u>any</u> deviation from the traditional approach.

D. IMPEDIMENTS TO THE APPLICATION OF EA

Thusfar, the discussion in Chapter III has dealt with a series of problems the Study Team found in exploring DoD's current approach to the acquisition of C^2 systems and the promise a new technique known as evolutionary acquisition (EA) has for resolving these problems. But the EA technique alone is not sufficient. For EA to be practiced adequately requires also that a number of important changes be made in the way the policy, requirements determination and validation, PPBS, and acquisition support communities go about their business. This section describes these needed changes and shows how they will act as impediments to EA if not brought about and adapted to EA when this is the strategy to be used on a program.

1. Status of Policy and its Implementation

As previously indicated, Section 13 of DoDI 5000.2 of 3/19/80 (Appendix D) was the first official DoD policy specifically on the subject of acquiring command and control systems. The Study Team reviewed this policy statement for validity in the face of its findings, as well as to determine the status of its implementation and understanding of its meaning within DoD. As a result of this review, the Team believes that Section 13 is basically sound and needed policy for acquiring these unique systems. However, the Team has also concluded that a serious impediment to the application of EA is the fact that the policy needs important modification and much more widespread dissemination and implementation throughout DoD than has occurred.

a. Adequacy of the Current Policy (March 1980)

Specifically, the Study Team reaffirmed the desirability of there being a special acquisition policy section for "Command and Control Systems" in a basic DoD-wide acquisition policy document like 5000.2. The Team also validated the primary thrusts of the current version of the policy (Section 13 of March 1980 5000.2) as regards:

- The evolutionary manner in which C^2 system requirements are best determined,
- The appropriateness of EA for C^2 system acquisition, and
- The important role some user must play in such acquisition, using a flexible "test bed" of some type as an instrument for accomplishing such role.

However, the Study Team considers that Section 13, as written in the March 1980 revision, needs to be modified in at least the following major respects (or new, separate policy should be written).

1) <u>Applicability of EA Should be More Precisely Defined</u>

Although an evolutionary approach to the acquisition of <u>any</u> individual C³I system program may turn out to be desirable, especially when the system's software content is to be high, Section 13, and the criteria for the type of system to which it applies, should be policy only for systems which are true command and control systems, i.e., systems which are primarily decision-aids aimed at enhancing the ability of some commander and/or his staff to perform the functions of commanding and controlling. The potential breadth of the application of the policy across C³I systems, as the policy is now written, is believed to be one of the causes of the failure of the policy to be accepted more widely. Simply, its criteria seem to encompass too many programs. (See discussion in Section C.3 of this chapter regarding these criteria and the types of C² systems most suitable for EA--p. III-38).

2) <u>Tailoring of EA Strategies Should be Encouraged</u>

The policy needs to be written in a fashion that assures that it is not interpreted too rigidly, i.e., that its tenets, especially the concept of evolutionary acquisition, are tailored to each program. (See the prior discussion (Section C.4.a of this chapter, p III-46) of EA being a <u>spectrum</u> of possible approaches.)

3) <u>Roles of User, Provider and Tester Better Defined</u>

Finally, the policy needs revision to reflect the intended and potentially shifting role of the provider vis-a-vis the user in a C² system acquisition, both from program to program and throughout the life cycle of an acquisition. (See the discussion in Chapter IV of this report for the different-from-normal and iterative roles of the user, provider, and tester in the acquisition of C² systems.)

These and other critical modifications are included in the revision of DoDI 5000.2 C² system acquisition policy proposed by the Study Team as a result of its findings (see Appendix A), as well as a number of other significant recommendations for improvement in the policy.

b. Adequacy of the 12 April 1982 OSD Rewrite

As this report was in preparation, OSD was revising all of DoDI 5000.2, including the section on command and control system acquisition. The present draft rewrite of this section (as of 12 April 1982) is included herein as Appendix C. THE STUDY TEAM CANNOT ENDORSE THIS VERSION OF THE POLICY. It is a major step backwards from the March 1980 version, let alone what should be in the present revision as a result of the extensive effort on the topic the Team has made on DoD's behalf, reflected in this report. The 12 April draft is inconsistent with several major conclusions and recommendations of the study, not the least of which are:

- The 12 April version does not require the use of evolutionary acquisition as the primary strategy for acquiring decision-aiding C² systems, nor even encourage it in appropriate circumstances,
- The 12 April version significantly diminishes the influence of the "real" (mission) user in C² system acquisition, and
- The 12 April version fails to reflect the need for DoD-wide adoption of a layered architectural model to facilitate the establishment of interconnect and protocol standards.

If Appendix A cannot be adopted by DoD for use in the update of DoDI 5000.2, the Study Team urges that, as a minimum, the changes listed in Appendix C be incorporated in the 12 April version. It is imperative that DoD have separate acquisition policy for C² systems. These changes are crucial to obtaining improvements in C² system acquisition.

c. Status of Policy Implementation

An even greater impediment to the successful application of EA is that the Section 13 policy (DoDI 5000.2 3/80) is not generally known or being applied in practice to any significant extent, especially as regards the full intent and significance of its evolutionary approach to acquisition and related "special management procedures." Rather, as previously indicated, the Army, Navy, and Air Force were all found to have merely adopted some of the tenets of Section 13 on a more-or-less <u>ad hoc</u> case-by-case basis, usually <u>after</u> a negative experience trying to acquire a particular C^2 capability in the traditional way. The full significance of the "lessons learned" from these negative experiences are not as yet being treated as a matter of official DoD Component policy as regards when and how to apply EA to C^2 system acquisition. Not only have the various DoD Components not issued adequate implementing versions of 5000.2, but as the current rewrite effort on 5000.2 illustrates, there is still important resistance to the existence of the policy at all, especially as regards mandating EA as the primary acquisition strategy for C^2 systems of the type on which this report focuses.

d. <u>Recommended Actions</u>

As a result, the Study Team recommends the following actions.

1) <u>Issue 5000.2 Verbatim at the DoD Component Level</u>

After revision to reflect the Study Team's recommended improvements (Appendix A), the C^2 system acquisition policy section of DoDI 5000.2 should be issued verbatim (or separate policy issued if an adequate 5000.2 treatment cannot be obtained) and without embellishment by the various DoD Components.

2) <u>Mandate Use of EA</u>

The policy should include a covering statement which places the burden-of-proof on those proposing to initiate C^2 system programs (or increments) to show, in their initial acquisition strategy document for the program, why (program circumstances) the tenets of the section are not being followed on the program when this is to be the case.

3) Issue Explanatory Guidance

A DoD Component task force, chaired by OSD, should be established to draft an explanatory guidance document in support of the section which:

- Elaborates and narrows the description of the types of programs to which the section is intended to apply,
- Expands on the meaning and intended range of choice of each of the tenets of the section, and
- Describes some of the circumstances under which EA might not apply (see Section D.1.e below).

This guidance document should not be made directive in nature until DoD has had at least three years of experience under it and has used this period to improve the document, and its underlying policy, from a field point of view (such as lessons learned on actual programs), i.e., the policy, too, should evolve.

4) Establish an Educational Program

A campaign should be mounted, with the help of the DoD educational community, to indoctrinate those responsible for acquiring C^2 systems in how the EA process works and in the lessons learned by others in applying it. The system acquisition curricula at the various Service and joint schools should be modified to include instruction on EA.

e. Exceptions to the Use of EA

Examples of types of program circumstances in which DoD may choose not to take an evolutionary approach--at least not a full one--even in the case of true C^2 systems, are as follows:

- DoD may know exactly what it wants overall in the particular case, and the path to obtaining it is easily defined,

- Conversely, program uncertainty may be so high in terms of either results or value that DoD cannot even specify an acceptable first effort (or "core") for the program and so first proceeds with one or more exploratory phases on the program (which may or may not end up in an EA approach being taken at some point), or

- DoD may be willing to take higher than ordinary risks on the program and either try to achieve what is wanted directly and fully, or reach relatively far out into the technology, in order to satisfy some urgent need, to save costs, or to reduce lead-time by eliminating certain steps or taking other short-cuts.

2. The Current "Requirements Process"

The operational requirements process is almost in a class by itself, as regards the importance of adapting it to EA. But the C^2 system operational requirements process, as it is now practiced, both in its requirements determination and its requirements validation stages, is a major impediment to successful EA.

a. Requirements Determination

The operational requirements determination part of the current requirements process acts as an impediment in two basic ways:

- It fails to recognize the fact that the difficulty of stating and quantifying the requirements for C^2 systems, and keeping up with the rapid changes in requirements that are called for, is among the foremost reasons for the EA approach to have to be taken in the acquisition of C^2 systems in the first place.

- Given this necessarily-fluid nature of the requirements definition process under EA, the process can act as a serious impediment to EA if it is too formal and lengthy.

EA is an acquisition technique which recognizes that for C^2 systems it is much harder, if not at times impossible, to follow the traditional acquisition approach. In a traditional method, a user first states what he wants in a requirements document. The requirement is then subjected to validation review by a higher DoD Component headquarters, and then submitted to a provider to obtain the required capability. Finally, independent testing is performed to determine if the user is getting what he asked for some years earlier.

EA, in contrast, accepts the notion that the only way requirements for C^2 systems can be determined adequately is for a highly-iterative relationship between user and provider to be developed in which requirements are determined, essentially, after-the-fact, by a process in which the user keeps trying out actual new capabilities in his own environment in digestible pieces ("build-a-little, test-a-little") and regularly provides feedback on them (in contrast to merely providing a piece of paper, however carefully thought out) until in fact he and the provider collectively judge that a useful and assimilable upgrade in his capability to perform his C^2 functions has been achieved. In other words, as the current policy (Section 13 of DoDI 5000.2 of March 1980) recognizes, not only must the acquisition of a C^2 system be evolutionary, but the requirements calling for this acquisition also must evolve.

The traditional requirements determination process, at present, is both quite formal and quite lengthy -- much too formal and lengthy to gain a prime benefit of EA: fielding new capability rapidly and often.

There is no need for this time-consumption and formality under EA, because, under EA, the need for them goes away. That is, under EA, the user is a highly influential member of the acquisition team, not simply a <u>post facto</u> reactor to its results. As a result, the user also does not have to be as precise and careful in his initial statement of his needs under EA because:

- Under EA's incremental approach, the user is no longer trying to predict these needs as far ahead as he does for other materiel, and
- One of the very purposes of the EA process is to help the user achieve the needed specificity.

b. <u>Requirements Validation</u>

Higher headquarters validation of requirements for resource allocation purposes can be accomplished essentially the same way it is under the traditional approach to acquisition, except for three things:

- Headquarters must understand that the requirement it is validating is "representative" within the context of an overall architecture, with only the immediate piece of the program to be worked on (the "core" or individual subsequent blocks) being specified to the degree that approaches the traditional. Later near-in blocks presumably also can be approximated, as the "representative" program, and the functional characteristics it calls for, are updated.

- The cost-benefit analysis used to allocate resources among the competing claimants for the Component's budget in any one year must include a type of sensitivity analysis in the case of C^2 systems which allows both the benefits and the costs of the proposed C^2 capability to be stated in a broader range than ordinarily would be the case for another type of program.

- As with the requirement determination part of the process, the requirements validation effort for a C^2 system must be accelerated under EA or it will cause the requirements process to get out of synchronization with the lead-times involved in the over-lapping, short blocks of effort which make EA such a promising technique for fielding useful C^2 capability early and often.

Regarding the cost-benefit analysis, a C^2 program must be allowed to be placed on the Component's program priority list on the basis of a type of analysis which rejects it outright only if there is a high probability that it will fall below some designated cost-benefit threshold. This analysis also should help the program find its specific place on the list, in a comparison with other possible uses for the same resources, on the basis of its attaining any value in its likely range of possible cost-benefit analysis outcomes.

Given the difficult evaluation problem C^2 systems inherently have, whether being acquired in a traditional or any other way, this is not an unsound approach to validating them in general. EA actually reduces the risk of selecting unsuccessful or less worthy programs, by providing for feedback of actual field results far more frequently than in traditional acquisition, and because it proceeds in smaller increments.

The speedup of requirements validation can be accomplished in a variety of ways:

- A faster determination of whether specific proposed C^2 capability upgrades are candidates for resources at all,
- Early release of those candidate capabilities that have been approved for work, up to some maximum level-of-effort, and
- Above all, not dealing with each block of effort as a new requirement to be validated, but rather as a "release" under a simplified and abbreviated procedure of part of the "representative" program, as long as it stays within designated performance and dollar thresholds.

c. Requirements Determination Support Facilities

The tools and facilities needed by users to help them play their proper role in a continuously-evolving requirements determination process are discussed in Chapter V.

3. "Business-As-Usual"

Simply providing policy at all DoD levels that requires EA to be the primary strategy for C^2 system acquisition is not enough. Even supplementing this policy with a guidance document and other educational devices to help assure thorough understanding throughout DoD of why EA is needed and how EA works (also discussed earlier) also will not be sufficient. Another major impediment to the successful application of EA is the tendency, when adopting the various tenets of EA on a program, to view them merely as minor perturbations on the current way of doing business (albeit in an incremental fashion).

Specifically, accepting the notion of having to acquire needed C^2 capability in useful increments or "blocks" is not enough, important as it is, if each such block is then viewed as a program in itself or is approached in the traditional way, as far as the functional activities which supplement the technical effort in an acquisition program are concerned. Indeed, there is every reason to believe that if each block of effort under the EA approach _is_ treated as a "stand-alone" individual program, subject to the normal "requirements process," PPBS, procurement, and program management lead-times and approaches, it is a virtual certainty that the overall C^2 capability needed by commanders to satisfy the demands of modern warfare will _never_ be achieved in any meaningful timeframe. In short, successful acquisition of C^2 systems _cannot_ occur on a "business-as-usual" basis, even when the EA strategy is applied.

Therefore, significant authority to be flexible and creative will have to be provided to most of the non-technical functions in a program if they are to be adapted adequately to the needs of EA. That is, Section 13 of 5000.2 calls not merely for EA, but broadly for "_special management procedures_" throughout in C^2 system acquisition. And this means something more than accomplishing "the design and testing of such systems...in most cases,..in an evolutionary manner." This part of the policy has gone relatively unnoticed to date, and yet, "special management procedures" are crucial to the success of EA.

Principal among the functions needing to be so tailored, in addition to the requirements process, are _budgeting_ (and related PPBS activities), _procurement_, and _program_ _management_. Each of these will be discussed in turn in this section.

a. _Budgeting/PPBS_

1) _Program Approval_

Once the "requirements process" has been streamlined, the next required step is acceleration of the program approval procedures for C^2 programs. In particular, emphasis is required on reducing the time between conceptual definition and program initiation because, under EA, total system definition results only from an iterative process which cannot be completed until after the user has regularly obtained hands-on experience. For this iterative process to begin, the "core" capability must be provided for test in the user environment relatively quickly.

Similarly, the cost estimates leading to program approval should emphasize the broad system capabilities desired, be viewed only as "best" estimates (or "ceilings"), and be judged primarily on the basis of affordability and worth considerations, as contrasted with attempting to define and cost in detail the total work breakdown structure of the program. This "design-to-approved-budget" approach requires that the user/provider team be able to trade-off "requirements" to keep the program within cost. Continuous user participation in the acquisition should assure better user understanding of the cost implications of his "requirements."

Finally, planning for and initiation of increments/ blocks must be handled in an overlapping fashion, or the full benefits of EA will not be realized. A typical program might see Increment #1 in operation (and configuration managed), Increment #2 being tested in the user environment, Increment #3 in development, Increment #4 in programming, and Increment #5 in planning.

2) Budget Approval

Normal PPBS procedures for establishing and gaining approval of budgets should be tailored to support the iterative process of "build-a-little, test-a-little" that characterizes EA. This will preclude the gaps in funding that would result from the normal two-to-three years lead time between budget formulation and funds availability, if each increment were treated as separate programs.

The Study Team discerned a distinct willingness on the part of DoD users to accept periodic useful increments of C^2 capability, rather than wait for the total satisfaction of their needs, if, in so doing, they could get something useful fielded as rapidly as possible. Signs of a favorable attitude towards establishing special procedures for budgeting, as well as acquisition, to get such periodic capability increases also were observed. However, each person with whom the notion of special procedures for budgeting was discussed thought that others, particularly at higher levels, would never allow it.

In this PPBS tailoring activity, budgets for C^2 capability needs should be approved initially within DoD on the basis of a "representative" overall program--an overall system concept and architecture--that is planned to be acquired in discrete blocks, with only the initial or "core" block being defined to the extent required in traditional acquisitions. This overall representative program plan is required under EA because budget lead times would introduce unacceptable delays in the program if each increment were subject to rejustification as a new start. Budget stability should be maintained if the program proceeds according to plan.

3) Congressional Approval

Procedures for assuring acceptance of the representative program plan in the Congressional budget approval cycle need to be devised. While the Study Team did not interview persons from the Hill, it appears that each submission to the Congress of the annual portion of the five year defense plan (FYDP) for a C^2 program will have to include a descriptive summary of the next block to be initiated that is supported to the same level of detail and firmness as a traditional acquisition. That is, for the purpose of Congressional and other higher-level determination of program performance, the program plan for each increment will have to provide sufficient visibility to assure that appropriated defense dollars are indeed providing meaningful increases in user C^2 capability sooner. One mechanism that could be developed to facilitate these higher-level reviews might be an annual report of satisfaction provided directly to them by the user.

This approach is similar to the currently-approved handling of a P^3I program, except that under EA, planned subsequent increments are defined as part of an ongoing effort, whereas under P^3I, subsequent increments presumably would be defined at the beginning of the program. With short-duration acquisitions and a predetermined budget for the total "representative" system, these incremental acquisitions will be defined and executed within approved budget ceilings, making EA largely a process for "designing-to-an-approved-budget."

Thus, only a modest modification of the traditional procedures for budget formulation and approval are required by EA, and while acquired capabilities may differ somewhat from their representative descriptions (which overall will be adjusted at least annually for Congressional review purposes), the budgeting and other PPBS adjustments needed under EA are hardly what some might pejoratively describe as "a license to hobby shop."

Ample past experience has demonstrated that traditional budgeting techniques, even though ordinarily based on detailed specifications, rarely produce initial budget estimates which match later actual costs. In contrast, since programs under EA are implemented within architectural frameworks designed specifically to accommodate change--thereby lowering the probability that subsequent increments will force extensive redesign of prior results--EA's largely "design-to-approved-budget" and adaptive requirements approach should provide a better estimate, in terms of being a match between initial estimates and eventual funds actually expended.

b. Procurement

1) Expedited Procurement Procedures

Expedited and abbreviated procurement procedures need to be devised that recognize the continuous and overlapping nature of a C^2 program under EA. Current procedures are too slow and too cumbersome to accommodate the rapid fielding of the "core" and subsequent increments for test and evaluation that are the essence of EA. In particular, procurement policy should emphasize that treating each such increment (or block) as a separate program should be avoided.

2) Procurement Personnel Should Have R&D Experience

While the Study Team emphasizes the importance of user involvement in C^2 system acquisitions, it also recognizes the importance of procurement personnel involved in EA having the R&D procurement experience that ordinarily is obtainable only from a developing agency. Lack of familiarity in buying R&D services, on the part of procurement personnel, can be a serious impediment to EA. R&D procurement sophistication is required to acquire intangibles like advanced software and related professional/development services that are inherent in a C^2 system acquisition. In addition, EA adds the difficult dimension of having to write contracts for, and measure the results of, what might be largely a level-of-effort, "design-to-approved-budget" activity. EA also requires advanced procurement planning of a type which must provide for much over-lapping of phases of activity, as a normal, not unusual, part of the effort. The Study Team found several cases where procurement people at using commands who were not experienced in procuring on such a basis (being more accustomed to procuring already-developed hardware and software) were responsible for running the procurement, and the procurement suffered.

3) Source Selection Criteria

Source selection criteria must be tailored to accommodate EA if the procurement process is not to impede gaining the benefits of EA by focusing on, or undesirably weighting, the wrong items, such as bid price to supply the "core." Under EA, the "core" may represent as little as 10%-20% of the total program acquisition cost. Therefore, under EA, more emphasis needs to be placed on such proposal items as:

- Understanding the operational commander's problem,
- Soundness of the technical approach being offered, including the provision of an architecture that facilitates growth and the introduction of subsequent blocks of effort with minimum redesign,

- Innovation in design approach, and
- Contractor's past performance <u>and</u> current capabilities.

Since EA is the antithesis of "total package" procurement, there is concern that contractors will be motivated to "buy-in" on the initial phase. This necessitates reorientation of source selection criteria under EA away from the more usual stress on awarding to the "low bidder." The motivation for a contractor to "buy-in" to win the initial increment occurs because, once "in" on an evolutionary acquisition, the very nature of the EA approach tends to promote the contractor's continuing incumbency. Thus the government needs to place much less weight on offerors' absolute bid prices or estimated costs in selecting a contractor to do the "core," and even subsequent capabilities. Rather, its focus here should be on determining the relative cost realism of the bids of the various offerors and to check their past history for cost overruns and high overheads. This increased requirement, under EA, to evaluate contractors' cost proposals for realism, places a greater obligation on the government both to: (1) improve its in-house ability to generate independent assessments of contractor proposals and (2) generate <u>realistic</u> program cost estimates for budgeting purposes. Experience shows that the government, especially the electronic acquisition commands, have too often created extremely low (optimistic) budgets (as compared to program scope and risk required by RFP's, specs and Statements of Work).

4) <u>Contract</u>

Regarding the contract itself, the Study Team favored a cost-reimbursable type of contract for EA in the ordinary case (or even a combined fixed-price/cost-reimbursable type of contract, with cost reimbursement covering difficult-to-specify levels of effort to support the required iterative interaction needed for evolution). Some on the Study Team found an award fee as providing both protection for the government and a strong incentive to the contractor.

One impediment to EA that the Team saw in the area of setting contract terms would be the failure to provide some flexibility regarding compliance with contract provisions. Given that the contractor's effort may not, in all cases, result in user satisfaction for reasons beyond the control of the contractor, establishing mutually acceptable criteria for determining when the contractor has met the obligations of the contract are required. There needs also to be a careful meeting of the minds on the definition of each increment under EA before its implementation. Finally, the fact that the EA approach could lead to a requirement for subsequent modification of an increment <u>after</u> operational experience with it, calls for a change in attitude and procedure for determining when and how successful contract performance is to be measured, if the full benefits of EA are to be gained.

5) <u>Maintaining a Competitive Atmosphere</u>

Finally, a major DoD policy of long-standing, most recently emphasized in so-called "Carlucci Initiative" #32 of 27 July 1981, requires providing for as much competition as possible in the acquisition process. However, a major problem of any incremental approach to acquisition, including P^3I as well as EA, is the inherent difficulty it presents of sustaining a competitive atmosphere in a high-technology program after the initial contract is let. Without specific effort on the part of DoD to establish a competitive atmosphere in incremental acquisitions, the successful contractor for the first increment has a substantial advantage in the competition for subsequent increments.

This situation is aggravated in the case of an evolutionary (EA) type of incremental approach to the acquisition of a C^2 system, because of three factors:

- Increments (or "blocks") of effort tend to be rather short in (time) length,

- Increments of EA effort deliberately are made to overlap, in order to allow feedback from user experience with early increments to be reflected in the implementation of subsequent blocks. Conversely, EA contemplates that the results of this user T&E will be fed back to earlier blocks, and

- Evolution of a given C^2 capability should never end.

Under EA, each increment is so short (the order of two-to-three years), that allowing for normal procurement lead-times of a year or more to hold direct competitions for each increment would totally defeat the EA goal of fielding useful increments of capability as rapidly as possible. As it is, even without such direct competitions, part of the end stage of each EA increment must be devoted to planning for the next and, to some degree, subsequent increments, in order to reduce time losses between increments as much as possible.

Because of planned (and necessary) increment overlap, it is quite difficult to treat each increment under EA as a clean, separable activity that can be accomplished readily, and hence competed for, by separate sources.

C^2 systems are "immortal," in the sense that they should have no FOC (Final Operational Capability) date. Because of this, C^2 system program acquisition efficiency dictates (even more than in the usual case of a high-technology program, which itself is great), that such programs, once contracted for, stay in the hands of the original source. In the "real world," the numbers of loose threads and incomplete internal feedback loops in the system engineering of such a continuously evolving program can be too great to do other than stay with the original source in all but extraordinary circumstances. EA recognizes and attempts to deal with this reality.

With so many things thus militating against providing for direct competition in increments of effort subsequent to the first in the EA of a C^2 system, policy must pointedly call for an attempt to provide for such competition whenever the likely fruits of competition in a particular case warrant the expense and effort required to obtain it in that case. Benefits such as enhanced contractor motivation to help DoD satisfy its goals for a program are not always derivable from direct competitions, of course. And conversely, contractors can be motivated in other ways as well (e.g., through indirect, or industry, competition and through taking into account performance on one program in the source selection for another, as discussed earlier). Therefore a policy mandate calling for such direct competition should be tempered by an opportunity for the program office to justify not arranging for competition, providing that this omission is justified overtly in acquisition strategy documents or advance procurement plans, for example. But where it is possible and worthwhile to conduct competition, the policy should require that a real effort be made to do so.

The Study Team offers for exploration as possible, if not always desirable, techniques for obtaining direct competition in subsequent increments of an evolutionary C^2 system acquisition, one or a combination of the following (there are undoubtedly others):

- Having different contractors perform different parts of the overall job under a single integrating/architectural contractor or Government unit, with the mix of the parts variable over time.

- Requiring that the system prime contractor compete and/or periodically re-compete various designated subsystem or equipment aspects of the program.

- Conducting competitions not for each increment but for every other, (or for intermittent) increments, on as forward a basis as possible (i.e., arranging for competition, say, for increment 4 or 5 as early as increment 2), in order to allow for the necessary user feedback and re-doing by the same source of those increments that are follow-on to each other.

- Having the DoD break out, and itself conduct, early competitions for those items that are to be acquired in multiples, either within a given C^2 system or as whole systems (e.g., where the system is to be essentially duplicated for like organizational units). In this latter regard, the breakout might be by geographical area (e.g., based on the needs of different theatres) or by differences in operational missions (e.g., the system as it is to be used in Washington, D.C., vs. its use in theatres).

- Multiple sourcing of the beginning of a program (e.g., "core" definition), carried as far as the benefits from the continuing competition outweighs its costs.

- Parallel efforts on the overall program based on different technologies or architectures that are focused on different timeframes in the future of the program.

Some of these techniques admittedly are more-or-less traditional, requiring only creative tailoring to adapt them to C² systems undergoing EA. But they, as well as the other items on the list, are included to make the point that direct competition can be obtained if a real effort is made to provide it in worthwhile cases.

Such competition can be facilitated basically by deliberately reducing an "incumbent" contractor's advantage in various ways. *For example, the software competition problem discussed earlier can* be reduced by having the higher-level application software work (the architecture, analysis, etc.) done in-house--as a number of Air Force elements were found to be doing to an increasing degree--to the point where the software specifications do permit a valid competition for the remainder of the work, in an area which ordinarily accounts for the bulk of the dollars in a C² program.

A significant caveat is that for the government to have hope of maintaining a competitive atmosphere under EA, it is mandatory that, *before* the "core" or an increment is placed on contract, the government must impose the flexible system and inter-system level architectures required by EA, the use of high-order-language programming, strong software quality assurance, and centralized configuration management (Chapter V). This investment will pay off by providing for the documentation packages needed to conduct competitions subsequent to the initial one, plus insuring a supportable package in its own right.

c. Program Management

The same "business-as-usual" attitude holds for program management considerations as well, and hence, also can be an impediment to the successful application of EA. Program offices generally are not organized, nor do they have the procedural flexibility, required to deal with the special needs of EA. In general, those directly responsible for acquiring C^2 systems have to take extraordinary actions to enable them to preserve their EA approach. To help avoid these extraordinary actions, the Study Team makes recommendations in two areas.

1) Management Structures

Management structure should be created, especially within C^2 program offices and among those responsible for C^2 program advocacy, that can cope with the more-or-less continuous flow of overlapping, and even concurrent, activity that can be expected in evolving C^2 systems programs, in contrast to the less-demanding structures required under the more-serial, traditional acquisition approach. EA implies a more continuous demand for analysis, design, engineering, test and support people, whereas in traditional acquisitions, program "front-end" is analysis and engineering "heavy" and the "back-end" is heavy in support and test personnel. A "combined" program office, including providers, users, and testers, has been an effective approach on some programs.

2) Program Manager Authority

C^2 program managers using EA should be given the general authority to shorten or revise procedures on their programs to obtain the benefits of EA, when this authority can be justified in their initial acquisition strategy; and they should be encouraged to seek such justification (presently this encouragement is essentially invisible). Program management functions in the procurement, management planning and control, and financial management areas in particular warrant such attempts at tailoring. Particular attention should be paid to any limitations on the ability of program managers to orchestrate such activities in matrix management organizations.

E. CONCLUSIONS AND RECOMMENDATIONS

1. Major Conclusions #1, 2 and 3

Three of the five major conclusions of this report are as follows (the other two are contained, one each, in the subsequent two chapters):

Major Conclusion #1 - THERE IS A MUCH HIGHER PROBABILITY THAT USEFUL COMMAND AND CONTROL CAPABILITY WILL BE FIELDED EARLIER IF AN EVOLUTIONARY APPROACH IS TAKEN TO ITS ACQUISITION.

All of the data gathered, including the case material, led to the conclusion that an evolutionary approach will provide measurably increased C^2 capability to a user, *fielded sooner* than if DoD waited for a "total" solution to the user's need. It also helps to assure more immediate and more nearly continuous user satisfaction with what he is getting, with less government risk of program failure and financial risk, and under architectures that more readily accommodate change and facilitate technology insertion.

Major Conclusion #2 - ALTHOUGH EVOLUTIONARY ACQUISITION IS POLICY FOR C^2 SYSTEMS, ITS APPLICATION IS SPOTTY AND IT IS NOT WELL DEFINED OR UNDERSTOOD.

Although the evolutionary approach to the acquisition of command and control systems has been required by policy "in most cases" since early 1980 (Section 13 of DoDI 5000.2), its overt application is still quite limited, continuing to consist more of pragmatic adaption of some of its tenets than an overall embracing of EA as a concept. This situation exists because the policy is being allowed to be interpreted as being permissive rather than mandatory. Also, EA has been inadequately defined, and there has been a failure to support the policy with helpful application guidelines and training. As a consequence, there is pervasive misunderstanding of what makes up the EA approach and how to apply it, much less an appreciation of its potential benefits.

Major Conclusion #3 - EVOLUTIONARY ACQUISITION WILL NOT WORK IF AN ATTITUDE OF "BUSINESS-AS-USUAL" IS ALLOWED TO PREVAIL IN THOSE LIFE CYCLE ACTIVITIES WHICH SUPPORT SUCH ACQUISITION, AS WELL AS THOSE WHICH ARE DIRECTLY PART OF THE PROCESS.

The design and testing of C^2 systems in an evolutionary manner will not succeed unless, in addition, other system acquisition functions such as requirements determination and validation, budgeting and other PPBS activities, procurement, and program management, are adapted to the special needs of EA. In addition to calling for EA, the policy (Section 13 of DoDI 5000.2) calls for "special management procedures" in the acquisition of command and control systems. Unless strong and clear direction is given in this regard, normal bureaucratic inertia will cause these non-technical system acquisition functions to continue to be performed as they are for any other type of system program, i.e., for an attitude of "business-as-usual" to prevail.

2. Major Recommendation #1

These major conclusions led to the first of the three major recommendations of the report:

EVOLUTIONARY ACQUISITION (EA) SHOULD BE BOTH MANDATED AND FACILITATED AS THE PRIMARY COMMAND AND CONTROL SYSTEM ACQUISITION STRATEGY OF DoD, UNLESS OVERT JUSTIFICATION TO THE CONTRARY IS PRESENTED IN AN INDIVIDUAL CASE BASED ON PROGRAM CIRCUMSTANCES.

First and foremost, the Study Team urges that high-level action be taken (see proposed DUSDR&E action memorandum (Appendix B)) to assure that evolutionary acquisition is designated, by policy, to be the principal DoD C² system acquisition strategy <u>and</u> that any alternative acquisition strategy proposed in a particular case be treated as a deviation from the norm, requiring specific justification. While basic DoDD 5000.1 acquisition policy is flexible enough to encompass EA as a strategy when appropriate, the bureaucracy that implements that policy is conditioned, in general, to carry out a system acquisition in the so-called "traditional" way. OSD, therefore, has to take a very strong policy action if it wishes to turn that bureaucracy around in the case of command and control systems. THIS REQUIRES THAT SEPARATE AND UNIQUE ACQUISITION POLICY BE ISSUED FOR C² SYSTEMS.

3. Sub-recommendations

Specific sub-recommendations in support of this major recommendation are as follows:

1.1 REQUIRE JUSTIFICATION IF ANOTHER ACQUISITION STRATEGY BESIDES EVOLUTIONARY (EA) IS PLANNED TO BE USED

For the vast majority of C² systems, the burden-of-proof should be on those proposing such deviation from the EA norm to show why an alternative strategy is warranted in the particular case, not vice versa. THE STUDY TEAM MAKES THIS RECOMMENDATION BECAUSE WE FOUND NO EVIDENCE OF SUCCESSFUL C² SYSTEMS ACQUIRED IN THE TRADITIONAL WAY.

This recommendation "flips" the policy 180 degrees, and may be noxious to some. For example, some may say: "The basic A-109/5000.1 policy is flexible enough to allow for an evolutionary approach, so we don't need a special policy for C² systems." In response the Study Team's point is that even though stated policy is flexible, the bureaucracy that implements that policy, in general, only knows how to acquire systems one way--the traditional way. DoD <u>must</u> turn that bureaucracy around! The only way to do so is by a very strong action to establish and ensure implementation of the new policy.

1.2 TAKE DIRECT STEPS TO FACILITATE THE USE OF EA BY MODIFYING OVERALL DoD REQUIREMENTS, BUDGETING, PROCUREMENT, AND PROGRAM MANAGEMENT POLICIES AND PRACTICES TO GIVE THEM THE FLEXIBILITY NEEDED TO ACCOMMODATE EA

All the evidence examined by the Study Team indicated that evolutionary acquisition (EA) can contribute significantly to improving the C^2 system acquisition process. But if EA is practiced on a "business-as-usual" basis in the requirements, programming/budgeting, procurement and management processes, EA cannot realize this potential. Therefore, the "special management procedures" tenet of DoDI 5000.2 should be extended to include the PPBS cycle and requirements activities, especially as regards specifying and budgeting for C^2 capability needs on the basis of a "representative" overall program that is planned to be acquired in sequential but overlapping increments, rather than on the basis of a fully-specified program that purports to satisfy a known, fixed requirement (be it incremental or not). In addition, C^2 program managers should be given much more flexible authority on their programs in the procurement, management planning and control, and testing areas, and should be encouraged to use this authority to achieve the benefits of EA.

1.3 DIRECT THAT PERTINENT DoD COMPONENT POLICIES AND PRACTICES BE ISSUED OR MODIFIED ACCORDINGLY

DoD Components normally issue regulations implementing OSD policy. However, at présent, implementing regulations pertinent to EA either do not exist or they do not conform to what is set forth on the topic in the March 1980 5000.2. This matter should be rectified promptly, if only by having OSD policy re-issued, <u>as is</u>, as Component policy. DoD Component <u>practices</u> with regard to the acquisition of C^2 capability should be reviewed and modified where found not to be in compliance.

1.4 DEVELOP GUIDELINES TO FLESH OUT THE DoDI 5000.2 POLICY ON C² SYSTEM ACQUISITION

In addition to the foregoing, an USD-led, intra-DoD task force should be formed to develop a guidance document that explains DoDI 5000.2 C² system acquisition policy. This document should be explanatory in nature--a roadmap that gives program managers and HQ staffs a better idea of when and how best to apply EA under various circumstances. The guide should be kept informal and itself evolve over time on the basis of actual experience with the application of EA. AFCEA is willing to help develop such a guide.

1.5 EDUCATE ALL OF THE PARTICIPANTS IN C² SYSTEM ACQUISITION ACTIVITIES IN THE TENETS OF EA

The last sub-recommendation is for DoD to initiate a major *ad hoc* effort to educate all of the potential participants in the C² system acquisition process about evolutionary acquisition--including users, user surrogates, "providers," "ilities" people, testers, HQ staffs, non-technical participants in C² system acquisitions, and pertinent Congressional staffs. These participants should be educated in how EA is intended to work and how to adapt functional activities to EA. The various DoD schools, such as the Defense Systems Management College, the Industrial College of the Armed Forces, the Air Force Institute of Technology, the Armed Forces Staff College, and the schools at Maxwell, Carlisle Barracks, Newport, and Monterey, are probably the most appropriate places in which to carry out this educational process on a regular basis. In addition, dedicated teams could be established to brief persons who will not have the opportunity or the time to go to these schools before becoming involved with a C² system acquisition. Finally, symposia could be held on the topic and/or it could be briefed as part of selected other symposia. AFCEA would be glad to support this educational process.

4. Other Conclusions and Recommendations

The Study Team formulated some additional conclusions and recommendations dealing with the acquisition process in addition to the major ones listed above:

- Section 13 of DoDI 5000.2 of 3/19/80 (Appendix D) with its stress on both designing and testing C^2 systems "in an evolutionary manner" in most cases and the provision of other pertinent "special management procedures," is sound and needed policy for C^2 system acquisition. However, the stated policy requires a number of modifications. Appendix A is a proposed rewrite of the policy, based on the findings of the Study Team. Appendix C contains Study Team's comments on DoD's 12 April 1982 draft rewrite of the policy. AS NOTED ABOVE, THE STUDY TEAM IS CONVINCED OF THE NEED FOR SEPARATE, UNIQUE ACQUISITION POLICY FOR C^2 SYSTEMS.

- The six so-called "criteria" of Section 13 of DoDI 5000.2 are not really criteria for determining when to apply "special management procedures" such as EA to the acquisition of C^2 systems, but rather, as a group, describe certain general characteristics of C^2 systems that distinguish them from other military systems. The revised 5000.2 should focus on just those few characteristics of C^2 systems which call for the EA approach and related special management procedures, in contrast to more conventional acquisition approaches.

- The degree of appropriateness of EA as an acquisition strategy is a direct function of the degree to which some user-commander needs to be involved in a program as an acquirer as well as a user. This degree increases to the extent such user-commander (and/or his staff) is himself a central element of the system. Only those C^3I programs in which this involvement thus needs to be high, i.e., which are true "command and control" systems, should be required to satisfy Section 13 of DoDI 5000.2 in the ordinary case. In the case of all other types of system programs use of EA should be optional.

- The evolutionary approach to acquisition is not a single, specific approach, but a strategy encompassing a spectrum of possible approaches. C^2 system program managers should therefore be encouraged by policy to select an approach to EA in a particular case that fits the circumstances of their program.

- There is a natural tendency for increments after the first one to be "sole sourced" under EA. Therefore, C^2 system program managers using EA should be required to maintain as much competition as possible within their programs and to justify in their advance procurement plans the absence of competition when such is to be the case.

- The necessarily greater role of the "user" in the evolutionary acquisition of C^2 systems can cause a focus on organizational missions at the expense of overall military missions, when these two types of missions are not the same for a given user-commander. Because of the importance of the latter type of mission in the evaluation of C^2 systems, any guidance drawn up to help implement Section 13 of 5000.2 should stress this potential difference. It should also suggest methods that individual users, in attempting to satisfy the needs of their immediate commands, might use to assure that they do not thereby detract from the acquisition of the C^2 capability needed to interface with weapons, platforms, and other C^3I systems to perform some war-fighting or war-preventing mission.

CHAPTER IV

CHANGED RELATIONSHIPS AND ROLES UNDER EA

CHAPTER IV
CHANGED RELATIONSHIPS AND ROLES UNDER EA

A. OVERALL RELATIONSHIPS

1. Current Roles

A major part of this study was devoted to reviewing relationships between the various participants in the C^2 system acquisition process. In general, present relationships are formal and "arms-length." While the details vary between the Services, essentially similar relationships exist within the Services. The traditional acquisition process usually consists of four phases:

- Concept definition and validation
- Advanced development
- Full scale engineering development (FSED)
- Production and deployment

The roles and relationships of the various participants in the acquisition process vary over these phases. Another way to examine the roles and relationships is to consider a traditional program life cycle as consisting of six interrelated "phases" that occur within the four phases listed above. These are:

- Requirement definition
- Concept validation
- Full scale engineering development (FSED)
- Operational testing
- Production including training
- Post-deployment

Figure IV-1 depicts the relative involvement of the various participants in the acquisition process under notional traditional and evolutionary acquisition strategies using the six program "phases" defined above. Of course, as developed in Chapter III, the six "phases" are more integrated under evolutionary acquisition. Within the spectrum of possible EA strategies discussed in Chapter III, Figure IV-1 is based on user involvement illustrated for a "Case II" EA. Consistent with the approach taken in the remainder of this study, the user (that is, the real operational user of the C^2 system) is identified separately from the user surrogate (who is responsible for representing the full range of users) in Figure IV-1. The participants in the process are considered to be the user, the user surrogate, the developer, the supporter (these two dubbed the "provider" earlier), and the independent tester.

Not unlike other system types, a traditional C^2 system acquisition begins with a "requirements process," which is that activity which precedes the issuance of an authorization for a provider to spend money in acquiring a C^2 system. A Command and Control Requirement can be originated from the DoD (JCS and WWMCCS Council)--usually <u>strategic</u> requirements, or the Services themselves--usually tactical requirements. Once a requirement is defined, validated and approved, it is documented in one of many forms ranging from a Required Operational Capability (ROC) (Army, Marines), Operational Requirement (OR) (Navy), Statement of Need (SON) (USAF), to a simpler Letter Requirement (LR), and generally becomes the responsibility of a single (sometimes lead) Service to develop, test, and field the resulting C^2 system solution.

Although real users can prepare Requirements, typically the "requirements process" is dominated by user surrogates (e.g., OPNAV in the Navy, TRADOC in the Army, Hq TAC for the Tactical Air Forces). That representative user prepares the ROC (OR, SON) and staffs it through the necessary using commands and headquarters for approval until it finally is identified as a Service-approved ROC (OR, SON). During this process, the user surrogate coordinates with all agencies which will be involved,

Phase / Participants	TRADITIONAL ACQUISITION						SAMPLE EVOLUTIONARY ACQUISITION[1]					
	Rqmts. Def	Concept Valida-tion	FSED	Opera-tional Testing	Produc-tion & Training	Post Deploy-ment	Rqmts. Def	Concept Valida-tion	FSED	Opera-tional Testing	Produc-tion & Training	Post Deploy-ment
User	M	L	VL	L	L	H*	$\underline{H}$*	$\underline{H}$	$\underline{H}$	$\underline{H}$*[2]	$\underline{M}$	H*
User Surrogate	H*	M	L	L	M	M	H	M	$\underline{M}$	$\underline{M}$	M	M
Developer	L	H*	H*	M	H*	L	$\underline{M}$[4]	H*	H*	M	H*	$\underline{H}$
Supporter	N	L	M	L	H	H	$\underline{L}$[4]	$\underline{M}$	M	$\underline{M}$	H	H
Independent Tester	N	L	L	H*	M	L	N[4]	$\underline{M}$	$\underline{M}$	H*[3]	M	$\underline{M}$

N - NONE

L - LIGHT

M - MODERATE

H - HEAVY

* - DOMINANT

V - VERY

_ - DENOTES INCREASED INVOLVEMENT/INFLUENCE UNDER EA

1 = The "Phases" here cover, for example, a single increment of a Case 2 EA (see Figure III-6, page III-48).

2 = Operational Suitability T&E

3 = Operational Utility T&E

4 = The "L," "M" and "N" refers only to definition of Requirement for initial "core." Once core enters test, the developer, supporter and independent tester are heavily involved in subsequent Req't definition activity.

Figure IV-1. Relative Involvement and Influence of Participants in Traditional and Evolutionary Acquisition Strategies including the development agency that will be responsible for developing and producing the equipment.

The time period for a requirement to progress through the ROC (OR, SON) documentation and Service Headquarters validation/approval stages can range from one to six-plus years. There are various reasons for this time differential, such as:

- Type of requirement (short or long-term),
- Projected costs of program,
- Service or joint ROC, and
- Availability of funds ($).

Usually the Joint ROC takes the greatest length of time before being approved for development and eventual production/fielding, due to multi-Service coordination requirements.

After a requirement is validated, a development directive is sent to the assigned development activity. At this point, the user/user surrogate usually recedes into a monitoring role--perhaps being represented, usually at program office discretion--in RFP reviews and source selection teams, and sending representatives to System Requirements Reviews, System Design Reviews, Preliminary Design Reviews, Critical Design Reviews, and Configuration Audits. Rarely are real users represented at these reviews, and if they are, it is even more rare that the same user representative person shows up with any continuity. The provider (usually the developer) dominates the Concept Validation and FSED phase with generally formal and "arms-length" relationships with supporters, testers and users.

The supporter's role typically is light during the early phases of the acquisition process and becomes increasingly heavier, peaking during the production and post-deployment phases of a program.

Dominating the operational test phase, the independent tester usually is involved in the earlier phases only to the extent of monitoring the program, while planning and preparing for conduct of the operational test. The tester usually has a role during the production phase (and sometimes post deployment) to ensure that any necessary Follow-On Evaluation is conducted.

In the concept validation phase, the initial technical specifications for the equipment are defined. Usually feasibility models are produced and tested. It is in this phase that a program usually can experience the beginning of many delays in the development process. Some of these delays are attributable to unforeseen technical complexity in design, "test problems," and changes in the "requirement" as a result of recommendations either from the user or the provider. Regardless, any significant problem usually causes a review of the program with pertinent agencies, including the user or his representative, to determine the most appropriate action before proceeding to FSED.

After a program successfully completes concept validation, it moves into the FSED phase. FSED is the most complex technical stage in the process, because in this stage the final design criteria are determined and engineering development models of the equipment are produced and tested. As noted above, during FSED, the user role usually is comprised of sporadic monitoring of program progress (e.g., at PDR, CDR, PCA).

Following completion of the development models, both the user (usually a surrogate) and the independent tester begin to take on more dominant roles as the equipment progresses to Development Testing (DT) and Operational Testing (OT). Many delays, generally the same type as discussed under concept validation, occur in this phase. Any changes in this phase as to scope and design usually require coordination throughout the entire decision chain depicted earlier and usually have large time and dollar implications.

Summarizing, in the traditional acquisition process, prior to development, the user (really the user surrogate) is dominant only in the requirements phase. His role diminishes in the acquisition phases, where the provider (e.g., CECOM, NAVELEX, ESD) normally dominates. Interaction between the user (surrogate) and the provider usually becomes sporadic during development. Surrogate users normally attend periodic program reviews (usually at program office discretion). The real user seldom participates in the development activity. The developer's role wanes as the system progresses into test, where the independent tester (e.g., OTEA, OPTEVFOR, AFTEC) becomes the dominant force. The provider is dominant in the production phase and finally, only in the deployment phase is the ultimate user dominant. These transitions occur because of the sequential nature of the program activity in the traditional acquisition process.

2. Modified Roles Under Evolutionary Acquisition

As noted in Chapter I (p I-15), the essential elements of the process of "evolutionary acquisition of command and control systems" are the following:

- Developing each system within an architectural framework which can accommodate change,
- Expediting the approval to develop, including accepting a short need statement outlining desired functional characteristics,
- Defining and fielding quickly an initial "core" capability representing a useful increment in operational capability,
- Designing and engineering subsequent increments based on continuing and intense user involvement/feedback,
- Keeping the increments relatively small ("Build a little, test a little."),
- Remembering that the "requirements process" is continuous and interactive.

As explained in Chapter III, the sequential nature of the traditional acquisition process will no longer exist for C^2 systems procured under the evolutionary acquisition (EA) approach. That is, no attempt will be made to detail either the total requirement, <u>or</u> the total solution, in advance. Many of the activities that are performed sequentially under the traditional acquisition strategy will become more parallel under EA, because of the overlapping increments. The result is that the interactions between the user, provider and independent tester, which previously were spread in time, now will become much more compressed, to the point of requiring continuous interaction between all elements of the program acquisition team. This modification of the relationships between the key organizations should be beneficial because the user, the provider and the tester will have a better understanding of each other's problems (e.g., the user will understand the resource implications of his requirements better and should be more willing to trade).

As an example, an evolutionary program might have the following overlapping activities:

- The "core" capability of a C^2 system is operational, is configuration managed and is under the direct control of the real (or designated "lead") user in performing his daily mission.
- Concurrently, the first increment of additional system capability is in the System Design/Support Facility (SDSF) (see Chapter V) and undergoing joint user/provider/tester T&E to ensure that it provides a useful increment in the user's command and control capability.
- In addition, the real user, the surrogate and the provider may be defining the requirements for the second increment of capability and employing a Rapid Requirements Definition Capability (see Chapter V) or the SDSF for that purpose.
- The third additional increment could be in the budgeting phase.
- The fourth additional increment could be in the programming (POM) phase.
- The fifth and subsequent additional increments could be in the planning phase.

Thus, interaction between the real (or lead) user, the developer and the tester would be occurring daily rather than on the periodic basis that occurs today under the more "arm's length" relationships that exist under the traditional approach.

The degree of parallelism that will occur under EA is dependent upon the specific program and the user's needs. In general, there will be a shifting from a "dominant" organization, depending upon the phase of the program, to a "team" organization for evolutionary acquisition of C^2 systems, where the user, the developer, the logistician and the tester are all key members.

The remainder of this chapter discusses the changed role of the user, the developer and the tester under the evolutionary acquisition concept, as well as the role of testbeds in evolutionary C^2 system acquisition and the changed nature of integrated logistic support functions under EA.

Appendix I is a reprint from the Defense Systems Management College's Program Manager journal which depicts the organizational elements involved in the acquisition process.

B. THE SPECIAL ROLE OF THE USER UNDER EA

As indicated in Section A above, one of the major differences between EA and the traditional acquisition process occurs in the changing role of the "user" and his interaction with the provider.

But, for military C^2 systems, who is the user? The Study Team denoted two kinds of users, "real" and "surrogate":

- The real user of a command and control system is he, and only he, who actually uses that system to accomplish his operational mission in war or for operational purposes, such as warning and crisis management, short of war.

- All others are surrogate users -- representatives of the real user.

And because "command and control systems" are so inseparably linked with the minds of the personnel who will use them in accomplishing operational missions, the Study Team states that this real user is the user of primary concern. We are talking about those people, in that command or staff element, the mind or minds of whom will combine with, and be part of the command and control system as the system does its job in wartime.

For "one-of-a-kind systems," such as the Cheyenne Mountain Complex which serves NORAD, these real users are easy to identify. In this case they are the commander of NORAD and his staff, and those of subordinate echelons and activities. There is no great need to find a surrogate user in this situation.

But command and control systems frequently are "several-of-a-kind" or even "many-of-a-kind." The need thus arises for a surrogate who can authoritatively represent the full range of "real users." In the Army, this need is, for the most part, filled by TRADOC and its schools and centers. In the Navy, OPNAV (OP 094 for C^3 Systems) performs this role for fleet users. The Air Force's TAC in the past has done this for overseas tactical air forces.

Notwithstanding the diligence and insight with which any of these surrogate users strive to represent the real users in the process of evolutionary acquisition, they can never take the place of the actual user. This basic truth stems from the very nature of command and control systems and their interaction with the minds of those who must use them in war. It also stems from the iterative, trial-and-error, learn-as-we-go, nature of the process of evolutionary development.

Further, it stems from the proposition that in the serious business of war, the person who must be granted decisive influence as to whether a command and control system meets an operational need is the one who must use it to meet that operational need.

1. Evolution in the User's Environment

As noted in Chapters I and III, the process of C^2 system evolution in the environment of the real user thus could go something like this:

- Start with what the real user actually has, and with what he is actually doing or wants to do. With the real user, the materiel developer (provider) and the requirements establisher (surrogate user) working together, design the "core" capability.

- This "core capability" is delivered to an actual operational user (or selected "lead user")--one who will rely on this system to perform his operational mission in wartime, or facing the enemy in peacetime.

- The user uses this system in actual operations and/or in simulations which closely resemble actual operations and/or in T&E. While he does so, someone from the surrogate user, the provider and/or tester is there, observing and interacting.

- Changes are agreed to and are made or the next increment is decided on and built. (Recognize that these changes are for the most part software, although they could be hardware.)

The Study Team is convinced that this sort of interaction must go on in the real user's actual environment, with the people of the real user present and participating. If it goes on in the domain of the surrogate, the process of development and fielding will suffer, because:

- The real user will be less inclined to accept the product of someone who does not "face the enemy" or "have the wartime mission."

- The surrogate, no matter how he may try, cannot duplicate the minds, thought processes, and intangible requirements of the people who really will use the system.

- The real user automatically considers the multi-national/multi-service imperatives.

The surrogate user has an essential role in:

- Reconciling the varying views of the many real users (and resource constraints),

- Ensuring that one real user's views do not dominate unduly,

- Fitting the evolving system into a harmonious "web of systems" that fits together in the field, and

- Helping the Service planners and budgeteers define the funding and fielding program for all real users.

However, the Study Team firmly believes that each and every command and control system development should have one, and possibly more than one, "lead" real user, and that the decisive evolutionary processes should take place in the environment of these real users.

2. Roles of the Real User and the User Surrogate

It is the process of evolutionary development in the environment of the real user that materially alters the user's role in EA. In the traditional acquisition approach, the real user may, but usually does not, participate in the generation of requirements. Following that phase, he has minimal involvement in the development and test process until the equipment is ready for fielding.

For evolutionary acquisition to be successful, the real/lead user must continually interact with the user surrogate, the developer, the logistician, and the independent tester. As indicated in the previous section, the core capability normally would be delivered to an actual operational user. The user is then deeply involved in the testing and evaluation of this core capability to determine its operational utility.

In order to perform this role, the user must participate during the development phase and his operations personnel must be suitably trained in the operation of the system prior to the start of testing. The user participates in the testing in order to evaluate the operational utility of the C^2 system in performing his mission. Concurrently, the independent tester is involved to determine the operational suitability of the core equipment in the field. Subsequent to operational testing of the core, the user is responsible for defining the requirements for the next increment of capability. The user surrogate is responsible at this stage to ensure that the continuum of users are adequately represented in defining the next increment to be performed. As a result of this interactive process, the user now plays a significant role in defining the C^2 system necessary for him to perform his operational mission.

The user surrogate's role also changes in the evolutionary acquisition process, but not to the same degree as that of the user. Specifically, the surrogate must now play a more important role during the operational testing of the core capability, as well as of the subsequent increments. Since this testing is taking place in the user's environment with heavy user involvement, the surrogate is responsible for ensuring that all other potential real users are adequately represented and that the views of the real user involved in the testing do not unduly dominate the process. Other than this aspect, the user surrogate's role is relatively consistent with his role in the traditional acquisition process.

3. Selection of the Real (or Lead) User

The Study Team's view is that continuous real (or lead) user involvement is the key factor in successful C^2 system implementation. Consequently, the selection of a real user to participate in a specific program being acquired under EA is of great importance.

Just as commanders differ in the way in which they employ C^2 systems to perform their mission, so do they differ in their desire to participate in the development and test of the C^2 system. A "receptive user" is someone who:

- Has substantial knowledge of the task that the C^2 system implements;
- Has intellectual drive and curiosity;
- Will take the initiative in testing the "core" and participating strongly in definition of (evolution to) subsequent increments;
- Enjoys being an innovator; and
- Wants to participate on this particular C^2 program.

The SIGMA program found such a receptive user in the VII Corps in Europe. That command is now participating in the development and test of the SIGMA/Maneuver Control System.

Obviously, assignment of the lead user role should be made to a receptive user. If a user is reluctant to participate on a C^2 program, the question must be raised as to whether that C^2 system has been sufficiently presented within the user community to convince the field commanders that it will help them perform their mission better.

The question might arise "How can a 'reluctant' user be motivated to participate"? The Study Team offers two suggestions:

- Convince him the "core" will help him do his C^2 job better--since the user will play a major role in defining the core, this should be feasible.
- Make the provider (developer and supporter) interact closely with the user at all levels (commander-on-down), so the user gains an early adequate understanding of what capabilities the "core" (and later increments) will provide and what it <u>won't</u> provide.

4. The User Needs Resources to Facilitate Evolution

However, these real users need help. Without some modest, but essential, resources they will not be able to perform their part in the necessary continuous and intimate user-provider-tester interaction.

First, they must be reasonably well informed as to what the technology is all about. They must know something about computers and how they function and what they can and cannot do. This calls for a degree of education, especially among more senior people who have not grown up with the computer.

Second, there must be, as part of the user's own establishment, a small and technically-well-qualified group which understands the user's situation and can represent this situation to the provider establishment in language both user and provider can understand. This small group must combine the practical-minded mission orientation of the user with the "intellectual drive and curiosity" which Professor Keen[1/] cites as essential. The group's basic role is to serve the user intelligently and skillfully as the user seeks to make better use of technology in performing his operational job.

Third, there must be, right alongside the user in his actual place of work, a small team from the surrogate user, provider, and tester establishments--in close touch with the user and his people on the scene and responsive to them as the "core" is put into place and exercised.

1/ Peter G. W. Keen and Thomas A. Gambino, "Building a Decision Support System: The Mythical Man-month Revisited," MIT, Sloan WP No. 1132-80, MIT CISR No. 57, May 1980.

Fourth, there should be some kind of special user capability for T&E. In some cases the capability is the actual command and staff element, into which the "core" capability is actually installed for operational use. In other cases, there may be a separate, off-line, prototype facility where experimentation can take place until such time as the user is satisfied that the mock-up version can be installed for actual operational use. The nature of this capability will vary case-by-case.

Fifth, there should be funds provided both to the user and the provider on the scene. The funds need not be very large, but should be sufficient to permit local software and hardware experimentation, closely coupled with a central configuration management facility.

Sixth, for those systems for which normal peacetime training and exercises cannot represent the conditions under which the systems will have to function in war (e.g., computer-based systems for assisting intelligence staffs), a capability must be provided for realistically simulating the expected actual information flow and other conditions of wartime use. These simulations may be fairly costly and rather demanding of technical manpower, but they are absolutely essential if the user is to play his role effectively.

Given the above resource support, the real user can, in his normal schedule of exercises and other training, actually use the "core" capability (and subsequent increments) as he would use it in war, and he can play his proper role in the user-provider dialog, through which the "core" capability will evolve incrementally, with both user and provider learning as they go.

Without resources along the lines of the above at the location of the real user, evolutionary acquisition of C^2 systems will not work well, and may not work at all. Since the Study Team believes that evolutionary acquisition is required for the successful development and fielding of C^2 systems, it follows the Team also believes that these resources must be provided to the field users involved in the process. In answer to the question "What should be the source of these needed resources, given the user is already straining to meet his mission"?, the Study Team recommends they come from within the TOA (people, dollars) of Hq DARCOM, NAVMAT and AFSC.

C. MODIFIED ROLE OF THE DEVELOPER UNDER EA

Although the developer's (part of the "provider") role under EA will vary from program to program, generally he will continue to provide the bulk of the technical and program management expertise so important to the development of C^2 systems. However, because of the unique nature of evolutionary acquisition, it appears advantageous to modify the "traditional" roles of the developer for certain types of EA programs. Just as the user must become more "technology conscious," so must the developer become more "user conscious" in C^2 acquisition programs. Also the developer must maintain an architecture that can readily accommodate change, growth and insertion of new technology. While the developer must become more "user conscious," he must also temper the user's natural "short-term fix" ("I need it now") orientation with a longer-term view toward the potential benefits that could be offered by planning to accommodate future technology. The developer must be sufficiently flexible to tailor his role and participate depending upon the program needs. In order to illustrate a potential modification to the traditional role of the developer, consider a possible EA case.

In this case, the C^2 system is intended for worldwide mobile application. For the first phase in this evolutionary process, the user or developer will procure commercial (or available) hardware and software and add some unique application software. The system will be installed at one of the real user's sites (the "lead" user) and evaluated under operational conditions. The system will be modified under user direction until a first increment (or "core") of operational capability is satisfactory. In this application, the system initially is employed to quantify the user's requirements for the "core" capability. During this first step, the role of the developer will be to provide technical support to the user as requested by the user, and to become familiar with the system requirements being defined. It is most important that a cooperative spirit be established early, so that both user and developer can make useful contributions in follow-on increments.

For the second phase, a militarized version of the "core" capability will be developed and units procured for worldwide development. During this step the developer would be in control, and his traditional roles generally would be followed. However, one important additional responsibility at the start of this phase would be to ensure that a system architecture is imposed that is compatible with future evolution of the system. This definition process must be a close, cooperative effort between developer and user. Thus, for the architecture definition task, the traditional user/developer relationship must be modified to a more integrated and shared-responsibility team relationship.

Following the fielding of the "core" capability, the system would continue to evolve with substantial user influence. However, the developer would continue to play a major role in the evolution process, in providing technology and developed solutions for the user's evolving requirements. For example, suppose that, in this case, the "core" only provided an operational capability for higher echelons (i.e., Corps or higher) and now the user wanted to provide an operational capability at lower echelons. This might require a large hardware procurement which the developer is best suited to handle, because of his management and procurement expertise.

Thus, there are three general roles for the developer under EA: (1) providing acquisition expertise (e.g., procurement, legal, program management, budgeting), (Given this expertise, the developer must recognize the unique nature of EA and be flexible in assuming roles which may be different from his traditional role, but which have merit to a specific EA program), (2) providing the technical expertise to define system architectures which are compatible with EA, and (3) being responsible for advocacy and timing of new technology insertion.

Other changes in the role of the developer can be foreseen. Under the traditional acquisition approach, system "responsibility" transitions from the developing command to the logistics command (Air Force example) following deployment of the system to the field. At that transition, the logistics organization assumes responsibility for the management, spares procurement and additional system reprocurement where necessary. In the Air Force, this transition is from AFSC to AFLC, while in the Army it would be a handover from the R&D center to the readiness side within CECOM. With the adoption of evolutionary acquisition as the C^2 system acquisition strategy, the developer will maintain an important role as long as the system is operational. This will tend to "eternalize" the role of the developer (and of the supporter).

In addition, the developer's configuration management function will increase in importance. Of particular concern is the task of software configuration management, since multiple revision levels of the C^2 software probably will exist simultaneously. Besides the basic software package released with the deployed C^2 system, it is possible that each major using command may wish to make local application software changes to make the C^2 system more responsive to their particular mission. In addition, the (lead) user will be testing other functions to be implemented in the next system increment. Coupling these requirements with the need for software and software modifications as other interfacing systems (both multi-Service and multi-national) evolve, highlights the need for a strong centralized configuration management organization. See discussion on page V-32.

D. MODIFIED ROLE OF THE INDEPENDENT TESTER UNDER EA

The role of the Independent Tester in the evolutionary acquisition of a military system is different from his role in traditional weapon system acquisition. This difference arises because of the unique and enhanced role assumed by the user in evolving and evaluating a C^2 system. The user, in operating the system, is a critical part of the system under test; and while he is using the "core" (or "core" and subsequent increments) in his operational environment, he simultaneously evaluates the operational utility of the "core" while evolving and evaluating new operational concepts. Through this highly important and often extremely complex process, the user, in actual fact, establishes the system requirements. It is this situation, namely, the evolution and refinement of requirements during operational T&E, that helps to distinguish the evolutionary approach from the more classical weapon system acquisition process.

The testing that takes place during operation of the "core" (and increments) should be directed towards evaluating total system concepts, tactics, man-machine interfaces and other factors relating to the operational utility of the "core" (or increment under test) and as such, requires close coordination among the user, provider and independent tester. The "arms length" approach characteristic of traditional acquisitions must not be permitted if successful T&E and subsequent system acquisition is to result. Also, test in the user's environment should approximate (or simulate) the capabilities, interfaces, and stresses of wartime.

On the other hand, care must be taken not to compromise the independence of the tester, since the independent tester still has the mission of determining how well the developed system meets established quantitative performance and operational suitability[1/] requirements that do not deal with operational utility.

The independent tester also provides several valuable services to the user, such as:

- Determining whether the "core" (or later increment to be tested) is sufficiently reliable and maintainable to support operation in the user's field environment.
- Providing expertise to the user and provider in the areas of experimental design, data acquisition, and data analysis.
- Supporting the user/provider team as required during user test operations in the user environment.
- Conducting operational suitability testing and analyses in such areas as reliability and maintainability on suitable test models (not necessarily the "core").
- Assessing whether the selected architecture has the capability to accommodate growth, change, and insertion of new technology.

In summary, the real user must be responsible for operational utility testing against his mission, working closely with both the provider and the independent tester. Operational suitability testing, on the other hand, should continue to be the responsibility of the independent tester. (While there was some disagreement on the Study Team over whether the user should run or conduct tests, most came down on the side of user-led operational utility T&E).

1/ DoD Directive 5000.3 defines Operational Suitability as "The degree to which a system can be satisfactorily placed in field use, with consideration being given (to) availability, compatibility, transportability, interoperability, reliability, wartime usage rates, maintainability, safety, human factors, manpower supportability, logistic supportability, and training requirements."

E. T&E IN THE USER ENVIRONMENT

Test beds are particularly valuable when concepts, size, interface complexity, and usage doctrine give rise to uncertainty in the final design or where there is high probability that significant decisions will be required during the development process rather than preceding it. The test bed can be a mechanism for design evolution and constitutes a method of permitting the evaluation of techniques and equipment in controllable environments. It can provide a common ground for interchange between users and developers, striving to establish a system configuration designed to quantify a set of end objectives, which in turn, are also refined in the process.

Broadly, there are two generic classes of test beds, INTEGRATIVE and INVESTIGATIVE:

- INTEGRATIVE test beds usually involve a significant amount of interface design, both hardware and software. The system requirements are more-or-less known at the outset of the design effort. This class is heavily test oriented: Examples are the Marine Corps user-operated Tactical System Support Activity (MCTSSA), and the Navy sponsored Combat System Engineering Development System (CSEDS) test bed employed for the AEGIS-class ships.

- INVESTIGATIVE test beds emphasize design and development, with integration as an adjunct supporting various system configurations and/or reconfigurations. Overall requirements (at least as target goals) are known, but system specifications are not. This class of test bed leans more toward evaluation as opposed to test. Examples are the joint Army/Air Force BETA test bed and the Army Operated SIGMA test bed.

The complex nature of most C^2 systems dictate the need for developmental test bed facilities early in program life. Chapter V calls this a Rapid Requirements Definition Capability (RRDC). In concept, this primarily INVESTIGATIVE facility can be used to simulate a variety of capabilities easily and quickly so that "user" experience can be obtained and a system concept (or architecture) specified sufficiently well to develop a "core" capability for use in an operational environment by the real user. A second necessary capability (necessary to sustain the evolutionary growth of the core system) would be a System Design/Support Facility (SDSF) of the type described in Chapter V. The SDSF could be a natural evolution of the RRDC.

While these different forms of test beds could apply to all complex military systems (particularly those with significant software content), because of the intimate involvement with cognitive processes of the user, C^2 systems require a "core" systems capability that is the user's operational baseline from which he evolves his requirements and future operational capabilities--which might be logically viewed as a User's Operational Test Bed.

In the course of its discussions, the Study Team found disagreement over whether a "core" capability can (or should) be a testbed, because a user would not accept a "core" which cannot "go to war," and "testbed" connotes something which cannot "go to war." This is largely a semantic problem. The Study Team does not intend that users be provided something not useful in battle. The Study Team does, however, want the user to "use"--that is, gain experience with--the "core" (i.e., an evolvable core) for feedback to the provider, with whom he works closely.

1. The Use of the Operational Core as a User's Test Bed

Unique characteristics of C^2 systems, particularly those associated with the determination of system specifications, many times will dictate the employment of a "core" assembly of hardware and software under the Commander-User's direct control in his own operational environment. Lessons learned over the past several years, and confirmed by the AFCEA Study Team's case studies, indicate that a highly-effective way of acquiring improved C^2 capabilities is to provide the Commander-User with a operational core which he can use in his operating environment as a learning tool to define and evolve required operating capabilities. This operational core system can, under these circumstances, become the key to the trial and evaluation of hardware and software techniques, together with the operational tactics and supporting systems. These can, in turn, lead to a refined system description. This evolutionary process can lead to system requirements suitable for subsequent acquisition actions. The role of this operational core in the development of future system requirements and specifications is particularly valuable, if not indispensable, in the development of those C^2 Systems which involve the direct interaction of the human commander. It is, in fact, critically necessary for users to operate this core as a test bed in their environment, so that they can validate the concept of operation, define specific operating requirements, and evolve required capabilities. This process is interactive and evolutionary in its applications.

2. Defining the Operational Core As User Test Bed

In defining the User Test Bed, one must first recognize that the user, in this context, is considered to be the "Real User"--that is, he is the field commander, as defined in Chapter I. Under certain circumstances, he may have to be a surrogate user, but in the final evolutionary stages, it is assumed that the test bed is operated in a real environment by the (or a selected "lead") actual battle commander-user. The "core" assembly of hardware and software, properly interfaced with the commander's communications, sensors, and weapon systems, is operated directly by the commander and his immediate staff. Generally, the operational "core" will consist of computers, memory (data base), displays and appropriate interface hardware and software. It normally would be programmed in accordance with an initial assessment of the commander's needs and his operational environment, and incorporate the flexibility of software modifications as operation in the field establishes the need for system change. While many approaches can be taken to define the operational core, the normative approach proposed by the Study Team is the RRDC described in Chapter V, (Section C, pp V-12-16).

3. Functions of the Operational Core

The operational "core" hardware and software capability should be usable immediately in the user's command environment on a regular, if not daily, basis. The operational core should provide the means whereby the commander and his staff can exercise his command resources to evaluate concepts incorporated into the core, interacting with the real world environment, operational plans and procedures, and his other systems (communications, sensors, weapons). While the operational core should be no more complex than necessary, it must be flexible and as such, readily modified as the user learns to use its capabilities and develop new ways in which to employ both the core and his other resources. Under some circumstances, the core can be used to learn how best to integrate new sensors, communications and weapon systems.

4. Responsibilities of the User

In order to insure a successful T&E in the user environment, the Commander-User must assume certain responsibilities. One of the first of these must be the development of a clear statement of his role and his needs. This statement can, in essence, serve as the starting set of requirements for the subsequent evolutionary process. At times, the user must also be prepared to take on the role of champion in advocating the core T&E program at Headquarters level. He must also be prepared to modify his first statement of need and requirements when budgeting and technology constraints dictate that this is necessary. ("It's the developer's problem" is not a satisfactory position, if the evolutionary process is to succeed in a world of fiscal and technical limitations).

As stated earlier, the Commander-User also has a responsibility to become educated technically to the level necessary to play his role competently. He must provide facilities and personnel needed to support the core and provider personnel. Most important, he must plan and then participate in tests and exercises of the core which can provide the type of feed-back needed to first test, and then modify the concepts programmed into the core. Finally, he must participate interactively with the provider, in the preparation of any system acquisition specifications or requirements which result from operation and evolution of the core.

5. Responsibilities of the Provider

The provider will, in general, be a development command or agency equipped to handle the acquisition of hardware, software, and services, including development efforts. A first responsibility of the provider must be that he acquires a sound understanding of the user's needs (to the extent these can be described), his mission and his operating environment. It should be the provider's responsibility also to provide the state-of-the-art technical knowledge needed for sensible trade-offs between a variety of hardware and software approaches, as well as between full Mil Spec hardware and suitably ruggedized commercial equipment for the core. Once procured, the provider should insure, by means of appropriate development testing, that the hardware and software are reliable without imposing lengthy formal test procedures on the program. Once deployed with the user, the provider must support the user with a qualified support team that can furnish training and maintenance support to the user while at the same time modifying the core in accordance with the evolutionary growth process. Finally, it should be the provider's responsibility (working with the user) to convert the system requirements as reflected by the evolutionary state of the hardware and software, to requirement statements and eventually, specifications suitable for system acquisition.

6. Compatibility

A universal characteristic of C^2 systems is that they must interact with a number of different, many times quite complex, systems that are already in place. In general, any new system must be able to accept a large number of widely different formats, protocols, and increasingly so, multiple computer interfaces. Communications tend to be a particularly challenging interface, varying from narrow-band HF to broad-band microwave, from land lines to satellites, and across Services and nations. A leading cause of difficulty for new C^2 systems is this variety. For successful "core" operation, the provider must address potential problems of interfacing with existing communications systems before the "core" is developed. Hardware and software deployed in the "core" should be designed with as much flexibility in its communications interfaces as is practical and should be such that this flexibility can be capitalized on by the field support team in a minimum of time. Likewise, careful attention must be paid to the interfacing of the information sources/sensors, taking only pertinent information when practical. Finally, the human interface with the Commander-User must be kept constantly in mind during the design of the controls and displays.

7. Operation of the Core in User T&E and Evolution

In general, the task of assembling the core hardware should involve little or no research and development, hence occupy a relatively short period of time. Concurrent with this process, the Commander-User should assemble a User T&E Action Team. A portion of this team should participate directly in defining the operational concepts and needs which determine the initial software package. Action Team members then should be prepared to operate the core immediately upon its being installed in (or near) the user command post. This Action Team should not delay its own involvement until the system is "tested and approved" by a separate testing agency, but should, in fact, participate in bringing the system on line. (No one will be able to define what is perfect or optimum at that time; that is what the user's Action Team will determine). Once the core is in operation and integrated with surrounding systems, it must be operated in the user's environment as a learning tool. This generally will mean exercising command and control functions during both routine operations and special operations which are directed specifically at evaluating concepts, tactics, and doctrines, as well as the performance of the hardware and software involved.

An important initial task will be the establishment of a factual baseline for the current concept. Once this has been done, continued exercising and evaluation can provide information as to how improved operational capabilities can be obtained through system evolution, changes in procedures, tactics, etc. At the time a decision is made to acquire additional increments and evolving the core based on T&E of the core (and possibly based on core implementation), the User T&E Action Team should work directly with the provider to prepare the requirements that in turn lead to the specifications of these systems.

8. Continuing Evolution and Sustained Concept Testing

The same characteristics of a C^2 system (e.g., changing environment, changing resources, differing command concepts and commander styles) that led initially to deployment of a core for T&E in the user environment, also dictate the continuing operation of the core and subsequent increments, for T&E even after a more "normal" acquisition process is initiated to procure additional systems like the core for other like users. This perhaps is the most difficult concept to sell to both the user and the provider communities. It takes the busy user's time and it "bothers" the provider because of its implication of "loose ends." Nevertheless, sustained concept testing is critical to the full success of the evolutionary acquisition process.

9. Summary of Operational Core Capabilities Required

In summary, the complexity of C^2 systems mandates the use of T&E and evolution in the user's environment. Developmental test bed facilities in which work of both INTEGRATIVE and INVESTIGATIVE nature can be accomplished are necessary. In addition to taking on an early form of Rapid Requirement Definition Capability (RRDC), a long-term System Design/Support Facility (SDSF) is required to sustain the evolutionary growth. Finally, and perhaps uniquely associated with the successful acquisition of C² systems, is the requirement for a flexible "core" system operated on a sustained basis by the real user in his own operational environment. The degree to which the last step is done will, in the end, determine the success of the C² system acquisition process.

F. CHANGED NATURE OF INTEGRATED LOGISTICS SUPPORT (ILS) FUNCTIONS UNDER EA

1. Current Situation and Implications for EA

In order to support the many systems they have deployed, each of the Services has evolved an integrated logistics support (ILS) system. The training and logistics commands who perform these functions are necessarily complex because of the variety of tasks they must perform, which include:

- Accomplishing planning
- Establishing stocks
- Providing maintenance and parts supply at the deployed unit, intermediate command, and depot levels
- Preparing documentation
- Training maintenance and operations personnel

These commands have been optimized to support large numbers of standard systems at the expense of flexibility and responsiveness to the non-routine and non-standard. Thus, for systems and equipment that involve new technology, low numbers, and a tendency to change significantly, as can be expected under EA, the traditional integrated logistics support systems of the Services are poorly postured to be responsive or effective.

Funding lead times for establishing an integrated logistics support base vary from three to five years. Further, establishing extensive and costly training, maintenance, and logistics service support resources for a "moving target" type of system baseline is more difficult. At the same time, it is essential that a new C² system be maintainable and supportable in potential wartime environments. Thus, ILS planning should be an essential part of each increment of an EA program.

Much of today's thinking on logistics support for field-deployed military systems is more in tune with commercial practices of the previous two decades than with today's commercial logistics support environment. Present practice of the larger commercial computer companies has evolved away from the earlier dependence on field maintenance resources, trending more toward dependence on self-diagnosing capabilities built into equipments, centralized remote diagnostics services for systems, and circuit board-level replacements (by low-level technicians), with defective boards returned to a central facility. Neither industry nor DoD can afford (nor subject itself to the force sustainability implications of) equipment that requires manpower-intensive technical support in the field. This basic economic (and strategic) fact of life has forced strong industry emphasis on reducing failure rates and shortening repair times. DoD should exploit the cost and resource savings available through more extensive adoption of these well-established commercial trends, including greater dependence on industry to provide for ILS of newer-technology systems. Examples of present industry ILS techniques that could be adopted for use on EA programs, and which could enable the Services to reduce investments in logistics, training, maintenance management and support resources, include:

- Limiting field-level technical support to board-level replacement
- Contracting with industry to provide training and maintenance support (facilitating replacement of defective boards, and appropriate stockage support, including wartime emergency supplies), at least for the "core"

a. A Model for ILS Under EA

Each of the Services has examples of innovative thinking and recognition of the need for changes in established ILS organizations, programs and methodologies to accommodate newer technology. A successful Air Force approach will be discussed here, as an example, but similar approaches can be found in various programs of the other Services.

On some programs (e.g., OASIS) the Air Force believes they are making considerable progress in addressing the issue of having "blue-suit" integrated logistics support, while ensuring the timely application of modern ILS approaches to fielded C^2 systems. The key lies in the role of the System Program Office (SPO), which is responsible for assuring an effective integrated logistics and training system for its program. The SPO, in effect, "contracts" with the Air Force Logistics Command (AFLC), the Air Training Command (ATC), and the using Commands to provide the most practical and cost-effective solution to the particular program's ILS needs. When the prime item involves extensive use of new technology (both hardware and software), SPOs have found that contractor support oftentimes is essential, even for tactical systems planned to be deployed to war zones (e.g., IBM Series I computers used by the USMC). Such integrated and cooperative planning and execution of ILS requires elimination of the "arms-length" relationships that classically have existed between developing and supporting activities, especially in the Air Force (where the relationship between AFSC and AFLC seems to be overly formal and "contractual").

Where hardware is non-standard, AFLC and the SPO develop a parts supply solution where the contractor provides an appropriate level of field-deployed "inviolate spares" for wartime emergency, in addition to supporting the normal operational and resupply system spares. The assigned depot would manage this contractor-furnished spares support to ensure that AF logistics requirements are satisfied and that the logistics system performs properly.

Regarding training, where the system and its operation are AF non-standard, the SPO negotiates with ATC for establishing the required training. The SPO and system user jointly develop the Training Plans Information document, which is the basis on which the appropriate Technical Training Center decides either to develop the training or to provide for contractor-supplied training.

Providing hardware and software configuration control is essential and critical to program success. Even if the user requires a responsive applications software modification capability, provisions must be made for centralized system and applications-level configuration control, preferably at a program office managed level. In an EA environment, the field-deployed system baseline must be kept compatible with the development baseline for subsequent increments.

b. Software Support

Typically, software will be the dominant life cycle cost factor on C^2 system programs. Although this subject is covered in Chapter V, several ILS aspects of software deserve comment here:

- Using contractor support for systems using current commercial technology (including software) could be more practical (in terms of cost, timeliness, and effectiveness) than attempting early development of in-Service support resources. This is particularly pertinent to facilitating timely deployment of the "core" capability.

- One potential risk in using commercial software is that the supplying third-party vendor may change or stop supporting the particular software used, as the technology represented by the software package becomes obsolete in the commercial marketplace. This risk can be reduced by using "mainstream" commercial products (which tend to be stable) and through early and continued coordination with the planning staffs of the supplier.

- Software evolution involves multiple systems and applications level software releases that must be tightly controlled and coordinated. This aspect of planning and managing the implementation of required new releases, particularly where both SPO and user-level software configuration could be involved, is the single most demanding software management issue. It also involves the greatest risks, if not properly controlled.

- The cost differential for overseas versus "stateside" software development is so high that contractor resources overseas must be kept to a minimum. One method that has worked is a small "trouble team" deployed overseas for problem identification and "urgent" fixes, backed by a larger PDSS (Post-Deployment Software Support) team "stateside" at the Permanent System Design/Support Facility (see Chapter V). SIGMA and ENSCE plan to use such an approach. Where applications software support for the user is needed, Service resources must be trained and provided for deployed systems. This approach also facilitates wartime supportability.

The Study Team found the AF OASIS project is a useful model for developing means to manage complex software development and maintenance programs. In this model, the SPO is responsible, in coordination with the user (USAFE) and other concerned AF commands, to assure that overall planning provides for system maintainability and supportability in wartime environments, where contractor support in the field is assumed to be unavailable. System redundancies, early training of key "blue-suit" personnel, OJT (on-the-job training) and self-paced training material, appropriate spares levels, modules-replacement-only maintenance policies at the field level, and provisions for contractor maintenance at depot levels, are all key elements of the overall OASIS ILS approach which the Study Team believes could have broader applicability.

2. Military Specification Considerations

Both the Army and Navy have been more constrained than the Air Force, relative to adoption of commercial technology for current C^2 systems, by their commitment to the extensive use of military-specification equipment for field Army and afloat Navy users. Particularly in those cases, insistence on using military-specification equipment that, prior to field deployment, is maintained and fully supported by a "green-suit" or "blue-suit" ILS system, is understandable, since such equipment must operate in high-threat environments. However, one could question if all field Army or afloat Navy C^2 systems should fall into this category. For Corps and echelons above corps (and equivalent Naval echelons), these Services should seriously consider wider use of commercial or ruggedized commercial technology for both hardware and software as a more cost-effective approach. This would permit the earlier deployment of more modern-technology C² systems. In addition, commercial ADP technology is getting more and more inherently rugged.

EA fielding of a core capability, and establishment of an evolutionary environment responsive to the user's needs and priorities, are software-intensive activities that are greatly facilitated by the use of commercial technology and its rich and prolific software base. Over 2/3 of embedded computer system acquisition costs are in software[1/], and this trend is increasing. Fortunately, more current commercial technology is becoming available to mil spec environments. (e.g., Rolm developing militarized Data General Novas and Eclipses, and Norden developing militarized DEC PDP-11's and VAX's).

A significant advantage in using mil spec versions of commercial technology in EA is the ability to use commercial, or ruggedized commercial versions for the core. Further, ruggedized versions of commercial systems are becoming fully acceptable solutions for needs that had previously been in the mil spec domain. IBM, DEC, and Honeywell commercial-type systems are in use or under development as standard systems for both ground tactical forces and ships at sea for the US and our allies.

The real risk, in the several programs where the Services are developing their own computers (instruction set architectures--ISAs) outside the commercial software mainstream, is the inherent inability of DoD to capitalize on the large investment industry makes in producing extensive and quickly-matured commercial software, which is viewed by many

---

1/ "DoD Digital Data Processing Study - A Ten Year Forecast," Electronics Industries Association, October 1980 as the major strength of the U.S. computer industry. Another important consideration arguing for extensive use of commercial technology is the rapid industrial "surge" flexibility commercial products provide, should national emergencies dictate rapid system expansions.

3. Role of Standards

Although system architecture and standards are discussed in Chapter V, it must be emphasized here that a well-defined C^2 system architecture and a well-selected and managed set of standards are essential to a viable logistics and training base, regardless of the acquisition strategy used. More specifically, cost-effective ILS requires a clearly-specified and supported C^2 system architecture that provides for network communications standardization, functional modularity, and both protocol and hardware standardization at system and component interconnected levels. In the ILS environment, the lack of such standards severely complicates the orderly and effective evolutionary replacement or addition of components and capabilities within large systems. Although each program office can orchestrate internal standards for its program, all major C^2 systems have extensive interfaces to other systems and programs. The current lack of interface standards between major programs can have major system development and ILS cost impacts.

One very important and relatively new opportunity where standards can be applied with favorable ILS impact is in local area system-component interconnects. Local Area Network standards, to facilitate system distribution at functional module levels, are essential to future systematic and cost-effective equipment module upgrades. The government presently has the flexibility (which will diminish with time) to influence standards in this rapidly-evolving arena.

Software standardization at the high-order language level (e.g., Ada) is important, not only for cost-effective life cycle software management purposes and to facilitate capture of commercial technology, but to permit standard training for C^2 system software development and maintenance personnel across multiple programs, Services, and users.

Further, regardless of the acquisition strategy used, DoD requires software standardization at the data element, graphic symbols, query language, forms management, distributed data base architecture, and high-order language levels, if the goals of establishing a viable logistics and training system are to be achieved. The lack of such standards assures that it will be difficult, if not impossible, to develop standard system components, such as graphic devices, personal workstations, or mass storage or data base sub-systems that can be centrally supported (with ILS) cost-effectively for use by multiple major defense programs.

WWMCCS and DODIIS are examples of DoD global systems architectures that articulate the need for standards -- although neither community has been as successful in establishing standards as the situation demands. WWMCCS presently is trying to evolve its software baseline to be closer to current and evolving commercial software.

Chapter V further discusses the architecture and standards issue.

4. Training Implications

Training already is a serious problem in C^2 system operation, regardless of the acquisition strategy chosen, due to the significant differences in how even similar equipment is utilized when deployed to different Commands and different theaters of operation. Further, the systems actually designed and deployed for different theaters often include both unique hardware and software. Evolutionary acquisition approaches that involve earlier fielding of newer technology could complicate this situation unless provisions are made for adequate training material and resources to deploy with the system. The centralized System Design/ Support Facility, discussed in Chapter V, could be useful as a training aid. Of course, the "core" system itself could be designed to incorporate some "built-in" training capability (e.g., use of commercial "self-help" features, tools and documentation, or a "user exercise simulation").

The long-term solution to consistent and cohesive training is establishment of common standards at the user interface level (data elements, graphics, symbols, forms, formats, inputs), plus comprehensive standards for communications protocols and device/systems-level physical interconnects. (See Chapter V.)

Commercial technology practices to cope with high costs of training include standardizing the user interfaces (extremely difficult in the required competitive atmosphere of DoD) and providing Computer-Aided Instruction (CAI) so new users receive self-paced training without the need for formal classes.

For many evolutionary acquisition programs, it will be most cost efficient to contract for training, including provisioning of training material and training of in-service trainers. But in the long term, it is the setting of appropriate user interface and system interconnect standards that will make the training problem more manageable. The following quote[1/] highlights the training impacts of standards within evolutionary development environments:

> "One benefit of the introduction of common components by DODIIS will be to ease the problem of operator and programmer training of intelligence personnel who move from one DODIIS site to another one, once a transition period is past. During the transition period, the situation should be no worse than it is now. Enhancements to common DODIIS software will present an ongoing training problem, but to the extent that DODIIS develops and sticks to standards for human interface design, the problem should be less than in today's environment. New functions will at least be consistent with old."

1/ Unpublished internal MITRE memo, dtd 06 March 80. Steve Lipner.

Finally, the provider needs to develop training materials from the program outset, and the thrust of these training materials and the training program should be to train persons in the using command to be able to train <u>other</u> persons in the using command. Training teams generally appear at using commands briefly and then depart. They need to leave behind material and capabilities to enable those who have been trained to train others. Training and training aids should include simulated wartime environments, so that the C^2 systems can be exercised. Insufficient attention has been given to this.

G. CONCLUSIONS AND RECOMMENDATIONS

1. Major Conclusion #4 (See Chapter III for Concl #1, 2, 3)

The major conclusion regarding the roles of the participants in the C^2 system acquisition process is:

Major Conclusion #4 - SUCCESSFUL EVOLUTIONARY ACQUISITION REQUIRES CONTINUOUS INTERACTION AMONG USERS, PROVIDERS, AND INDEPENDENT TESTERS AND A MORE INFLUENTIAL ROLE BY THE REAL USER.

The relatively serial relationship among the real user, the provider, and the independent tester that exists under the traditional acquisition approach inhibits the effective use of evolutionary acquisition. Even though the use of evolutionary acquisition for C^2 systems acquisition was made policy over two years ago, the classic relationship still remains. The provider is dominant during development. The independent tester dominates the test with few program exceptions. With some exceptions, there is little, if any, <u>continuous</u> participation by real users. This <u>must</u> be changed to a situation where the real user (or combination of lead user and user surrogate for multi-user systems) is dominant in the acquisition process of C^2 systems, with significant support from providers and independent testers. A "combined" program office, comprised of users, providers, and independent testers, is one promising approach.

The most significant problem, with respect to the roles of the participants in the C^2 systems acquisition process, is insufficient continuing real user participation throughout the acquisition process. In most programs reviewed that encountered problems, there was little real user participation and influence, especially in the initial definition of "core" capability and the feedback of user test data to the provider in near-real-time. The Study Team found a general attitude, especially in provider and user surrogate organizations, that periodic (e.g., at SDR, PDR, CDR) user or user surrogate participation is adequate to enable the acquisition of C^2 systems. It is the Study Team's strong view that the real user (or lead user plus surrogate) must be involved continuously with the process of evolving the C^2 system, its requirements, its design, its testing, resource allocation, etc.

2. *Major Recommendation #2* (See Chapter III for Recommendation #1)

The above major conclusion leads to the second major recommendation of the report:

> ALTER THE ROLES AND RELATIONSHIPS AMONG USERS, PROVIDERS AND TESTERS TO ENABLE CONTINUOUS INTERACTION, RATHER THAN THE MORE "ARMS LENGTH" APPROACH USED IN TRADITIONAL WEAPON SYSTEM ACQUISITION.

3. Sub-recommendations

Specific subrecommendations in support of the major recommendation are as follows:

a. Role of the User

2.1 INCREASE SUBSTANTIALLY THE REAL USER'S INVOLVEMENT IN AND INFLUENCE OVER THE ACQUISITION OF C^2 SYSTEMS THROUGHOUT THE ACQUISITION PROCESS.

1) Provide for Continuous User/Provider Interaction

The first subrecommendation is to increase substantially the real user's involvement and influence throughout the C^2 system acquisition process. For multi-user systems, a lead-user/user surrogate team should be defined to work interactively with the provider and independent tester.

The notion of a "combined" program office, where responsibility is shared by provider and user, appears appropriate for many applications. The Air Force has used this approach on the Cheyenne Mountain Complex Upgrade with success. The question of who (user or provider) should lead, program office location, and numbers and types of personnel, will vary from program to program. It should be remembered that continuous user/provider/tester interaction is necessary--not quarterly, not semi-annually, but daily--as members of an acquisition team.

2) Provide Resources to the User to Facilitate Participation

As discussed earlier and amplified in Chapter V, resources should be provided to facilitate participation by the user. Specifically, resources are needed to increase the real user's requirements analyses capabilities (people, tools, funds, and facilities) for developing mission needs and architectures and defining system capabilities. Tools should also be made available to the user to support this activity, aimed at breaking down the "cultural" or "language" barriers among user, provider, and tester. These tools will ease visualizing the implications of new technology and system design alternatives, and could include testbeds, a rapid prototyping capability (RRDC), and/or battle simulations. As amplified in Chapter V, rapid prototyping is especially appropriate for new developments and major enhancements. Here, the user should have early access to a means to mockup or simulate desired capabilities rapidly. This allows the user to develop a concept of operations for employing the system and to understand the potential operational impact. When the system architecture and initial core capability have been developed, the core should be integrated with this RRDC, and made available to the user, to serve as a point of departure to define changes and enhancements to be incorporated in future increments.

Providing resources to the user to facilitate his participation does not necessitate giving large RDT&E or procurement funding to the user. The proportion of funds should be tailored to program specifics. The user does, however, require resources to support his continual participation in evolving the C^2 system. One might ask: "Where would a DoD Component find the resources in a zero sum military personnel situation?" The Study Team's view is that the people ought to be provided by billets taken from Hq DARCOM, NAVMAT, and AFSC, with the people to be co-located with the user. These ought to be technical billets, with appropriate positive recognition made of people going into these billets, so that assuming such a role (especially for a joint or multi-national user) isn't a career detriment. Traditionally, the path to "flag" status requires filling certain career "squares". The Team wants to assure that people who assume these responsibilities don't get penalized (much as GEN Jones comments regarding his proposed changes to the OJCS).

3) <u>Real/Lead User Should Determine When "Core"/Increments Are Ready for Operational Utility Testing</u>

The real user (or a lead real user working with the user surrogate for multi-user programs) should be the determiner of when the "core" capability and subsequent increments are ready for <u>operational</u>

effectiveness test (i.e., T&E to determine contribution to operational mission). The independent tester or provider should not be allowed to inhibit the deployment of the system for user operational test, if the user wants to have the system deployed. Stated bluntly, since feedback from user T&E is the key to evolution, all blocks to getting the subject increment to the user must be eliminated! [Of course, if initial suitability T&E indicates failure to meet interoperability or supportability goals, the user must be aware that deployment likely will not result in an ability to determine contribution of the core (or increment) to mission performance.]

4) Real/Lead User Emphasis on Inter-Service/Multi-National Wartime Use

Lastly, it should be assured that real/lead user involvement includes early emphasis of the potential wartime inter-Service and multi-national employments of the C^2 capability being acquired. Potential joint and multi-national wartime employment of C^2 systems is receiving insufficient attention, especially for C^2 systems acquired in the traditional way. This problem is especially critical for theater and maritime C^2 systems at Corps and echelons above Corps (EAC) and equivalent echelons in the Navy and the Air Force. In the case of systems at Corps and subordinate echelons (CASE) and equivalent echelons in the Navy and Air Force, the interfaces that are required for these lower echelons to operate properly in wartime with top command coming from other Services or other nations also often are not being considered sufficiently. Nor, for that matter, is interoperability between systems developed by two Services which ought to work together in wartime, such as the ability of the MIFASS and TACFIRE computers to "talk" with each other without operator intervention in an RDJTF mission.

b. Role of the Independent Tester

2.2 CHANGE THE APPROACH TO C^2 SYSTEM TEST AND EVALUATION TO RECOGNIZE THAT T&E IS AN ESSENTIAL PART OF THE CONTINUING "REQUIREMENTS PROCESS."

1) C² System T&E Is Part of the Requirements Process and Must Be Interactive

The second major subrecommendation is to change the approach to T&E of C^2 systems to recognize that T&E is an essential part of the continuously-ongoing "requirements process". This does not rule out the independent tester. Rather the independent tester, the user and the provider must reduce the traditional "arms length," pass/fail relationship and become more like partners in the evolutionary C^2 system acquisition process.

2) T&E in the User Environment With Joint User/Provider Determination of Operational Utility

The real user should have a dominant role (be responsible for ?) operational utility T&E against his mission, working jointly with the independent tester. This is very different from traditional IT&E, where an "arms length" relationship exists between tester and provider, and some user personnel are part of the test team, but the independent tester clearly is "in charge" of the T&E. The operational utility T&E should be in the user environment. The test program should be designed to exercise the wartime role of the real/lead user, especially as regards inter-Service, multi-national interfaces and command structures.

3) <u>Tester Provides Expertise/Resources and Is Responsible for Suitability T&E</u>

In the case of the operational <u>utility</u> determination (that is, how well does this C^2 system help the commander and his staff perform the functions of command and control), it is the Team's view that the user is in the best position to make this determination, and the role of the independent tester, in this regard, is to provide the appropriate test resources. This includes helping design the test so that it represents an efficient utilization of test resources, in terms of people, dollars, and time. Of course, for multi-user systems, operational utility determination should be a joint lead user/user surrogate/tester process, but here again, we believe the real/lead user should be heeded more. <u>Suitability</u> T&E (reliability, maintainability, and other objective measures not relating to how well the system enhances the commander's ability to command and control) should be led by the independent tester, with user/user surrogate participation, as is done traditionally.

4) <u>People Involved in the T&E Should Comprise the IOC Cadre and the "Manning System" Should Be Altered to Facilitate This</u>

The people who were involved with the operational test of a C^2 system are the most familiar with the system and, therefore, are best equipped to make the initial deployment of the system. The existing personnel-manning "system" in the military is not set up to accommodate this. In the case of one particular program, a program manager had to take extraordinary action and go to the highest echelons of HQ DA to get the personnel "system" to bend such that people from the operational test cadres could be assigned, by name, to the initial deployment units. As a corollary, users who participate in the development should take the system to the field for T&E in the user environment.

c. <u>Changes in the Roles of Participants in the "Requirements Process"</u>

2.3 THE "REQUIREMENTS PROCESS" MUST BE ABBREVIATED AND EXPEDITED TO ENABLE EARLY FIELDING OF THE "CORE" TO INITIATE EVOLUTION BASED ON FEEDBACK FROM T&E IN THE USER ENVIRONMENT.

The third and last major subrecommendation is that the "requirements process" must be abbreviated and expedited. DoD must take actions to assure that requirements preparation/validation and program approval persons will recognize that, for C^2 systems, the requirement process is continuous and interactive--an "eternal" process based on feedback from T&E in the user environment.

Giving an overall framework and desired functional characteristics should suffice to get the program started. This "requirement" could define the "core" or basic capability desired (NOT the hardware/software solution), within an overall mission/architectural framework designed to accommodate change, and within a mutually-understood resource constraint.

In short, the primary difference is that the objective is to field the initial "core" within a suitable architecture. Feedback from T&E of this "core" in the user environment (and, of course, subsequent increments, when fielded) is the primary means of refining, amplifying and evolving to the "true" requirement. Unlike the approach for weapon systems, the requirements process for C^2 systems is "eternal." While it is advisable to keep the requirements documentation updated, these updates should not be serial to approval of the next increment/block.

d. A Major Caution

A key point in all of these recommendations regarding altering the roles of the user/provider/tester is: DO NOT INCREASE THE BUREAUCRACY. The Study Team does NOT advocate more approval cycles, more people in the signature act, etc. Coupling users, providers and testers could do just that, IF evolutionary acquisition is implemented merely by unimaginatively jamming together all the bureaucratic procedures and approvals for which there are now three separate processes! The end objective is to field increments of military capability sooner. CAVEAT IMPLEMENTOR!

4. Other Conclusions and Recommendations

Based on our review of ILS, especially as applied to EA, the Study Team concludes the following:

- An aggressive, knowledgeable, and technology-oriented program office is the key to planning, developing, deploying, and evolving effective ILS for C^2 systems. ILS planning (and Plan execution through adequate funds) is a crucial part of successful EA.

- Closely-coordinated joint planning by Logistics Commands, Training Commands, and User organizations (with the program office serving as planning catalyst) is the key to successful ILS, regardless of the acquisition strategy used.

- Dependence on contractors for critical maintenance, parts supply, and training is practical so long as the Logistics, Training and User Commands (with the program office serving as planning catalyst) insure that an adequate soldier/sailor/airman capability for wartime ILS is established in the earliest practical time.

- Dependence on contractors for parts supply and maintenance at depot levels is practical and cost effective even in wartime, provided the Logistics Command and User (with the program office serving as planning catalyst) have staffed and implemented their part of the overall ILS.

- Requirements for logistics support of future-deployed C^2 systems will be affected by the major strides being made by technology in reducing size, weight, and cost of hardware while improving performance, inherent ruggedness, reliability, and maintainability. An additional factor is the ongoing implementation of innovative remote diagnostics and simple field repair capabilities, including provisions for module swap in the field by minimally-trained persons using minimum special tools.

- Software configuration management is the single most complex and riskiest portion of ILS. System-level hardware and software configuration management at the program office level, with contractor support, and complementary configuration control at the field applications level, will facilitate user-responsive evolutionary development approaches.

- Software standardization at the Ada HOL level is important within DoD and should be strongly supported.

CHAPTER V

USE OF IMPROVED SYSTEM ARCHITECTURE AND DEVELOPMENT PRACTICES

CHAPTER V
USE OF IMPROVED SYSTEM ARCHITECTURE AND DEVELOPMENT PRACTICES

A. INTRODUCTION

As has been noted earlier in this report, C^2 systems are different in several major respects from other systems. These differences are such that many aspects of the traditional system design and acquisition process need to be changed in order to ensure useful and cost-effective C^2 systems. The Study Team found that the concept of evolutionary acquisition was well suited to the C^2 system characteristics that presented problems for traditional approaches and methods. In fact, the Team could find no successful C^2 program acquired in the traditional manner. This section will explore those architectural and technical issues related to acquiring C^2 systems in an evolutionary manner and will discuss the applicability of current concepts and technology.

B. C^2 ARCHITECTURE[1/]

C^2 systems can be viewed from two major perspectives: (1) from an operational or mission view and (2) from a systems or technical view. The operational or mission view is concerned with what the C^2 system must do to support a given military mission(s) and how information, decisions, and tasks which are collected, made, and performed relate to force deployment and employment, given a set of threat environments. The systems or technical view is concerned with how (and how well) the C^2 system collects, analyzes, transmits, and displays appropriate information in a timely, reliable, and understandable manner. C^2 systems exist and must respond to change in each of these dimensions.

1/ See Appendix H for a comprehensive discussion of information system architecture in C^2.

Hence, two architectures are required to co-exist: (1) a theater/mission architecture to define the operational context and functional (military) requirements (how they may change) and (2) a system architecture that defines system capabilities (technical characteristics and performance) and interfaces.

Neither of these "architectural" tasks is easy. Since every command has, and must use an existing C^2 system, any C^2 architecture must use the bits and pieces of organizational and systems "materiel" (including doctrine and concepts) that are holdovers from another time, both militarily and technically ("backward compatibility").

Operational concepts (at least those practiced in crisis or wartime) often are constrained and sometimes even determined by the technical aspects and capabilities (or lack of) of the C^2 systems which are available at the particular time the new or improved C^2 system is needed.

As a practical matter, many of today's systems are clusters of subsystems or computing elements developed by different people with different perspectives at different times, which are later connected together to extend the range of their capabilities. This ad hoc interconnection of disparate computing elements is, of necessity, a marriage of convenience, often motivated by a desire to extend capabilities developed at one node in a network (often at considerable expense) to other nodes in the network. Often implemented via "black box" technology rather than planned compatibility, this solution to interconnection is not fully satisfactory even when new, and is subject to rapid obsolescence.

The current threat and the likely nature and tempo of future conflicts require a far more capable and flexible C^2 system than can be provided by these *post hoc* "black box" approaches to interconnection.[1/]

Sections 1 and 2 below discuss the two architectural prerequisites necessary to achieve needed C^2 capabilities: (1) the implementation of Theater/Mission Architectures and (2) the use of a layered system interconnect reference model, such as the International Standards Organization (ISO) reference model for systems architecture.

1. Theater/Mission Architecture

The Theater/Mission Architecture is the reflection of the force structure which comprises the operational environment(s) in which each C^2 system must perform. Because C^2 systems include embedded processing which support both the Commander's decision-making activities and the Commander's connectivity with other players, each change in the operational mission may require changes to the C^2 system. Therefore, changes in the Task Organization, Contingency and Operations Plans, Command Structure, or Force Assignment must be reflected in supporting C^2 systems, in order to have a battle-ready, deployable system.

1/ These represent a "bottoms-up" piecemeal approach to architecture which has resulted in some very disturbing interoperability problems for the DoD. "Top-down" management of DoD programs has also had its share of problems. The Study Team believes the systems architectural approach discussed in this section will help in achieving a workable blend of "bottoms-up" and "top-down."

As established earlier, user involvement in the development of the theater/mission architecture is of paramount importance. An analysis of the force structure, communication requirements, and information processing needs must be performed and regularly updated to form a basis for individual (nodal) C^2 system development. This mission architecture also must support the identification of nodal functions, identification of users (many of whom are multi-task, multi-function, multi-mission), and the need for functional interfaces between systems.

The Study Team found that C^2 systems have and are being developed largely to satisfy the needs of the Service components, with little attention paid to the operational demands of the Unified Commands, within whose wartime chain of command these Service component systems must operate. More often than not, unified and/or multi-national commands will form the chain of command, with corresponding rules of engagement, which determines the battlefield context of C^2 system interoperability and survivability. In addition to the issue of prioritizing Service needs versus CINC's needs, each theater has disparate geographic, political, and operational needs. If not addressed in advance, interoperability among these systems, as it now exists, would have to be attempted in real time in a crisis or wartime situation. Programs to protect C^2 system interoperability, such as JINTACCS, currently are aimed at operational, character-oriented message systems. Developed systems are retrofitted to achieve limited system-to-system interoperability. This is the most expensive possible method of achieving interoperability and is also rarely satisfactory. There now exists an urgent need to provide a preplanned framework which reflects the needs of the theater, tasks, and operations plans. Under evolutionary acquisition (EA), there will be a continual upgrading of existing capabilities. The lack of a plan to support the EA approach, and for integrating the individual systems into a coherent "system of systems" in the potential wartime environment battlefield, could lead to utter chaos for the user.

In each environment, there is a natural hierarchy of users that must be considered. Figure V-1 represents a simplified US chain of command for Europe (prior to CHOP) and shows the hierarchical structure of a chain of requirements using NAVEUR as a vertical example. Each organizational element must exchange information with its components and subsidiary elements. (This chain, of course, is different after CHOP to NATO at Alert. The architecture must also address these different relationships.) The actual C^2 systems exchanging information, the types and quantities of information, and the decisions which must be supported at each level varies with the Operational Plan/Mission/Force Assignment/Rules of Engagement combination.

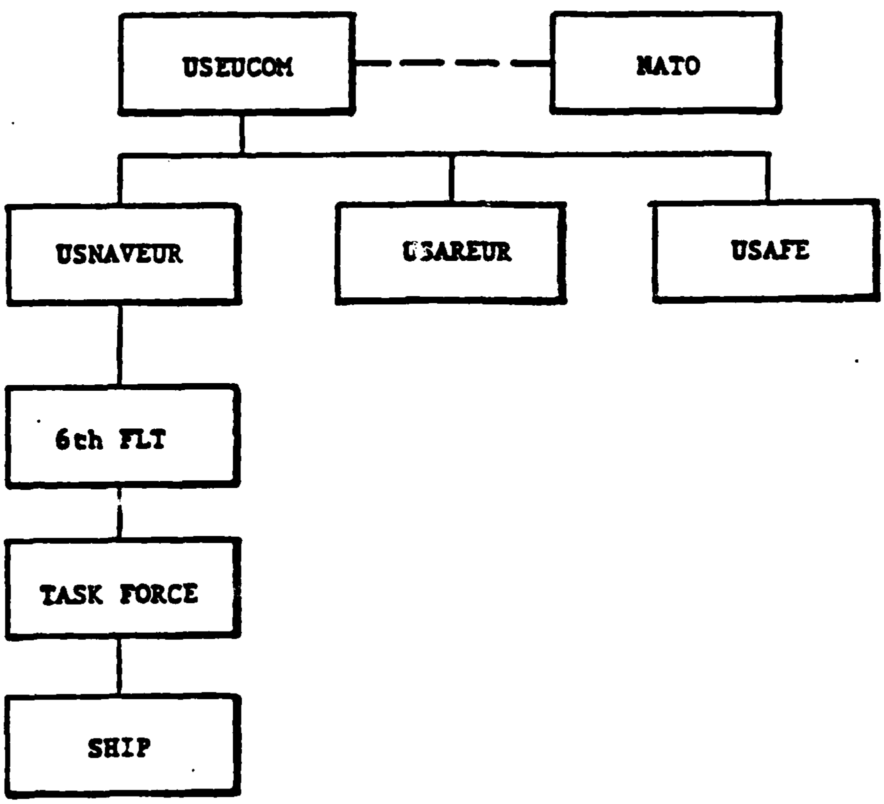


FIGURE V-1. SAMPLE OPERATIONAL CHAIN OF COMMAND FOR US FORCES IN EUROPE

Each echelon must be capable of specifying its information needs in terms that recognize the existence and requirements of other portions of the chain. Without this there can be no operational "system of systems" (mission architecture). Each integral part of this chain must be able to introduce new support subsystems without destroying interoperability. Each user level requires a definition and plan for introduction of information systems capability. In the example shown in Figure V-1, HQ USEUCOM would define information needs for the European theater and the concomitant responsibilities for the Component commands. Information needs must be defined in conjunction with the Contingency/Operations Plans for the theater. Subsequently, the Component commands would develop their plans based on the theater plan and projected systems introduction. New system introductions will be both more frequent and more accurately schedulable under Evolutionary Acquisition.

2. System Architecture

The integration of multiple systems into a "system of systems" to form an integrated command and control capability to support C^2 functions, and which is capable of evolving over time, requires careful technical planning and the employment of a suitable system architecture.

This need for such a common architectural framework is not unique either to C^2 or the military. Work on a suitable architectural methodology has begun in the commercial field, and products employing these concepts are in place and being used effectively today to manage interfaces in complex systems. Over the past few years, the International Standards Organization (ISO), has developed an architectural framework called the ISO Open System Interconnection Architecture (ISO OSI). This framework, or an adaptation (such as the DARPA effort), embodies the attributes needed to support the evolutionary acquisition of C^2 systems, and should be reflected in application of system architectures to DoD C^2 systems development. The concepts employed in ISO's OSI model need to be augmented further to handle problems of particular concern to DoD, such as multi-level security and internetting of individual networks.

The ISO OSI Architecture is a layered structure which defines and supports interfaces between system functions and, hence, provides a framework for compatibility and interoperability. The layered structure, a universal architecture for the development of future protocol standards, has gained world-wide acceptance and is recommended as a viable approach to DoD C^2 system development. While it is not yet possible to define a standard interface to allow interconnections and digital data exchange between all systems (there are multiple possibilities), considerable progress has been made toward defining a structured approach to interface logic which decomposes the process into elements which may be standardized. The use of the model has been very successful in the development of specific standards, and we believe that the layered concepts of this model can be extended to the C^2 integration problem with the same benefits.

The term "open system interconnection" refers to systems which are "open" to communication with other systems by virtue of their mutual adherence to standardized procedures. Technical interoperability between two C^2 systems is no longer as simple as radios tuned to the same frequency, or even of messages/formats containing data in recognized fields. Computer-based information systems interact in a much more complex way. Several architectures have emerged for system interconnection; the majority of commercial architectures have been developed for a single manufacturer or product line. The ISO OSI Architecture is a universal architecture with concepts which apply for communication and interoperation between heterogeneous networks and end-user systems (or components).

As the most widely accepted and promising approach to achieving a technical foundation upon which evolution can be based, the use of a layered architectural model is mandatory for successful C^2 system development.

The ISO OSI layered model, with its substantial, already proven features and benefits, is being adopted commercially worldwide and standards implemented. Therefore, the cost implications and time delays in inventing a new DoD model appear unwarranted.

The remainder of this section will discuss the ISO OSI model from a conceptual approach only.[1/]

The OSI model is concerned only with the exchange of information between system layers, and exists to provide a structure for the development of interconnection standards. As such, the model defines and uses seven conceptual levels, or layers. The layers, as presented in Figure V-2, are hierarchical, with each layer using the functions of the lower layers to accomplish its own functions. The following is a description of the functions of each layer, from lowest to highest.

- Physical Layer - Provides the electrical, mechanical, and functional characteristics to activate, maintain, and deactivate physical connections for bit transmissions. The physical layer is concerned with the flow of 0 and 1 bits and would define, for example, the voltage levels representing 0's and 1's, the speed of transmission, assignment of network connector pins to leads in the cable, etc. The EIA RS-232 interface standard is an example of one widely used Physical Layer standard.

- Data Link Layer - Provides the functional and procedural means to establish and release data link connections between network entities. The Data Link Layer handles transmission of "frames," which are the basic unit of information exchanged. Since the Physical Layer receives and identifies 0 and 1 bits, they are passed to the Data Link Layer to be accumulated into error-free

1/ The reader with detailed interest in data communications is referred to "Reference Model of Open Systems Interconnection," ISO/TC97/SC16/N227, and "Progress of the Reference Model of OSI," ISO/TC97/SC16/N309. Appendix H also contains some amplifying material.

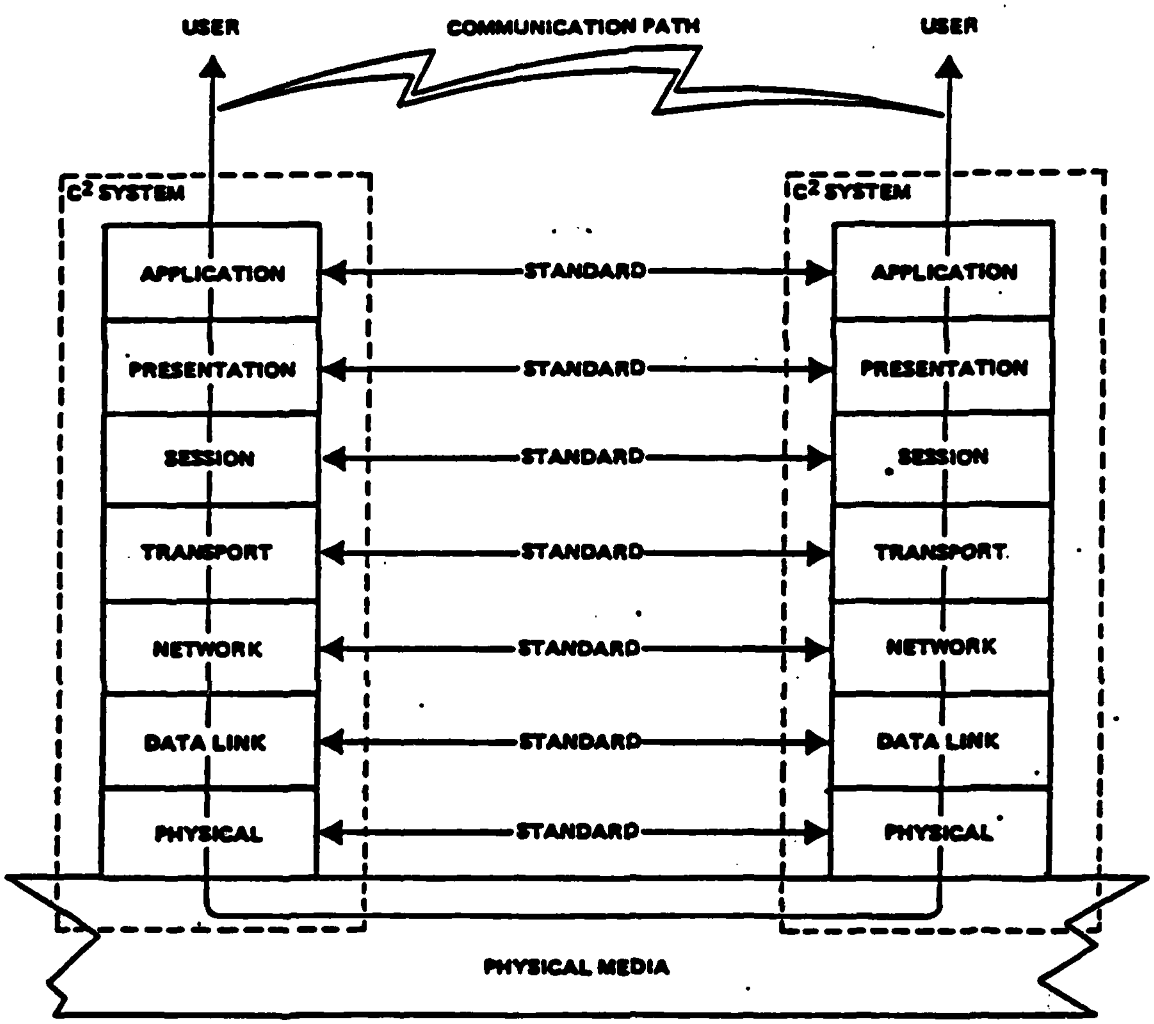


Figure V-2. System Interface via the ISO OSI Layered Architecture frames. The Data Link Layer is concerned with knowing how to tell start-of-frame, end-of-frame, error detection, address bits, and other control bits. The Data Link Layer sends frames back and forth, but ignores the information content within a frame. The BISYNCH and Synchronous Data Link Control (SDLC) procedures are two IBM Data Link Layer Standards in common use.

- Network Layer - Provides the means to establish, maintain, and terminate network connection between communicating systems. It provides independence from routing and switching considerations. After the Data Link Layer receives an error-free frame, it removes the header and trailer bits from the frame and passes the information content, called the "packet" to the Network Layer. The Network Layer then uses only the header portion of the packet to route the message portion of the packet to the proper destination in the proper sequence (a long message may be sent in several packets, not necessarily received sequentially). The Network Layer can become quite sophisticated, to include dynamic rerouting around congested nodes and to include accounting and traffic charge costing. Examples of existing Network Layer standards are TELNET and TRANSPAC systems.

The preceding three layers are sometimes referred to as the "Lower Layers" or Communication Net. Together, they move data from "System A" to "System B," which is sufficient for data relay or simple systems.

- Transport Layer - Provides end-to-end control of the data exchange and provides transfer of data between Session entities. Real networks usually are composed of subnetworks and grow in time, adding new users with new equipments and new requirements. The Transport Layer handles translation between subnetworks, providing network management within systems in a fashion similar to the Network Layer between systems. The Transport Layer allows multiple distributed computers, terminals, and data links. Because of the necessarily diverse combinations, no single Transport Layer standard has emerged in common use.

- Session Layer - Provides connection services which establish a dialogue between Presentation Layer entities and supports the orderly exchange of data. For example, if a line printer is connected to a communications link, the Session Layer ensures that data is sent to the printer at its rated lines per minute capacity. If a terminal is connected to a data base manager, the Session Layer supports rapid responses to the terminal and pauses from the terminal end (human think and reaction times).

- <u>Presentation Layer</u> - Presents information to Application entities in a way that preserves meaning while resolving syntax differences. Different computing systems have different file formats, CRT screen formats, line and character formats, and data compression-expansion or encoding-decoding mechanisms. The Presentation Layer performs format conversions so that the end result is meaningful to the receiving application computer program.

- <u>Application Layer</u> - The "catch-all" name which designates all computer programs within each C^2 system which accomplish the unique tasks of the particular C^2 system. The definition problems of this layer directly relate to the definition and specification of a Theater/Mission Architecture. To achieve interoperability and transfer of meaningful data, the Application Layer must be defined in the context of the other layers.

These seven layers provide a structure which forces a more formal and visible definition of interfaces and the process of interfacing data. This, in itself, provides a basis for developing standards within each layer. Secondly, because of the functional layering concept, each interface is reduced only to the specific relationship between the adjacent layer (above and below) and the peer process in the interfacing system. This reduces the scope of each interface to a manageable size. At present, only the lower four levels have had extensive development of specific standards. Work is in progress on the higher levels.

The Study Team believes the application of a layered architectural model provides a potentially powerful tool for the integration management of C^2 systems because it provides the following:

- A systems engineering tool for defining system interfaces at all functional levels,

- A management tool which defines interfaces in a manner which allows structure in implementation,

- An architecture which supports an evolutionary approach to systems integration, and

- A common reference point for the developers of individual systems to define interactions and compare requirements.

The Study Team did not specifically investigate DoD's utilization of standard Instruction Set Architectures (ISA), but it was generally believed that such standardization outside of (and precluding use of) the commercial software technology mainstream is very risky, particularly for C^2 EA environments, where provision for technology insertion needs to be made. Related considerations are covered in Chapter IV.

C. TECHNICAL SUPPORT TO THE DETERMINATION AND EVOLUTION OF SYSTEM REQUIREMENTS AND CAPABILITIES

The definition of C^2 system capability requirements, and the communication and translation of these requirements into terms useful to the development community, has been an extraordinarily unsuccessful process. Two factors seem to be at the root of the problem. The first is a function of the difficulty of knowing what capability a C^2 system should have (a blend of what is needed and what is feasible), while the second involves the communication among the parties involved.

This section discusses the development and use of tools or capabilities to facilitate: (1) better understanding of a system's potential and employment on the part of the operational user, and (2) the communication between the user and system designer/developers.

1. Understanding What Systems Can Do

The development of automation has been rapid and continues to accelerate. The applications of automation and the "sciences" of computers are immature. Hence, the basic concepts of automation and the intuition-level understanding of what computers are, what they can do, and how they work has not had time to disseminate through society. Computer knowledge is still largely restricted to "technicians" who specialize in computers. Even these technicians, because of the rate of change, have yet to establish a common vocabulary. Peer-group technical communication is frequently characterized by efforts to define terms "for purposes of this discussion" and the heavy usage of examples to ensure understanding.

2. Understanding What is Needed

Experience has shown that even when individuals are capable of accurately envisioning how a system would operate (as a result of prior experience, education, and training), their ideas change substantially after "hands-on" experience. New uses were discovered and previously anticipated uses proved to be impractical or unnecessary. For individuals without previous experience or training, the situation was less positive. Systems which were delivered bore little resemblance to what was (rightly or wrongly) imagined by the user and were greeted with reactions ranging from open hostility to benign neglect.

3. Facilitating Understanding and Communication

The user currently cannot (and should not be expected to) adequately visualize what a system could do or how it might work without experience with a similar system. This fundamental requirements development problem in the C^2 area, partially caused by the "language gap" between the users and providers, can be significantly alleviated under Evolutionary Acquisition.

EA recognizes that a user's perception of the capabilities which must be provided by a C^2 system is strongly influenced by experience. Hence, under the traditional acquisition process, the fielding of a C^2 system represents, to some extent, the beginning rather than the end of the system requirements definition process. EA seeks to allow a user to gain experience before a system design is cast in concrete and provide a process permitting change, which is desired as a result of experience to be easily incorporated into the system.

For C^2 systems, consider how the traditional acquisition process, a well-defined and sequential set of steps based on written communication, results in the development of a set of requirements from an abstract concept. EA, a process which is based upon "hands-on" experience with and interactive feedback between operational experience and design, provides a mechanism for exploration of alternate concepts.

As discussed in Chapter III, the establishment of a combined Program Office (user/provider/tester) could provide the proper organizational mechanism for ensuring continuous user/provider interaction. The combined Program Office should establish, as a first order of business, a Rapid Requirements Definition Capability (RRDC) which will later evolve into a permanent System Design/Support Facility (SDSF). The SDSF will be a key mechanism to allow solution of the technical development problems associated with EA. The RRDC/SDSF may be colocated with the user or the provider community, but during the initial requirements definition phase, there should be strong user/provider participation at the C^2 system deployment location. The RRDC/SDSF can, quite inexpensively, provide a capability to help overcome "language" and experience gaps, both during the critical definition of the first evolutionary increment ("core"), and throughout the life of the system. The Study Team envisions that the RRDC/SDSF can be used to advantage in the following two ways:

a. Rapid Requirements Definition Capability (RRDC)

Recently-developed computer system technological capabilities have made it possible to rapidly develop "operational mock-ups" which can be used to provide "hands-on" experience.

This operational mock-up capability is distinct from, and should not be confused with, either a prototype or a test bed.

A major difference is that a prototype or test bed[1/] actually implements the capabilities which are envisioned. This attribute of a test bed or prototype makes these facilities take longer to develop and be more costly to build, particularly when they are close to pushing current technology (as they often are). Further, system prototypes lack the capability to change quickly, since they were designed to deliver the intended capabilities efficiently . In addition, test beds and prototypes usually require significant development in themselves, adding additional time and cost to a "design" process which needs to be foreshortened under EA. Prototypes should not be undertaken until the system capabilities needed are well (better) understood.

In concept, RRDC can be assembled quickly since it can be put together using off-the-shelf technology. The objective is to design a RRDC to be flexible enough to mock-up or simulate a wide variety of capabilities easily and quickly, so that user "experience" can be quickly obtained, and a system concept, architecture, and a set of initial ("core") capabilities can be developed. Many of the mock-ups can be accomplished through the use of display variations using existing table-driven report generators, graphics packages, and data base management systems running on inexpensive general-purpose machines. Commercial technology is rapidly evolving flexible man-machine interfaces which would play an important part in such a facility.

1/ Although many test beds contain simulations, these usually are present to provide external interfaces (e.g., threat and environmental inputs) and not to provide system capabilities for the system of interest.

Using the RRDC, the users and providers can perform hands-on "what if" experiments. Actual implementation of a capability is not necessary to determine if, how, and to what benefit the user can employ a given capability, and to obtain feedback from the user with respect to desired modifications. If these desired modifications can be "made" quickly (the results of the modifications appear on a display screen), the user can again react, and through an iterative process, develop a good sense of the operating characteristics which are important in a particular C^2 system.

Interactive war games, or battlefield simulation, also could be provided for exercises, to allow the user to select and quantify information flow and access requirements. These tools could exceptionally useful in the context of assessing alternate or propo changes in Theater/Mission Architectures.

b. System Design/Support Facility (SDSF)

Upon implementation of a system architecture with an initial or "core" capability, a RRDC can be augmented by "real" capabilities (evolve into a SDSF), and serve: (1) as a basis for determining the nature of future increments and (2) as a place for the testing of newly-developed system changes or enhancement. During the design and development of each increment, the SDSF will be used to review and monitor the process (when not implementing or administering the development contract) and can represent a focus for system planning.

The traditional development process necessarily engenders pressures to motivate short-term optimization at long-term expense. A current common example is the decision to waive software documentation and/or programming standards that is too-often made on programs to save perhaps ten percent of the development costs, with the future effect of doubling the maintenance costs throughout the life of the system. The

SDSF, as the organization which monitors new development in each increment, while performing the maintenance on deployed increments, would institutionally adopt the balanced viewpoint which best served DoD interests.

D. DESIGNING FOR CHANGE

Since it is difficult, if not impossible to specify C^2 system details in advance, a requisite for any C^2 system design is an ability to adapt to changes with minimum redesign and reprogramming. As this and other studies have documented, the inability of a system to anticipate and gracefully accommodate change results in delays and cost growth in system development, later IOC, user dissatisfaction upon deployment, excessive maintenance costs, and rapid obsolescence. This section will discuss briefly the sources of change for C^2 systems and the methods and tools which are available to "design and build for change."

1. Sources of Change

There are four major sources or drivers of change in C^2 systems: Two are derived from changes in the nature of the environment; a third stems from the nature of the human element; and the fourth is derived from technological changes.

a. Environmental Sources

Changes in the environment come from either: (1) an evolving threat or (2) an evolving organizational, doctrinal, and systems environment (which are often reactions to forecasted threats). The nature of the changes that are required of C^2 systems in the face of this dynamic environment basically can be satisfied by changes in: (1) capabilities, (2) connectivity, and (3) interfaces. Examples of each of these are as follows:

- Capabilities - In response to an evolving threat, additional information about enemy forces and coverage of a larger area of operations are needed.
- Connectivity - A new node or change in the communications paths among nodes to provide, for example, more direct or timely information (e.g., a direct link between the E-3A and an Army Air Defense command post).
- Interfaces - A new system (or replacement for an existing system) is fielded and must connect to other systems which comprise the "system of systems."

b. Management Style

The intimate relationship between man and machine in C^2 systems generates its own set of requirements for change. These changes may be the result of: (1) variations among users (essentially style driven), (2) user's experience with the system and environment (learning), or (3) changes in threat or operational situations. The "learning" process, while continuous, is most rapid in the initial phase of a system or new capability and, therefore, the Study Team has recommended the employment of a Rapid Requirements Development Capability (RRDC) discussed in the previous section. In order to accommodate change in the style, threat, and situation, the system must have built-in flexibility to easily change functional processing, procedures, displays, messages, data bases, and system interfaces in the field.

c. Technology-Driven Change

The third major source of change is technology-based (technological "push" rather than demand "pull"). As advances are made in technology, these are reflected in changing user expectations and aspirations and new ideas in the development community. Performance or capabilities which were once (and still might be) adequate from a mission standpoint become old and outdated in the minds of developers and users. Further, the existence of improved technology (capabilities) often gives birth to new concepts of operation not previously possible or practical. The result of these altered perceptions is dissatisfaction with the present system and a corresponding demand for changes. For example, recent improvement in data base retrieval technology and associated hardware cost reductions make it possible to deal with data bases of greatly increased size and complexity. Correspondingly, demands for or offers to provide more detailed data have increased.

If the system has been designed and implemented in a modular fashion (see discussion later in this chapter), this insertion would not be difficult. On the other hand, many existing systems have software and hardware implementations which have logic and data functions so intertwined that this would be almost impossible to do without major or complete redesign.

Thus, if a C^2 system has the ability to insert, on a continuing basis, the products of rapidly-advancing hardware and software technology, it would greatly enhance the system's expected life and its value. Correspondingly, benefits of an evolutionary approach to C^2 system acquisition would be enhanced by allowing for greater freedom and flexibility within a given system architecture.

With these potential benefits in mind, C^2 system program managers should be encouraged to implement capabilities (based on what is available) in a way which will facilitate the insertion of future technology, rather than to try to have each program push the frontiers of technology to combat expected obsolescence.

2. Cost of Change

The need for C^2 systems to be designed and built to accommodate change has been emphasized frequently throughout this report. The remainder of this section will be devoted to the cost and penalties that systems which have not been designed, built, and operated to accommodate change incur, and will continue to incur, as well as the design and development considerations and methods which can be expected to avoid or reduce these costs.

Changes to systems may either involve additional or new hardware or software, or "fixes" to software. Hardware or software replacements or upgrades have traditionally been associated with new features or increased capacity, while software "fixes" usually have been associated with correcting error. Experience shows that while some of this software "maintenance" can be traced to "errors" in the code, the overwhelming majority of this activity is involved in accommodating the types of change discussed in the previous section. That is, these fixes are not the result of programmer error, but rather from a desire to have the system behave differently from its current specifications.

The software "maintenance" activity has been specifically singled out here because it has become a dominant factor in the life cycle costs of systems of the type we are considering here. In fact, software costs have accounted for an increasing percentage of computer-based system costs over the past decade.

A recent EIA study projecting DoD data processing needs and costs for the 1980s[1/] estimates that software and services (operations, maintenance, and training) costs will continue to grow at a much faster

1/ "DoD Digital Data Processing Study - A Ten-Year Forecast," Electronics Industries Association, October 1980.

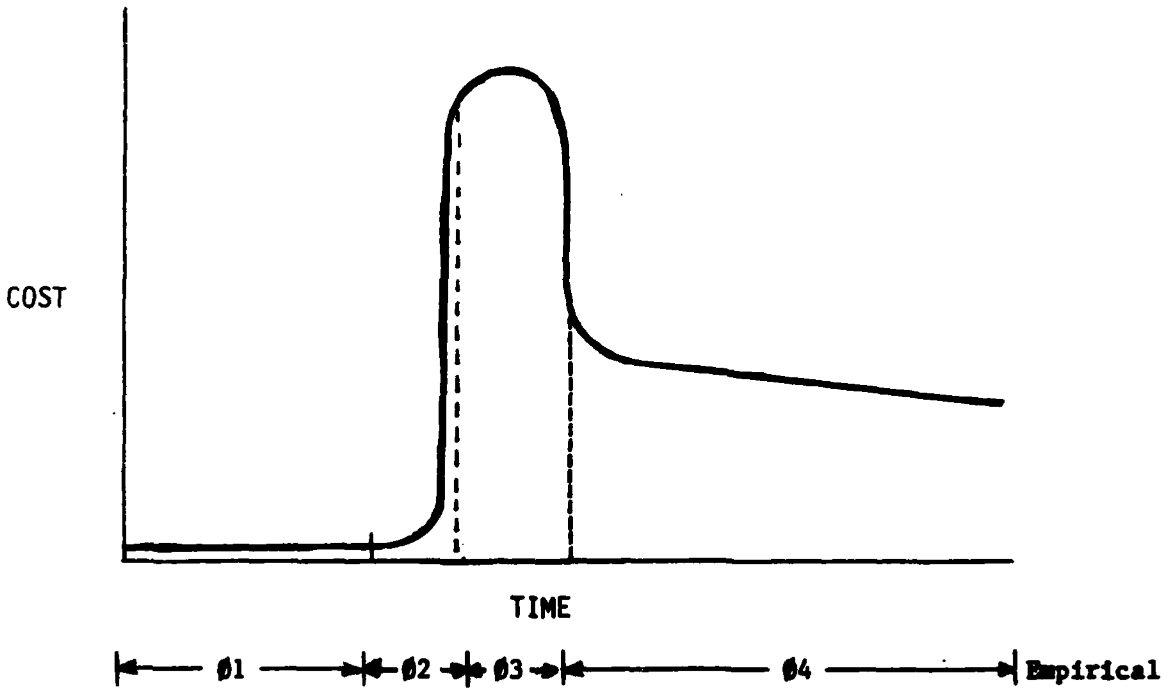
rate than hardware costs throughout this period. They estimate that software and services will go from almost 70% of the total defense computer expenditures which were experienced in FY-80 to slightly over 80% by FY-90. During this same time, DoD expenses for computers, currently $6.7 billion in FY-80, are expected to grow to almost $46 billion by FY-90. Thus, software will be taking up a larger part of a rapidly expanding pie.

In breaking down the components of software costs as a function of life cycle phases (conceptual, requirements, development, operations & maintenance), it was found that roughly half (50%)[1/] of total system life cycle costs were incurred after installation, that is, in what is commonly called operations and maintenance (see Figure V-3). These data were derived primarily from "one of a kind" systems, which would mean that for systems with multiple copies (such as tactical C^2 systems), operations & maintenance costs would exceed amortized development costs.

The analysis cited above further reported a number of statistics, discussed below, which indicate that systems acquired in an evolutionary fashion and employing the design concepts and techniques recommended in this report could be expected to cost significantly less over their life times, and have capability available more quickly while achieving greater levels of user satisfaction.

As indicated above, the "operations & maintenance" phase of a software life cycle is devoted more to enhancements and elimination of errors introduced into the system as a result of implementing these enhancements than to "fixing-up" errors in the software which escape

1/ "The Economics of Software," D. S. Alberts, The MITRE Corporation, 1978.



Phase 1:	Conceptual		1%
Phase 2:	Requirements		1½%
Phase 3:	Development		47½%
	Analysis and Design	34%	
	Coding and Debugging	18%	
	Validation	48%	
		100%	
Phase 4:	Operations & Maintenance		50%
			100%

FIGURE V-3. SOFTWARE COSTS VARY OVER LIFE CYCLE development testing. While little data were found which allow one to put a precise number on this percentage, it is generally believed to exceed half of O&M phase costs. Exacerbating this situation is the fact that these "enhancement-related" costs increase as a system ages, while performance and reliability often decrease. The cited study also found that nearly half of the cost of development could be attributed to finding and correcting errors ("validation").

In analyzing the types of errors which were found, the cited study concluded that the overwhelming majority of the cost of error (83%) could be attributed to <u>design</u> errors, with only a fraction (17%) attributed to coding-type errors. Design error was defined to involve situations in which the system did what it was supposed to do, but that is not what was really wanted. In other words, design error was equivalent to what we have been calling either a system "enhancement" or a change in requirements. When taken together, these costs associated with change account for almost half (44%) of total life cycle costs.[1/]

Two ways exist to reduce these costs. The first is to develop a better understanding of system needs, and a better mechanism for their articulation. The RRDC described earlier in this section can be expected to achieve this. This can be expected to result in fewer changes in the period immediately following installation (currently a tumultuous shakedown period) and fewer "false" steps in subsequent increments. The second is to reduce the cost of making changes. An architectural model like ISO's OSI, and the use of better software development practices (discussed later in this chapter) should achieve a reduction in the costs of changing software.

1/ Calculation of cost attributed to change/enhancements.

Phase	(1) % Total Life Cycle Cost	(2) % Attributed to Error in that Phase	% Cost of Error Attributed (3) to Design Error/ Enhancement	(1)x(2)x(3)
Development	47.5%	48%	83%	19%
O&M	50.0%	50%	100%	25%
				44%

Delays in capability being delivered to the user is another form of cost and occur as a result of the following:

- A lengthening of the time to develop requirements and specifications,
- Numerous decisions required for moving from phase to phase, and
- Changes which need to be defined and implemented and errors which need to be detected and corrected.

Since 1960[1/], the length of time required to field major DoD systems has increased 42% to an incredible 17 years. Thus, the lead time far exceeds the expected useful life for many systems (and covers about five generations of "chip" technology). This time increase from conception to deployment has been experienced, for the most part, in the "front end," that is, in the time from the recognition of the need, to the development of system test beds or prototypes. In 1960, this time averaged about two years. More recently this phase of a system's life cycle was averaging more than six years. EA concepts and the use of a RRDC clearly would serve to reduce this front-end time and, therefore, get useful increments in military capability to the user more quickly. Delays also would be reduced in two other ways. First, by reducing error rates, delays associated with their detection/correction also will be reduced. Second, an orchestrated EA approach will reduce "lead-time" associated with decision milestones. Although "lead time" reduction can be achieved, to a limited extent, under the traditional acquisition approach, decision milestones in an EA approach are more amenable to a continuous process.

1/ Report of the Acquisition Cycle Task Force, 1977 Summer Study DSB, OUSD(R&E), 15 March 1978.

3. Design and Development Considerations

Rapid technological advances in hardware have radically changed the economics of processing power and computer networking. Somewhat less but notable progress has been made in software design and development techniques. While these technological advances offer opportunities to design and implement systems which are based upon significantly different concepts, system design and acquisition practices have not kept pace.

Advances in technology have made feasible more "designing for change":

- The movement of substantial capability closer to individual users with significant improvements in the interface between the user and the system will allow for more tailoring to the individual,
- Improved system architectures and protocols (e.g., ISO OSI model and its implementation) will make it possible to become more independent of hardware and system software, and
- Networking technologies, both local and long-haul, will provide for more flexibility in connecting and reconfiguring a dynamic set of users.

These three fundamental achievements will make it possible to alter significantly the way systems are conceived, built and utilized, and mark the beginning of an era which will be characterized by "evolvable systems." The concept of systems which can gracefully accommodate change is not new. What is new is that current technology and methodology now make it feasible to apply this concept to large information and control systems.

Implementing these changes may necessitate either software or hardware modifications, or both, depending upon the architecture, design, and the way in which the system(s) was implemented. The ease or difficulty and cost associated with achieving a particular enhancement also is a function of these three factors. The remainder of this section will address the attributes and characteristics of software and hardware, as well as methods of minimizing the impact of the changes which a system is called upon to accommodate.

a. Software Issues

While modularity as a concept can apply to hardware, software, and operational functions, the application of this concept to software design and development offers the most potential for improving systems. As new technology has dramatically driven down hardware cost, software has become the dominant factor affecting C^2 system performance and schedule. As indicated earlier, software costs now far exceed hardware costs, often even when multiple copies of the system are proliferated. As indicated earlier, a very large percentage of life cycle costs can be attributed to the accommodation of change by altering software or reprogramming. Should Evolutionary Acquisition be mistakenly taken only to mean "build a little, test a little" and result in a series of increments which duplicate the traditional approach to design and acquisition, rather than be clearly associated with the development of an overall system architecture which has been designed to accommodate change (with the system "core" being the initial increment within this architectural framework), EA would be a prescription for even greater delays and cost overruns. It is mandatory that the "core" capability be based upon and structured within a suitable architectural framework. If special provisions are not made to ensure that the initial or "core" capability exists within such an architecture, subsequent increments not only will cost more to develop but the costs of system "maintenance" will grow substantially, and reliability and performance will be degraded as the system becomes a partially-documented patch-work "cluge" of "new" and "old" code. Figure V-4 graphically depicts the cost profile which might occur if EA is practiced without the architectural foundation it requires, while Figure V-5 depicts the cost profile which EA has been designed to achieve.

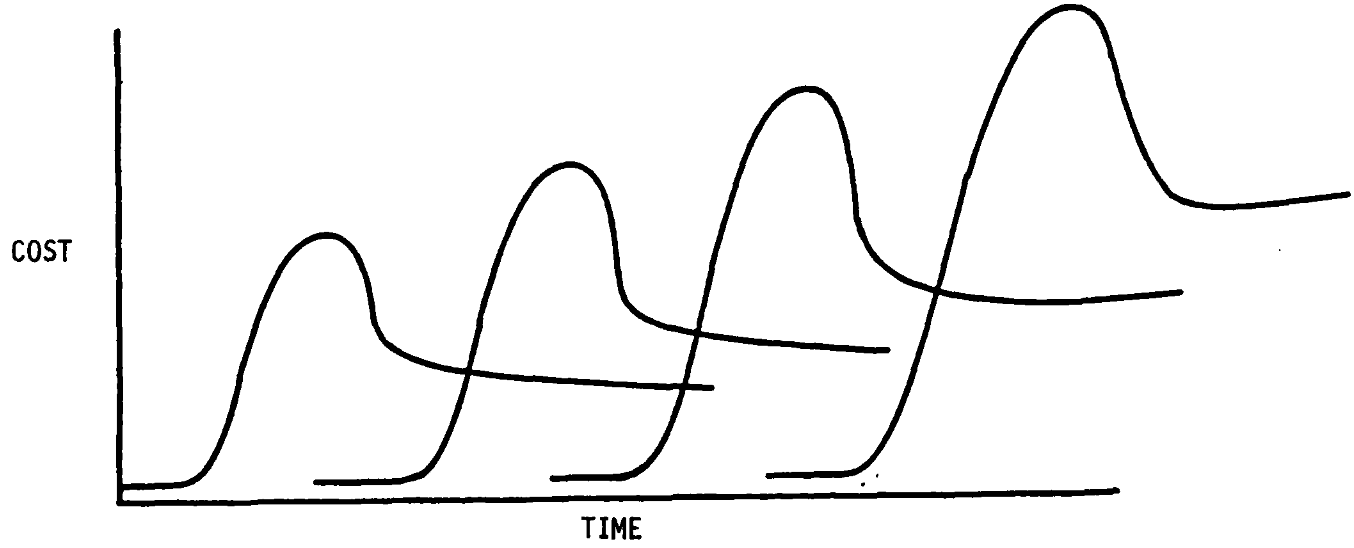


FIGURE V-4. SOFTWARE LIFE CYCLE EXPENDITURES UNDER ITERATIVE TRADITIONAL CYCLES

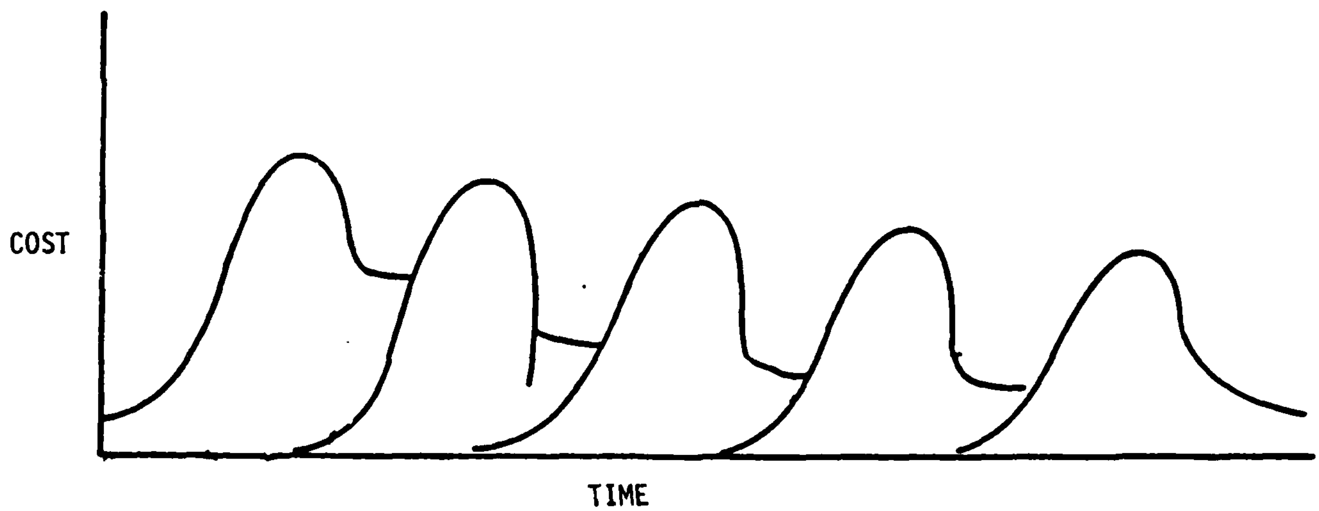


FIGURE V-5. SOFTWARE LIFE CYCLE EXPENDITURES UNDER EA SYSTEM DESIGNED TO ACCOMMODATE CHANGE

The reason that the initial development increment under EA is larger than the other increments (Figure V-5) is that: (1) some "up front" costs are associated with the development of the architecture, and (2) it includes some added costs of "excess" capacity and flexibility which are incorporated into the first increment.

1) Software Design Modularity

The key to achieving a decreasing cost profile like that shown in Figure V-5 is to develop a proper architectural framework within which both the initial "core" and subsequent increments are designed, and to design the software to accommodate change from the very beginning. The single most important factor in designing for change is modularity of design.

a) Modular Design, Not Just Structured Programming

Modularity has long been accepted as a desirable characteristic for software. However, as implemented, the emphasis has been placed almost exclusively on modular programming (closely coupled with Structured Programming techniques). Modular programming (coding) has been shown to result in higher coding productivity, fewer errors, lower testing costs, and lower maintenance costs. Modular programming is a beneficial approach which should be retained in Evolutionary Acquisition; but modular programming alone does not enforce modular design, which becomes of crucial importance in Evolutionary Acquisition.

Modular Design requires that all computer functions performing a single activity be grouped into a single module. Modular Programming then ensures that one or more code modules perform that design module, but that no code modules perform or affect any function that is not included in the design module. Modular Design further disciplines the access methods to each Design Module to control the software interactions.

Modular Design is essential for achieving the time and resource pattern of Figure V-5. With Modular Design, the additional requirements of each succeeding increment can be inserted into the existing design. Some requirements changes will cause the addition of design modules; many will cause only the revision of code within a single module. That is, with modular design, much of the evolution can take place as a subset of the 18 percent of cost devoted to coding (see Figure V-3). Without Modular Design, the development phase starts over with each increment.

Modular Design carries with it some additional significant long-term benefits. Modular Design allows for technology insertion on a "plug in" basis. Modular Design within each C^2 system will allow for eventual common modules (or firmware module replacements) across C^2 systems. Modular Design will increase the feasibility of using commercial software in C^2 systems to reduce acquisition time and development risks. Design Modules, discrete units which perform the totality of a Function, allow management visibility into the software, because they represent concrete building blocks of the system, rather than abstract concepts.

b) Management Visibility on Design Rather Than on Code

The enforcement of Modular Design should become a software design requirement for Evolutionary Acquisition. A second key change, which is itself desirable, should naturally occur in the enforcement process. The software design itself should receive increased visibility and be treated as a product or end item. Currently, the software design usually is visible only to system developers, with resulting emphasis on the executable code, which is visible (in its functionability) to all. Under Evolutionary Acquisition, the code is inappropriate as the visible product because it will exist in a state of constant, controlled change. Emphasis on the design will allow management monitoring of a relatively stable baseline, while allowing the system developers to modify the code easily as needed.

c) Enforcing Modular Design

Modular Design should be encouraged and enforced via two processes:

- For all Evolutionary C^2 systems, a periodic design review should be held with an organization external to the C^2 System Program Office as a check and balance against the natural tendency to trade off long-term benefits for short-term results.
- Management reporting for costs, performance, and risk assessments should be performed at the software design module level. In this manner, traceability and accountability can be added to software development with concurrent improvements in communication.

There are additional approaches for software development under Evolutionary Acquisition which will achieve efficiencies. A current requirement is for the use of DoD-approved High Order Language unless a waiver is granted. High Order Language usage will be of increased importance under Evolutionary Acquisition with the eventual use of Ada to achieve great economies. Waivers granted to the use of High Order Language should be granted only at the Design Module level, and only after technical need is firmly established.

b. Hardware Issues

Several areas of hardware technology advances enhance the ability to architect for and implement evolutionary systems.

An evolvable system requires an ability to accommodate change and growth. Today, most change (in the short run) is achieved by software and most growth is achieved by hardware. Growth has three major aspects:

- Size (processing power or input/output),
- Connectivity to other systems, and
- Functionality (different operator/system interface).

Processor family upgrades, to increase processing power and throughput, are becoming more feasible as most computer manufacturers develop upward-compatible lines of hardware which can execute software which was developed on older or smaller machines. The IBM legacy from 370 to 3080 lines are examples of older-to-newer, while within the 3080 lines there are numerous possible upgrades which can be accomplished in a short time, in the field, to increase the processing power of the machine. This type of upgrade capability, common to most modern processor lines, offers enormous evolutionary possibilities.

E. MANAGEMENT CONSIDERATIONS IN APPLYING EA

The evolutionary acquisition of C^2 systems requires that special consideration be given to those aspects of technical management which control the dynamics of system change, specifically change with respect to systems requirements and configuration.

1. Evolutionary System Design Requirements Management

The management of system design requirements analysis and specifications for an evolutionary program must ensure a suitable basis for the incorporation of change. Changes may involve retrofit to an existing "core" capability, may be expansion of a current "core" or may involve a new capability added to but required to interface with the existing system. The causes and sources of these changes were discussed in the previous section.

EA requirements management is further complicated by the expected continual nature of C^2 system upgrades. Unfortunately, C^2 system needs will not stand still while a group of designers define and developers build either a "core" capability or a subsequent increment of a given system. Each component or functional element of a C^2 system needs to progress at its own schedule. Various portions of the "system" are at differing degrees of completion at any given time. Any given function may be in the process of development under either evolutionary or traditional methods. It is in this world of change and varying schedules that requirements management must take place.

The key to requirements management is the ability of the developed system to meet operational objectives (mission-related) and therefore is directly linked to the satisfaction of the user. The ability to manage requirements is the responsibility of the user in conjunction with the provider. The means for effecting this management will be reflected in the usage of proper system design and development techniques such as proper configuration management, modular design and development, and proper interface management. These proper design and development techniques are even more important in an EA, which exists in a dynamic environment.

2. Configuration Management (CM)

CM is the discipline of identifying the configuration of a system at discrete points in time for purposes of systematically controlling changes to this configuration and maintaining the integrity and traceability of this configuration throughout the system life cycle. Proper configuration management is crucial to successful evolutionary acquisition. Hence, as the "core" and each subsequent increment completes development, each must go under CM to minimize "breakage" and maximize compatibility with subsequent increments.

The primary objective of CM is the effective management of a system's evolving configuration. A concept fundamental to this management process is that of a baseline. A baseline is a reference point or plateau in the development of a system, and is established by Government review and acceptance of a baseline specification document. At any time in the system life cycle, all of the previously established baselines constitute the contractual identification of the system and its configuration items.

Software CM covers the four basic areas briefly described below:

- Configuration Identification - The description, or identification of the software system, increasing in detail as the system evolves (i.e., baseline specifications on design modules and interfaces among them, as well as support documents).

- Control - The process by which changes to Government-approved baseline items, and to internally-baselined items (design modules) are initiated, classified, evaluated, approved, documented, implemented, and verified. The items include software products, performance and functional requirements, test products and parameters, drivers and results, and internal and external software interface formats and operational requirements.

- Configuration Status Accounting - The recording and reporting of all the above-mentioned baselined items of software, and the updates to these items as the system develops.

- Configuration Audits - The audits conducted by Quality Assurance personnel to validate the performance of software requirements and to establish the end product. The Functional Configuration Audit is a formal examination of the software system, prior to acceptance, which verifies that the software has achieved the performance specified in the system specification. For the particular evolutionary increment being evaluated, the Physical Configuration Audit is a formal examination of the as-built configuration of the modules of the software system against its technical documentation in order to establish the software's final product configuration identification.

3. Interface Management

A crucial role in the implementation of any system is the definitization, monitoring, and management of the internal and external system interfaces. Interfaces may be defined as any demarcation line across which data must pass, regardless of the methodology of data transmission. The various methodologies may include digital transmission via electrical cables, facsimile data transmission, voice transmission, the activation of signal lamps, or the presentation of graphic or alphanumeric displays to the data recipient.

Regardless of the transmission technique, encoding and decoding algorithms, or the transmitting technology, it remains imperative that the transmitting and receiving entities both operate from a common, clear and concise document which describes the operating characteristics and data parameters of the interface, such that both parties can independently construct receiving and transmitting devices which, when integrated, will permit the accurate exchange of intelligence over the interface. Such a philosophy is apropos for all interfaces, both internal and external to any system.

Traditionally, the definitization and documentation of interfaces in a systematic manner has been reserved for those which are external to the system or subsystem. The terms "system" and "subsystem" have been oriented more towards contractual entities than the true isolation of functionally complete units. This approach creates a problem for evolutionary acquisition, since the formal documentation of interfaces between architectural components at a consistent level of any given system usually is non-existent, particularly if the components were procured by a single contractor and were considered internal interfaces during development. The absence of complete, formal interface specifications between system architectural components necessitates the reversion to hardware and software specifications for a full understanding of the internal data transfers within a system. Although usually adequate for development personnel, this situation inhibits an ability for future system enhancement.

The requirement to define accurately and monitor closely the implementation of intersystem interfaces has led to the development and current use of an extensive range of documents and procedures which have been found to be helpful, including:

- Documents which define the responsibilities of the various interfacing parties,

- Drawings which detail the information to be transferred and the methodology of its transfer, and

- Procedures which delineate the techniques and processes for baselining and/or changing the drawings.

F. CONCLUSIONS AND RECOMMENDATIONS

A major concern of the Study Team was that the EA concept and objective of quickly fielding an initial "core" capability might be pursued while ignoring the architectural context, design approach, and development practices discussed in this section. This would be nothing short of a blueprint for disaster, resulting in systems which would become obsolete rapidly and be costly to remedy.

Accordingly, the Study Team recommends that programmatic approval for C^2 systems (and major enhancements) be based, among other considerations, upon the existence and adequacy of the following:

- A statement regarding how the C^2 system fits into its theater/mission context,

- A systems architecture which embodies the concepts of the ISO's OSI, and

- A Rapid Requirement Definition Capability (RRDC) which evolves into a System Design/Support Facility (SDSF).

- Proper management techniques, such as: High-Order Language programming; Software Design Modularity; rigorous software quality assurance; strong, centralized Configuration Management and Interface Management; and wider use of commercially-developed software and support services.

1. Theater/Mission Architecture

Any C^2 system must be viewed in the context of an overall theater/mission architecture. While such architectural definition work can (and must) proceed independently of the system acquisition strategy selected, the Study Team strongly affirms the need to proceed with such efforts with highest priority.

2. System Architecture

DoD represents about 6 to 7 percent of the ADP marketplace. Since the other 90-plus percent of the ADP marketplace is evolving in the direction of the internationally-accepted ISO open system interconnect (OSI) model, it would be prudent for DoD to move in this direction as well, rather than trying to invent its own model(s) (or to use none). The Team recognizes that there are potential problems with the ISO model, such as multi-level security and internetting issues, and that these will have to be worked. Even so, the Team believes DoD should give strong consideration to the cost and flexibility implications of inventing its own model (e.g., acceptance and implementation of standards) or having no model versus adopting a system for which there is going to be widespread software and services. Adoption of a layered architectural model should enable DoD-wide network standards, and should facilitate enforcing software modularity in the design phase, both of which will increase compatibility for accommodating growth.

3. RRDC/SDSF

There should be a Rapid Requirements Definition Capability established at the outset which could evolve into a permanent System Design/Support Facility jointly operated by provider and user. Such a facility will serve to facilitate early definition of evolving requirements, "post-deployment" software support of current increments, and be a tool for defining and analyzing the requirements and proposed design approach for future increments.

4. Other Recommendations

a. Exploit Commercially-Developed and Maintained Capabilities

The Study Team also recommends that DoD strive to better exploit commercially-developed and maintained capabilities which are under continuous "marketplace" testing, improvement and enhancement, rather than seek to develop its own standards (unless a valid military reason justifies the kind of significant investment, delay, and lack of flexibility which will be incurred).

b. Enforce Use of High-Order-Language Programming

High-Order-Language programming or the equivalent should be enforced to ensure software flexibility and application software transportability. While Ada is not the only solution, the Study Team is especially pleased with the accomplishments to date in Ada and is hopeful that Ada will solve a number of the problems that existed in past computer programming languages and will be able to be genuinely adopted as a DoD standard and find wider application.

c. Implement Software Design Modularity

The single most important factor in designing for change is design modularity. Modular programming (coding) has been shown to result in higher productivity, fewer errors and lower testing and maintenance costs. Modular design, which requires that all computer functions performing a single activity be grouped into a single module, allows the additional requirements of each succeeding increment to be inserted into the existing design without starting over with each increment, and allows for technology insertion on a "plug-in" basis. Modular design of software minimizes the probability of introducing failure or spurious behavior in other parts of the system, and hence should be enforced under EA.

d. Insist on Rigorous Software Quality Assurance

DoD should insist on rigorous software quality assurance throughout the evolution. The software quality assurance should ensure the existence of understandable documentation to support maintenance. Some on the Study Team advocated independent verification, to assure the software is "bug free." This proved to be controversial within the Team, with those from prime system contractors generally stating that independent verification was costly (5%-25% of software acquisition cost) and not worth the investment. Some believed that software quality could be assured if the program office is staffed with a small, technically-competent software management team to monitor the contractor's software planning and documentation, and quality-checking progress against milestones--without incurring the cost of a separate verification activity. The experience of those on the Study Team from the professional services community (who are paid to perform IV & V) and the experience of GEN Hilsman, is that even with good software people in the program office, software that is not independently verified is, more often than not, of unacceptable quality.

[illegible]. Use Centralized Configuration and Integration Management

Centralized configuration and integration management throughout the system life cycle also is critical. For EA, control during the "core" (or increment) development should be a program office responsibility. Changes to completed increments derived from user "hands-on" experience should be accomplished along with acquisition of the next increment. Continuing centralized CM of system-level code, utilities, and centralized applications programs is essential. However, system-level tools should be provided to facilitate development of site-unique applications within the centralized CM system.

APPENDICES

A. RECOMMENDED REVISIONS TO DoDI 5000.2 C^2 SYSTEM ACQUISITION POLICY

B. DRAFT IMPLEMENTATION MEMORANDUM

C. AFCEA ASSESSMENT OF APRIL 1982 PROPOSED OSD REVISION TO DoDI 5000.2 C^2 SYSTEM ACQUISITION POLICY

D. CURRENT (19 MARCH 1980) DoDI 5000.2 C^2 SYSTEM ACQUISITION POLICY

E. CASE SUMMARIES

F. ABBREVIATIONS AND TERMS

G. THE ROLE OF EVALUATION IN THE C^2 SYSTEM ACQUISITION PROCESS

H. INFORMATION SYSTEM ARCHITECTURE IN COMMAND AND CONTROL

I. DEFENSE ACQUISITION MANAGEMENT ORGANIZATION

APPENDIX A

RECOMMENDED REVISIONS TO DODI 5000.2
C^2 SYSTEM ACQUISITION POLICY

APPENDIX A

This Appendix represents the AFCEA Study Team's proposed rewrite of the DoDI 5000.2 policy on C^2 system acquisition (listed in the April 1982 "For Coordination" version as Section 27).

The Study Team is convinced that separate acquisition policy for C^2 systems is required. Based on our interactions with the participants in the C^2 system acquisition process (e.g., requirements validation/approval, budgeting, contracting, "ilities," T&E, ILS), it is clear that, if no such policy is forthcoming from OSD, little will be done in the DoD Components to alter present practices so that EA is facilitated for C^2 systems.

27. Command & Control (C^2) Systems

a. Although normal acquisition procedures are appropriate for many C^2 systems (e.g., for sensors and communications), the types of systems which involve or augment the decision-making and decision-executing functions of operational commanders and their staffs in the performance of their command and control functions, require a different and more flexible acquisition strategy, and related special management procedures, to deal with the unique aspects of such systems. Principal unique aspects are: (1) the acquisition cost of these systems normally is software dominated and the product of the application software development is highly interactive with the cognitive processes of specific mission users and is highly dependent on, and subject to change depending on, the specific doctrine, procedures, threat, geographic constraints, mission scenarios and management approach of specific (or classes of) mission users; (2) these systems are characterized by complex and rapidly changing internal and external interfaces at multiple organizational levels, many of which are inter-Service and multi-national; and (3) the operational requirements, acceptance criteria and measures of worth of these types of C^2 systems cannot be articulated and quantified adequately in advance. (Also included in these types of C^2 systems are those which constitute automated management information or intelligence information/exploitation and management/force planning and control aids).

b. Unless specific justification is provided to the contrary, these types of C^2 systems shall be acquired in an evolutionary manner, by evolutionary acquisition (EA). EA is an adaptive, incremental approach wherein only a quickly-fieldable "core," representing a useful increment in operational capability, is acquired initially based on abbreviated and expedited need. This "core" is defined within: (1) a representative description of the overall capability desired, including desired functional characteristics; (2) an architectural framework where evolution can occur with minimum subsequent redesign (i.e., within a layered architectural model that facilitates establishment of inter-connect and protocol standards); and (3) the context of a plan for evolution towards an ultimate capability.

c. Therefore, programming, budget approval and acquisition management procedures shall be tailored to encourage and enable early implementation and field evaluation of a basic or "core" system that represents a useful increment in operational capability with subsequent increments based on continuing feedback from testing in the user environment, evaluation, operational usage and, in some cases, application of new technology. The EA strategy should be tailored in each case, because there can be a spectrum of possible program circumstances (e.g., increment size, quantity of systems, echelon(s) of application, required manner of user involvement). To facilitate early field implementation of readily reconfigurable capability, detailed required operational capability and analytical justification documentation may be waived initially, and simplified expedited procedures such as letter requirements substituted, with operational and interface requirements and operational utility criteria evolved with the participation of actual mission users (or lead user and appropriate user surrogate for multi-user systems) in regular and continual interaction with developers, independent testers and logisticians regarding what is feasible and appropriate. Since a basic purpose of the evolutionary approach is to field useful blocks of capability as quickly as possible, expedited solicitation and award techniques should be used in procurement, both for the "core" and for subsequent increments, based on an advanced procurement plan which establishes how competition will be retained in the program. Selection criteria shall emphasize problem understanding, soundness of technical approach, and an architecture that facilitates both growth and introduction of subsequent increments with minimum redesign. Cost realism and contractor past performance shall govern as opposed to offeror's bid price.

d. The mission user (or a lead user and an appropriate user surrogate) will assume a major responsibility for the Demonstration and Validation phase, taking a dominant role in definition of the "core" capability, and a major continuing role throughout the entire development and acquisition process. Additionally, using flexible "test beds" of a type which range all the way from permanent system design and support facilities, to ad hoc capabilities which merely demonstrate the feasibility and value in an operational environment of various commercial configurations, the user shall work jointly with developers and independent testers to evaluate needs, concepts, the "core" and subsequent increments, and potential applications of new technology, testing various configurations in an operational environment. The user also will play a dominant role, working jointly with the independent tester in determining readiness for operational use of the "core" system and subsequent increments. The permanent system design facility shall also be used to accomplish post deployment software support of fielded increments under centralized configuration management. Normally, the DoD Component shall recommend in the acquisition strategy that the Concept Exploration Phase be combined with the Demonstration and Validation phase. The end result of combining these phases shall be a definition of a hierarchically-interoperable command and control system, including validated software specifications tailored to meet the mission user's needs, hardware specifications when needed, and the documentation necessary for operational employment. When this level of definition has been achieved, the DoD Component shall normally recommend that the system be procured in sufficient numbers for user-wide fielding. In other cases, the DoD Component may decide to use the results of the test to initiate a Full-Scale Development Phase. Consideration will be given to the use of commercial equipment, related system software and firmware, and contractor maintenance (with warranties) whenever logistic and interoperability considerations as well as field conditions permit it (especially when expected ultimate procurement quantities of a system are low).

e. The procedures described above are equally applicable to those non-major command and control systems of the type described above.

f.[1/] Those elements of command and control systems which must survive and endure in strategic or theatre nuclear warfare shall be as survivable as the weapon systems they directly or indirectly support. A proper mix of survivability techniques must be applied. Existing military and commercial hardware, software, and procedures should be used only if it can be shown that they can be protected against and made resistant to wide-area threats such as jamming, spoofing and electromagnetic pulse and that they can provide reasonable functional/system/path redundancy against direct attack, sabotage, etc. Interoperability and battlefield sustainability will be key considerations.

1/ Section f. is not a product (or focus) of the AFCEA Study Team's work, but is included herein for convenience, since it appears in all of the recent OSD re-drafts of the policy statement.

APPENDIX B

DRAFT IMPLEMENTATION MEMORANDUM

APPENDIX B - DRAFT IMPLEMENTATION MEMORANDUM

This Appendix is the Study Team's recommended means for notifying appropriate parts of the DoD Community that the DoD Acquisition Executive recognizes the special management procedures (and practices) required to facilitate C^2 system acquisition and desires that the DoD community take appropriate action to assure implementation and effective practice of evolutionary acquisition for C^2 systems that fit the criteria established in this report.

APPENDIX B: DRAFT IMPLEMENTATION MEMORANDUM

MEMORANDUM FOR DISTRIBUTION

SUBJECT: Command and Control System Acquisition--ACTION Memorandum

REFERENCE: DoD Instruction 5000.2 "Major System Acquisition Procedures"--19 March 1980

The reference notes that the characteristics of certain types of command and control (C^2) systems are sufficiently different from weapon systems that these types should be acquired, in most cases, via an evolutionary approach involving special management procedures, rather than the traditional approach. The specific C^2 system types that most require an evolutionary acquisition strategy are those which augment the decision-making and decision-executing activities of operational commanders and their staffs, including those which constitute automated management information or intelligence information/exploitation and management/force planning and control aids.

These types of systems: (1) have numerous complex and changing external and internal interfaces, often of an inter-Service and multi-national nature; (2) involve operational requirements, user acceptance criteria and measures of worth which cannot adequately be specified and quantified in advance, and (3) are software dominated, with the software highly interactive with the cognitive processes of specific mission commanders and their staffs at multiple organizational levels.

For some time, we have been reviewing the degree of implementation of the evolutionary acquisition policy for C^2 systems, and have found that its application is spotty. One reason for this is that the concept of evolutionary acquisition is not well understood.

Based on review of a number of past and present C^2 system acquisitions, it is clear that evolutionary acquisition gives a much higher probability that useful improvements in C^2 capability will be fielded sooner and more often. It also is clear that evolutionary acquisition will not work on a "business- as-usual" basis. Organizations and personnel involved with C^2 system requirements determination/validation, planning, programming and budgeting, contracting, the "ilities," development, test and support must recognize that evolutionary acquisition is a different, but necessary, strategy for these types of C^2 systems. Among the most prominent relational change required for successful C^2 system acquisition is the need for continuous interaction among users, developers, testers, and supporters, rather than the more "arms length" approach often used for weapon systems.

Finally, and perhaps most importantly, it is essential that C^2 system acquisition must proceed within an architectural framework that allows flexibility to facilitate growth and insertion of new technology with minimum redesign of the existing system.

In view of foregoing, I am taking the following actions:

(1) DoDI 5000.2 will be revised to mandate evolutionary acquisition as policy and clarify its use for these types of C^2 systems. (ACTION: DUSD/AM, working with other addressees.)

(2) An intra-DoD task force will be established to prepare and issue a guide to amplify evolutionary acquisition policy. (ACTION: DUSD/AM and DUSD/C^3I, working with Chairman JCS, MILDEPs, Director DCA, and other addressees, as necessary.)

(3) DoD procurement policy and practices will be revised to reflect the special needs of these types of C^2 systems. (ACTION: DUSD/AM, working with other addressees.)

(4) A program will be established to educate all participants in the C^2 system acquisition process on the merits and basic tenets of evolutionary acquisition. (ACTION: DUSD/AM, working with DUSD/C^3I and other addressees.)

(5) The approach to testing and evaluating these types of C^2 systems will be changed to assure joint user/developer/supporter/tester T&E, in the user's environment, and joint user/tester determination of operational utility. (ACTION: DUSD/T&E working with DUSD/C^3I, Chairman, JCS; MILDEPs.)

(6) Strong action will be taken to adopt a layered systems inter-connect reference model within DoD, to enable establishing interconnect and protocol standards that will facilitate growth and insertion of new technology with minimum redesign. The effort must recognize that the ISO Open Systems Interconnect Reference Model has been adopted by NATO and by the world wide commercial ADP industry. Also, DoD Components must expedite efforts to establish theater and other operational mission architectures, within which individual C^2 systems can evolve. (ACTION: DUSD/C^3I working with Chairman, JCS; CINCs; MILDEPs; and Director, DCA.)

The ASD Comptroller is requested to review and revise, as appropriate, OSD and DoD Component PPBS policy and practice, to reflect the special approaches required in planning, programming, and budgeting for these types of C^2 systems. (ACTION: ASD (Comptroller) working with other addressees.)

The Chairman, JCS, working with the CINCs and MILDEPs, is requested to take the following actions:

(1) Change the current approach to the requirements determination/validation process for these types of C^2 systems to abbreviate and expedite the process to recognize that this process is continuous for these types, and that feedback from testing in the user environment is the primary means of refining and amplifying requirements and evolving to the needed capability. Revise appropriate policy/regulations. (ACTION: Chairman, JCS; CINCs; and MILDEPs, working with DUSD/C^3I, USD(P) and other addressees, as necessary.)

(2) Assure substantially increased (continuous) real user involvement (or lead user, coupled with user surrogate(s) for multi-user systems) throughout the acquisition of these types of C^2 systems and assure appropriate resources are provided to real/lead users to support this involvement. (ACTION: Chairman, JCS; MILDEPs, working with DUSD/C^3I.)

The DoD Components are requested to take strong actions to assure use of improved development practices which facilitate accommodating change, for these types of C^2 systems, to include: use of already-developed system software; high-order-language programming; transportable, modular application software; rigorous software quality assurance; centralized configuration and integration management; and permanent system design/support facilities, jointly operated by users, developers, supporters, and testers. (ACTION: MILDEPs and Director, DCA, working with Chairman, JCS, CINCs and DUSD/C^3I.)

The Deputy Under Secretary of Defense (C^3I) is requested to provide me with a monthly memorandum report of implementation status of these action items; starting sixty (60) days from the date of this memorandum.

Signed

Under Secretary of Defense Research & Engineering

DISTRIBUTION
Secretary of the Army
Secretary of the Navy
Secretary of the Air Force
Chairman, Joint Chiefs of Staff
Unified & Specified Commanders (CINCs)
USD (Policy)
ASD (Comptroller)
ASD (Manpower, Reserve Affairs, and Logistics)
DUSD (Acquisition Management)
DUSD (C^3I)
DUSD (Test & Evaluation)
Director, Program Analysis and Evaluation
Director, Defense Communications Agency (DCA)

APPENDIX C

AFCEA ASSESSMENT OF APRIL 1982 PROPOSED OSD REVISION TO DODI 5000.2 C² SYSTEM ACQUISITION POLICY

APPENDIX C

ASSESSMENT OF APRIL 1982 PROPOSED OSD REVISION TO DoDI 5000.2

This Appendix reproduces the draft C^2 system acquisition policy in the "For Coordination" draft of DoDI 5000.2 of 12 April 1982.

Also included are AFCEA's comments on this draft, proposed revisions to the draft policy statement and rationale for the revisions.

APPENDIX C - ASSESSMENT OF APRIL 1982 PROPOSED OSD REVISION TO 5000.2

Quoted below is the C^2 system acquisition policy in the 12 April 1982 "For Coordination" version of DoDI 5000.2.

"27. Command and Control (C^2) Systems

a. The types of systems that augment the decision-making and decision- executing functions of operational commanders and their staffs in the performance of C^2 require a tailored acquisition strategy. The principal characteristics of such systems are: (1) acquisition cost normally is software dominated; (2) the system is highly interactive with the actual mission users and is highly dependent on the specific doctrine, procedures, threat, geographic constraints, and mission scenarios of these users; and (3) these systems are characterized by complex and frequently changing internal and external interfaces at multiple organizational levels, some of which may be inter-Service and multi-national.

b. The use of P^3I is a procedure highly appropriate to systems and should be considered when appropriate. This is an adaptive, incremental approach where an initial, relatively quickly fieldable "core" (an essential increment in operational capability) is acquired initially. This approach includes: (1) a description of the overall capability desired; (2) an architectural framework where evolution can occur with minimum subsequent redesign; and (3) a plan for evolution that leads towards the desired capability.

c. Programming, budget approval, and acquisition management shall be tailored to encourage and enable early implementation and field evaluation of a "core" system. Subsequent increments must be based on continuing feedback from operational use, testing in the operational environment, evaluation and (in some cases) application of new technology. Operational and interface requirements and operational utility criteria should be evolved with the participation of actual mission users (or lead user and appropriate user surrogate for multi-user systems). There must be regular and continual interaction with developers, independent testers, and logisticians.

d. The user shall support the independent T&E agency in determining readiness for operational use of the "core" system and work closely with the development activity and independent tester in evaluating subsequent increments of new technology. A centralized facility shall be used to accomplish post deployment software support of fielded increments under centralized configuration management. Consideration shall be given to the use of existing commercial equipment, related system software and firmware, and contractor maintenance (with warranties) whenever logistic, interoperability, readiness considerations, and field conditions permit it.

e. Those elements of C² systems that must survive and endure in strategic or theater nuclear warfare shall be at least as survivable as the weapon system they directly or indirectly support. A proper mix of survivability techniques must be applied. Existing military and commercial hardware, software, and procedures should be used only if it can be demonstrated that they can be protected against and made resistant to wide-area threats such as jamming, spoofing and electromagnetic pulse, and that they can provide reasonable functional/system/path redundancy against direct attack and sabotage. Interoperability and battlefield sustainability will be key considerations.

f. The procedures described above are equally applicable to similar non-major C² systems and control systems, as well as counter-C^3, electromagnetic countermeasures, and electronic warfare systems."

- - -

AFCEA cannot endorse this policy statement as now written. The proposed policy statement is a major step backwards from the March 1980 version of DoDI 5000.2 C² system acquisition policy, let alone what AFCEA believes should be in the present version, based on the results of our study.

A few of the more significant inconsistencies between the 12 April proposed version of DoDI 5000.2 and the Conclusions and Recommendations of our study are that the 12 April version: (1) does not require use of EA as the primary strategy for acquiring decision-aiding C² systems (nor even encourage it in appropriate circumstances); (2) significantly diminishes the influence of the "real" (mission) user in C² system acquisition, and (3) deletes reference to the need for DoD-wide adoption of a layered architectural model to facilitate the establishment of interconnect and protocol standards.

Appendix A to this report is AFCEA's proposal for inclusion in DoDI 5000.2 as C² system acquisition policy. If this version cannot be adopted by DoD for use in DoDI 5000.2, we urge that, as a minimum, the changes listed below be incorporated. The Study Team believes the changes we propose are crucial to obtaining improvements in C² system acquisition.

<u>AFCEA Comments on DoDI 5000.2 Encl 2, Sect. 27</u>

1. <u>Sub-section a.</u> Revise third line to read: "...the performance of C² require a <u>different and</u> more flexible acquisition strategy and related tailored management procedures, than that ordinarily employed, viz, evolutionary acquisition (EA). The..."

RATIONALE: The first sentence now states that C² systems "...require a tailored acquisition strategy." But DoD 5000.1 correctly indicates, on both pages 2 and 7, that <u>all</u> systems require a tailored acquisition strategy. Therefore, the opening sentence of Section 27, as now written, omits the important justification for why a different acquisition strategy is appropriate.

2. <u>Sub-section b</u> - Delete first sentence and replace it with: "...Unless specific justification is provided to the contrary, these types of C^2 systems shall be acquired in an evolutionary manner--by evolutionary acquisition (EA)."

RATIONALE: From a policy standpoint, the first sentence here is the operative sentence in the clause, making all that follows of little significance unless the first sentence is strengthened. As written, the first sentence is even weaker than the March 1980 version, which at least required that the design and testing of C^2 systems be accomplished under EA "in most cases." In contrast, the currently-proposed version neither references EA nor states or provides criteria for when EA is appropriate.

If the "burden-of-proof" language the Study Team proposes is used, one might also wish to use more specific language in Sub-section a. about just what kinds of systems are intended to be covered. The systems the Study Team recommends that the clause cover are not only the decision-support systems already indicated, but also those which constitute automated management or intelligence information/exploitation and management/force planning and control aids, such as (in increasing order of the need for EA): weapon/platform control systems (e.g., JSS); intelligence information and exploitation systems; tactical battle-management automation systems; and top-level strategic force management systems. In contrast, the following types of C^3I systems ordinarily could be acquired in the normal way (although benefitting from the use of EA at times): ADP embedded in weapons, platforms, or in communications control elements; common-user communications systems; data links; and sensor systems of the stand-alone radar type or those used in fire control and navigation type applications. These excluded systems might most easily be handled in the policy simply by adding, as a lead-in to the first sentence of the policy: "a. Although normal acquisition procedures are appropriate for many C^2 systems (e.g., for sensors and communications)..." (See Appendix A).

If the above-recommended wording changes cannot be accepted, the Study Team strongly recommends that the term "Evolutionary Acquisition (EA)" be substituted for "P^3I," or (and we favor this even less) provide an antecedent to the use of the word "evolution" twice later in this sub-section by changing the first sentence to use both terms, viz, "The use of P^3I (EA)..." Or, finally, if "P^3I" must stand alone, we recommend rewriting the first part of the second sentence, leaving out the word "This" and substituting the expression, "When P^3I is adapted to C^2 systems, it..." This would clarify that what is described subsequently in this sub-section as the way P^3I is practiced when applied to a C^2 system. Not only is P^3I covered already for <u>all</u> systems in 5000.1, but it is subsumed to evolutionary in 5000.1 by <u>being</u> made an example (i.e., P^3I definition deals only with the portion of the meaning of evolutionary which is the opposite of revolutionary, but does <u>not</u> deal with the part of the meaning of EA which deals with adaptive or heu<u>ristic</u> design).

3. Subsection b, line 5,6 - Revise to read: "...overall capability desired including desired functional characteristics; and add at the end of (2) and before (3): "...with minimum subsequent redesign including a layered architectural model that facilitates establishment of interconnect and protocol standards; and (3)...

RATIONALE: Change under (1) is clarification. Change under (2) arises from a (perhaps the) major conclusion of this study of C^2 system acquisition: that there is a potential for chaos if DoD does not adopt the use of a layered architectural model to accommodate readily, and with minimum redesign, both change and the insertion of new technology.

4. Sub-section c, 5th line Add as the third sentence "The EA strategy should be tailored to reflect the fact that there can be a spectrum of possible program circumstances (e.g., increment size, quantity of systems, echelon(s) of application, and required manner of user involvement)."

RATIONALE: As written, the 12 April version of 5000.2 speaks only of a core system (can be any useful increase in C^2 capability) and omits recognition of the need for tailoring the EA strategy to ensure the program flexibility required in C^2 acquisition.

5. Sub-section d, 1st sentence Revise to read "The mission user (or lead user and appropriate user surrogate for multi-user systems) shall play a major role with the independent T&E agency in determining..."

RATIONALE: The evidence, in case after case studied, has convinced the Study Team of the necessity for strong, continuous real/lead user participation in C^2 system developments. For multiple-user systems, assessment of the evidence also has convinced the Study Team that only mission user or user surrogate participation is not enough--strong, continuous participation in development and test by both a lead "real" user (i.e., a user who "faces" a potential enemy daily--e. g., theater forces in Europe or Korea, the RDJTF, deployed fleet forces, SAC Wings, NORAD) and a user surrogate (to represent all other potential users and to aid in resolving conflicts between "requirements" and resources) is crucial to maximizing the probability of program success.

6. Sub-section d, 4th line Insert after first sentence: "Additionally, using flexible "testbeds" or permanent system design facilities, the user shall work jointly with developers and independent testers to evaluate needs, concepts, the "core" and subsequent increments, and potential applications of new technology, testing various configurations in an operational environment."

RATIONALE: The need for testbeds or permanent system design facilities, which, incidentally, could also be used as the post-deployment software support facility called for in line 5, is a major recommendation of our study.

7. The words in Sub-section "e," are not a product (or focus) of the AFCEA Study Team. Similarly, regarding Sub-section "f," our analysis concentrated on C^2 (decision-support) systems, so we drew only inferential conclusions concerning application of EA to the other C^3I systems mentioned.

APRIL 1982 PROPOSED OSD REVISION TO 5000.2 AS AMENDED BY AFCEA

Quoted below is the C^2 system acquisition policy in the 12 April 1982 "For Coordination" version of DoDI 5000.2, as amended by the AFCEA comments given in the foregoing (amendments underlined):

"27. Command and Control (C^2) Systems

a. Although normal acquisition procedures are appropriate for many C^2 systems (e.g., for sensors and communications), the types of systems that involve or augment the decision-making and decision-executing functions of operational commanders and their staffs in the performance of C^2 require a different and more flexible acquisition strategy and related tailored management procedures, than that ordinarily employed, viz, evolutionary acquisition (EA). The principal characteristics of such systems are: (1) acquisition cost normally is software dominated; (2) the system is highly interactive with the actual mission users and is highly dependent on the specific doctrine, procedures, threat, geographic constraints, and mission scenarios of these users; and (3) these systems are characterized by complex and frequently changing internal and external interfaces at multiple organizational levels, some of which may be inter-Service and multi-national.

b. ~~The-use-of-P^{3}I-is-a-procedure-highly-appropriate-to-systems-and-should be-considered-when-appropriate~~. Unless specific justification is provided to the contrary, these types of C^2 systems shall be acquired in an evolutionary manner--by evolutionary acquisition (EA). This is an adaptive, incremental approach where an initial, relatively quickly fieldable "core" (an essential increment in operational capability) is acquired initially. This approach includes: (1) a description of the overall capability desired; including desired functional characteristics (2) an architectural framework where evolution can occur with minimum subsequent redesign; including a layered architectural model that facilitates establishment of interconnect and protocol standards), and (3) a plan for evolution that leads towards the desired capability.

c. Programming, budget approval, and acquisition management shall be tailored to encourage and enable early implementation and field evaluation of a "core" system. Subsequent increments must be based on continuing feedback from operational use, testing in the operational environment, evaluation and (in some cases) application of new technology. The EA strategy must be tailored because it can cover a wide range of possible program circumstances (e.g., increment size, quantity of systems, echelon(s) of application, and required manner of user involvement). Operational and interface requirements and operational utility criteria should be evolved with the participation of actual mission users (or lead user and appropriate user surrogate for multi-user systems). There must be regular and continual interaction with developers, independent testers, and logisticians.

d. The mission user (or lead user and appropriate user surrogate for multi-user systems) should play a major role with ~~shall support~~ the independent T&E agency in determining readiness for operational use of the "core" system and work closely with the development activity and independent tester in evaluating subsequent increments of new technology. Additionally, using flexible "testbeds" or permanent system design facilities, the user shall work jointly with developers and independent testers to evaluate needs, concepts, the "core" and subsequent increments, and potential application of new technology, testing various configurations in an operational environment. A centralized facility shall be used to accomplish post deployment software support of fielded increments under centralized configuration management. Consideration shall be given to the use of existing commercial equipment, related system software and firmware, and contractor maintenance (with warranties) whenever logistic, interoperability, readiness considerations, and field conditions permit it.

e.[1] Those elements of C² systems that must survive and endure in strategic or theater nuclear warfare shall be at least as survivable as the weapon system they directly or indirectly support. A proper mix of survivability techniques must be applied. Existing military and commercial hardware, software, and procedures should be used only if it can be demonstrated that they can be protected against and made resistant to wide-area threats such as jamming, spoofing and electromagnetic pulse, and that they can provide reasonable functional/system/path redundancy against direct attack and sabotage. Interoperability and battlefield sustainability will be key considerations.

f.[1] The procedures described above are equally applicable to similar non-major C² systems and control systems, as well as counter-C³, electromagnetic countermeasures, and electronic warfare systems."

[1] Sections e. and f. are not a product (or focus) of the AFCEA Study Team's work, but are included herein for convenience, since they appear in the April 1982 OSD re-draft of the policy statement.

APPENDIX D

CURRENT (19 MARCH 1980) DODI 5000.2 C^2 SYSTEM ACQUISITION POLICY

DOD Instruction 5000.2 "Major System Acquisition Procedures" 3/19/80

13. Command and Control Systems

a. The major characteristics of command and control systems that require special management procedures are a rapidly evolving technological base, multiple requirements for internal and external interfaces, and reliance on automatic data processing hardware and related software. Such command and control systems differ from other weapon systems: they are acquired in small numbers, in some cases only one of a kind; their operational characteristics are largely determined by the users in an evolutionary process; and commercial equipment exists that can emulate the function. For command and control systems meeting the above criteria, acquisition management procedures should allow early implementation and field evaluation of a prototype system using existing commercial or military hardware and software.

b. Upon the recommendation of the appropriate using command, the DoD Component or the ASD(C^3I), an alternate acquisition procedure shall be presented for approval by the Secretary of Defense. Following the documentation of a command and control major system requirement in a MENS approved by the Secretary of Defense in a SDDM, the design and testing of such systems should, in most cases, be accomplished in an evolutionary manner. These command and control systems shall be configured initially as prototypes using existing military or commercial equipment to the maximum extent possible and with a minimum of additional software. The designated users should be tasked to test various configurations in an operational environment using prototype and laboratory or test bed equipment and to assume the major responsibility for the Demonstration and Validation phase. In these cases, it shall be necessary for the DoD Component to recommend in the MENS that the Concept Exploration phase be combined with the Demonstration and Validation phase. The end result of combining these phases shall be a definition of a command and control system, including operational software, tailored to meet the commander and user needs and the documentation necessary for operational employment. When these objectives are achieved, the DoD Component shall normally recommend that the system be procured in sufficient numbers for initial fielding. In other cases, the DoD Component may decide to use the results of the test bed to initiate a competitive Full-Scale Development phase.

c. The procedures described in this paragraph are equally applicable to those non-major command and control systems that meet the criteria described above. Developers of such systems should be encouraged to pursue these alternative procedures when appropriate.

APPENDIX E

CASE SUMMARIES

APPENDIX E - CASE SUMMARIES

Case studies were selected as one means of obtaining information about past and present C² system acquisitions. At least two Team members were assigned to research each case selected. In some cases, the study followed an acquisition through more than one iteration, under different names, over periods as long as 25 years.

Case research done by the small individual case teams were evaluated by the entire Study Team. Also, there were occasions when members other than a case team participated in discussions and interviews regarding specific cases.

The summaries in this Appendix briefly describe each case. Additional information about the case study methodology, characteristics of systems, acquisition experience, and lessons learned is included in Chapter I (Section E.2) and Chapter III (Sections B.4, C.1, and C.2).

1. BATTLEFIELD EXPLOITATION AND TARGET ACQUISITION (BETA)/JOINT TACTICAL FUSION PROJECT

The BETA project was initiated to provide a joint test bed to evolve an automated system that would correlate sensor data and integrate fragments of intelligence into a "picture" of the battlefield, to enable nomination of the most lucrative potential targets. Originally, it was planned that the BETA test beds would be exercised at three types of organizations--Army corps, Army division, and Tactical Air Force.

Development of the BETA test bed was started in 1977. The plan called for deployment of the test bed to Europe for user experimentation and subsequent evolution of the system. Commercial hardware and software were extensively used in building the system. Some real users were represented in the Joint Project Office. The very tight schedule (37 months for development through demonstration) was not met. Accordingly, at the critical milestone review, BETA was found inadequate for deployment, largely due to hardware reliability problems. Estimated cost of the project increased from $21.1 million to $48.3 million. The initial definition of the BETA "core" capability proved too ambitious for the desired schedule and had to be drastically reduced, post facto.

Although not yet deployed, the majority of the technical goals of the reduced-scope BETA (now called the Joint Tactical Fusion program) now have been met, using a centralized software development/support facility at Hurlburt Field, Florida. The Air Force plans to deploy a BETA test bed to Europe in 1982(Limited Operational Capability - Europe--LOCE) and evolve it into an operational test bed and later utilize knowledge gained from BETA in evolution of the Enemy Situation Correlation Element (ENSCE). The Air Force plans to retain a centralized software development/support facility at Hurlburt Field, Florida. While not planning to deploy BETA, the Army is operating a BETA test bed at the TRADOC Combined Arms Test Activity (TCATA) at Ft. Hood, Texas for user surrogate testing, and is reflecting some BETA capabilities in its requirements for the All Source Analysis System (ASAS).

Some Important Lessons

Schedules should be event driven and realistic. Development time was shortened by the use of some existing commercial hardware and software, but was prolonged by hardware and software problems with other components that had to be developed. The number of different programming languages imposed by using existing software also caused problems. Real user participation in the development has been invaluable, but could have been more extensive. The initial "core" definition was too comprehensive for the required schedule. A centralized development/support facility has been quite effective.

2. TACTICAL OPERATIONS SYSTEM (TOS)/SIGMA

In 1956, a study group in the Army was established to identify battlefield applications of computers. Two years later a project office was organized to develop an Army Tactical Operations Center (ARTOC), intended to be fielded in 7th Army. The ARTOC was assembled and delivered to Ft. Leavenworth in 1963, where it was tested for two years. In 1965 the Army started a new program to develop and field a test bed TOS in Europe, which was called EUROTOS. Contracts were let in 1966 and a system, using commercial components, was deployed to Europe in 1968. EUROTOS consisted of a Central Computing Center, four Remote Station Data Terminals and 18 User Input/Output Devices. The command and control functions supported were: friendly unit information, enemy situation, nuclear fire support, effects of enemy nuclear strikes, and enemy order of battle.

EUROTOS was sent to a "real user," 7th Army, along with support resources and used in exercises until 1970. Overall results were favorable, with 7th Army recommending further evolution of the system in Europe. Unfortunately, due to resource constraints caused by the Southeast Asian war, Headquarters, DA would not provide additional funding to 7th Army so EUROTOS was moved to III Corps at Ft. Hood, Texas where it withered.

In 1972 a TOS/Operable Segment (TOS/OS) project was started to develop a division-level TOS, mainly using existing hardware. Software development was mostly in-house and constrained by the hardware. Tests in 1977 revealed substantial software and system design problems. Also in 1977, CINCUSAREUR expressed an urgent operational requirement for a TOS, so development of a division level TOS (DIVTOS) continued. By 1979 the DSARC approved initiation of engineering development of DIVTOS. A GAO report in 1979 strongly criticized the DIVTOS program and was followed by a Congressional decision to eliminate DIVTOS funding. According to the GAO, at least $93.4 million had been spent on TOS and several major defects remained.

Following the demise of TOS, the SIGMA project, employing an evolutionary approach, was launched. Phase I employs the Tactical Computer System (TCS)/Tactical Computer Terminal (TCT)--by-products of the TOS program--as a "core" capability. This equipment is now installed and operating at several units in VII Corps for message transmission and handling. As the user identifies new requirements, the software is developed at a central software development/support facility at Ft. Leavenworth, Kansas and added in the field. Phase II SIGMA will add the Initial Maneuver Control System and is to add force-level control functions.

Some Important Lessons

EUROTOS, following an evolutionary approach, was fielded in about two years and performed successfully in Europe. When removed from the real user environment, interest in EUROTOS waned and the project was dropped. TOS/OS seriously suffered from use of an obsolete militarized computer not designed to accommodate growth that limited software flexibility. Acceleration of software development increased difficulties. After 26 years, the Army requirement for an automated TOS still has not been fulfilled. If support had been sustained, the EUROTOS system, which followed an evolutionary approach, probably would have provided a valuable capability many years ago. Also, a real user was not involved in the TOS/OS development, which probably contributed to challenges of the concept and effectiveness of the system. The present SIGMA is following many of the positive attributes of evolutionary acquisition.

3. TACTICAL FIRE DIRECTION (TACFIRE) SYSTEM

Although in 1959 the Army recognized the requirement for an automated fire control system, a specific statement of the requirement for TACFIRE was not approved until 1966. TACFIRE was to automate 12 field artillery functions, with computer centers at Division Artillery and firing battalions, along with input and output devices at other organizational elements.

The initial acquisition of TACFIRE was under a total package procurement contract for a total solution, which was awarded in 1967 with a ceiling of $122.3M. Numerous problems, both hardware and software, were encountered during development, with software the greater difficulty. Software efforts included development of a special new high-order language called TACPOL. An important factor in overcoming many problems was the designation of the Field Artillery Center, Ft. Sill, Oklahoma, as the "using agency" to assume responsibility for adequacy of TACFIRE for its fire mission role and to achieve a closer tie between the user representative and contractor. A large cadre of Artillery Center people were assigned to the Program Office and to the contractor's plant.

Full scale production of TACFIRE was authorized in 1978 and is to be completed in 1984. Since the involvement of the Artillery Center, the system has gained wide acceptance by the operational user community.

Some Important Lessons

Large delays and cost overruns may be attributed to difficulties in developing complex software in a large step function rather than in smaller increments. Though late in the program, close involvement of a knowledgeable and motivated user surrogate in the development was of significant benefit. Another lesson is that independent T&E can be more productive if it includes data gathering and fixes to make the system more useful, as well as more traditional "go no-go" checking. Finally, it is probable that development and deployment of a useful "core" capability could have cut 3-6 years from the 13 years (after contract award) for the first of the full systems to reach the field.

4. POSITION LOCATION REPORTING SYSTEM/JOINT TACTICAL INFORMATION DISTRIBUTION SYSTEM (PLRS/JTIDS HYBRID)

This is a computer-based system which provides real-time, jam-resistant, secure data communication, and position location and reporting information for tactical elements, mostly in Army divisions. A Friendly Situation (FRIENSIT) element is included and represents an important decision aid for tactical commanders and staffs. Plans call for fielding about 1,500 JTIDS and 16,000 PLRS terminals.

In 1978, a five-phase development and test program was launched, based on equipment and experience from the Army/Marine PLRS and Joint JTIDS programs. IOC is scheduled for 1986. Overall Hybrid requirements are expressed in a Letter of Agreement (LOA). Most system requirements have been specified, although flexibility in changing users and data transfer rates is covered in the program. Militarized computers and the military standard CMS-2 programming language are employed. Substantial software development is necessary and is expected to make up about 45% of the total program acquisition cost.

The acquisition strategy includes use of a test bed to allow the developer and user to gain experience with the system while evolving to the full-scale development model. Both real users and surrogates have participated in the program.

At the time of the case study, the program was about 3-6 months behind schedule, but technical results were favorable. The first two phases have been completed within budget. Total program costs have not been estimated, although cost through Phase V (testing) is expected to be about $115M.

Some Important Lessons

Early user involvement and availability of a test bed have been very helpful in shaping the program.

5. ORIGINAL TACTICAL FLAG COMMAND CENTER (TFCC)

The Tactical Flag Command Center (TFCC) is a shipboard command and control system. It is intended to provide the tactical commander at sea with information from on-shore and task force sources, pertaining to status of our forces and the location and intention of enemy forces. The system was planned for deployment aboard ships of the CV, CG, LCC, and CGSN configurations.

In 1972-1973, NAVELEX began preparation of an RFP for a Tactical Flag Command Center, using as a requirements basis, the results of a large number of analytical studies and employing a traditional acquisition approach. An interim TFCC (ITFCC) was evaluated aboard the USS JOHN F. KENNEDY using a Graphic Analysis Control Terminal (GACT). Results were neither positive nor conclusive. After a lengthy competition, a contract was awarded to develop TFCC. The estimated cost for an Initial Operating Capability (IOC) by all competitors was between $5M and $10M. When the two year design phase was completed, the IOC cost had risen to between $25M and $30M. Cost, schedule, and disagreement within the Navy all combined to cause rejection of the proposed development. After two years, CNO approved a development program but encouraged a speed-up.

NAVELEX proposed a two-phase development utilizing the AN/USQ-81(V) targeting system. The original approach was dropped in favor of this evolutionary approach.

Some Important Lessons

The primary lesson learned from the original TFCC program is that the conventional approach of lengthy analytical studies followed by a detailed specification procurement for a "total" solution, and then another lengthy paper design period, is not appropriate for acquiring a command and control system that must interact with battle commanders. The "final" system specifications must be established by the operational user as he operates a flexible test bed in his own environment.

6. CURRENT TACTICAL FLAG COMMAND CENTER (TFCC)/OUTLAW SHARK PROJECT

The evolutionary or current configuration of the TFCC is built around the AN/USQ-81(V), an over-the-horizon targeting (OTH/T) system developed in a series of "sustained concept testing" programs, starting with the OUTLAW SHARK program in the 1974-1975 timeframe.

After establishing that the original TFCC approach would not meet IOC requirements (including cost and schedule), NAVELEX initiated a two-phase evolutionary development utilizing the AN/USQ-81(V) targeting system as the basic building block. This system itself was the result of an evolutionary development starting in 1972-1973, wherein operational commanders interacted with flexible test beds in their operational environments to establish the system parameters and "specifications." Under the current TFCC plan, the AN/USQ-81(V) was to be deployed in the current configuration on several platforms as developmental test beds to refine requirements. The second phase was to reconfigure the AN/USQ-81(V) with existing equipments that were approved for service use and logistically supportable by the Navy. The thrust of the acceleration was that the AN/USQ-81(V) possessed sufficient capability to justify deployment and that this basic capability would be incrementally enhanced to incorporate the many functions necessary to support the tactical commander at sea. Evaluations of the performance of Engineering Development Models aboard the aircraft carriers MIDWAY and AMERICA generally have indicated that the engineering development models were satisfactory as test beds, but do not have the necessary command and control decision aids necessary to support the many missions of the embarked Flag staff. These reports have resulted in a compilation of system upgrades which will be necessary to incorporate in the baseline. CNO decided in 1981 to approve a limited procurement of six shipboard systems and two shore-based systems in a baseline configuration. All of these systems are to be installed by 1984. In addition, a parallel activity will be initiated to redesign the TFCC software and hardware using a High-Order Language (HOL) and Navy Standard Computer (UYK-43 or UYK-44). In 1984, a decision is to be made to continue procurement of the existing systems or to procure the new system with Navy standard computers.

Some Important Lessons

The primary lesson learned from the TFCC Program is that the command user must be directly involved in the definition of his command and control system via the hands-on use of a reliable, flexible "users test bed" in his operational environment. He needs this capability to validate his concept of operation and to gain the experience necessary so that he can define his specific operating requirements. Subsidiary lessons are that the test bed must use proven hardware and software and actually contribute to the commander's day-to-day mission during the test phase.

7. MARINE INTEGRATED FIRE AND AIR SUPPORT SYSTEM (MIFASS)

MIFASS is an automated tactical data system to aid a Marine Corps commander in controlling and coordinating air, naval gunfire, artillery, and mortar assets in support of ground maneuver forces. Users of MIFASS include echelons from Task Force headquarters down to mortar platoons.

MIFASS is a total-solution development, proceeded by a comprehensive test bed program, the purpose of which was to define detail system requirements. Requirements for MIFASS were developed in a non-operational test bed during 1972-1977. A ROC was approved in 1975. The test bed staff included user representatives. Also, personnel from user units have participated in all operational-type tests in the test bed. Standard commercial hardware and system software were used in the test bed, while considerable special software development was required.

It is planned that engineering development and production models will utilize the standard military computer AN/AYK-14 with its standard support software. Engineering development started in 1980 and is scheduled for completion in 1984, reflecting a one-year slip from the original schedule. Software problems have been the main cause of the delay. Estimated cost of engineering development is $40 million, about double the original contract bid. Recently, Hq USMC representatives have indicated that due to development problems, MIFASS is "backing into" an evolutionary approach post facto.

Some Lessons Learned

Non-operational test beds can be useful in evolving initial system requirements, however, considerable time (and possibly funds) could have been saved if operationally-acceptable hardware and, especially, software had been used in the test bed evolution. At this time, no prognosis can be made regarding user acceptance of the system to be delivered some time after 1984. It is likely that considerable time could have been saved by defining a "core" capability back in 1972 and fielding it for test and evolution in the user environment.

8. OPERATIONAL APPLICATION OF SPECIAL INTELLIGENCE SYSTEMS (OASIS)

This program was conceived in the mid-1970's as a way to improve US Air Force intelligence capabilities in the European Theater at the USAFE level. The primary functions of OASIS are: improving information handling and improving interfaces with external organizations. Automation is heavily employed in both functions.

Because requirements could not be defined fully at the beginning of the program, considerable evolutionary development of requirements and system elements was planned. Close working relationships were established between the Program Office and key players, including the real user, tester provider and others. USAFE, the real user, was tasked to define requirements. A system development facility was available at NSA. Early in the program more than 20 system enhancements were planned, each with an implementation schedule of 9-18 months, and with some to be made by the Program Office and some by the contractors. To avoid almost constant contract negotiations, enhancements were grouped into "work packages." Also, quick reaction capability (QRC) procedures were established to enable fast contractor response to critical needs.

During 1981 two enhancements were delivered to the theater. Work is underway to expand those and add other enhancements, all under configuration management. Contract costs from 1978 to 1985 are estimated to be about $32 million. Type of contract is cost plus award fee.

OASIS is operational at USAFE Headquarters.

Some Lessons Learned

Evolutionary acquisition can achieve rapid fielding of an operationally useful core system and expeditious upgrades. Multiple concurrent enhancements require intensive and strong management by the Program Office. The user must be closely involved with the developer in budgeting, as well as in other processes.

9. TACTICAL AIR CONTROL CENTER AUTOMATION (TACC AUTO)

This project was intended to provide automated assistance to the C^2 element of the USAF Mobile Tactical Air Control Systems (TACS). TACC AUTO was to be employed by Theater Air Commanders. The requirement for TACC AUTO was based on a ROC approved in 1967.

A traditional acquisition strategy was followed in the TACC AUTO project. A lengthy and detailed traditional "requirements process" proceeded selection of a prime contractor after competition. Delays in the project were caused by uncertain requirements (specifications), software development problems, funding perturbations, cost overruns, and disenchantment with ADP hardware deemed obsolete prior to completion of the TACC AUTO development. Although the system was judged a conditional success after testing, the serious problems encountered in the program caused Congress to terminate TACC AUTO. About $80 million had been spent on the development.

Some Lessons Learned

The absence of a strong user role throughout the program and the lack of flexibility to adapt to changing requirements were key factors in causing the program to fail. Difficulty in automating many functions, which under the traditional acquisition approach followed had to be done in one development cycle, resulted in prolonged delays and cost growth.

10. COMPUTER AIDED FORCE MANAGEMENT SYSTEM (CAFMS)

CAFMS' role is similar to TACC AUTO's except, reflecting experience from the TACC AUTO problems, a less-ambitions level of automation in the TACS is being provided. Requirements were defined by a TAC/ESD/MITRE working group. CAFMS is a user-led development, with "off-the-shelf" hardware bought competitively via O&M funds and the application software written by TAC/Data Automation people, based on the TACC AUTO software design.

The program was started in 1979, partially employing an evolutionary acquisition strategy. There were several false starts due to less-than-adequate bid packages. There has been intimate user involvement throughout the project (though the Hq TAC "user" really was a surrogate for the real users, 9th and 12th Air Force). Performance goals instead of specifications were used. TACC AUTO can be considered to have been a form of test bed for CAFMS. In 1981, CAFMS was formally tested and accepted for deployment to 9th and 12th Air Forces, who are pleased with the result. Cost of hardware for one CAFMS system is about $400K.

Some Lessons Learned

One program manager, rather than the fragmented approach taken, should be in charge. Systems Command should give contracting and technical support to the user, when the user manages a system of this type.

11. CONSTANT WATCH

The Constant Watch system, to be in a hardened facility in Korea, is an improved, automated system primarily to aid data handling for information about status of units and bases, aircrews and aircraft, targets, munitions, weather and mission. Requirements have not been firmly defined, but are expected to evolve.

The development program is a combined one by USAF (PACAF) and ROKAF. Conceptual studies were started in 1975/1976. Phase I (Hardened Tactical Air Control Center, baseline of automation of intelligence functions, and communications upgrade) became operational in 1981. Phases II and III have not been completed. ESD and RADC have provided some engineering support, while major parts of the program are run by the ROKAF. In some respects, an evolutionary approach is being followed; however, the CINC has had little voice in the requirements or development process, which probably has been a factor in some serious interoperability problems. The user has conducted and approved the testing. There is no test bed for the program.

Current hardware and software technology is being applied in the project.

Some Lessons Learned

Strong participation by the "real user" is needed in requirements and system development. Developer support at the operational site must be carefully planned and adequately provided.

12. EIFEL I AND EIFEL II

In general, the EIFEL systems provide automated assistance for tactical air control by creating schedules, retrieving and displaying data, preparing reports and messages, and storing and updating data bases. The forerunner system, EIFEL, was initiated by the German Air Force in the late 1960s and has been in use since 1974 at two ATOCs. DISTEL is a companion system to aid command and control of offensive air forces. The EIFEL/DISTEL system is now called EIFEL I, and the USAF is acquiring the DISTEL portion for use in ATOC Sembach. EIFEL II is planned to be an advanced system employing the latest in computer technology, data communications, and software.

Though USAF EIFEL I was originally intended to be an off-the-shelf replica of the German test bed system, it has changed extensively. Much of the software has required modification by the contractor or the German Air Force.

From the US viewpoint, EIFEL I is a turnkey buy from the German Government, while from the German viewpoint the system acquisition is evolutionary based upon an installed test bed system. After IOC, planned for 1982, the US system could well become an evolutionary base for an extended system. There has been extensive real user involvement in the program.

It appears that transition from EIFEL I to EIFEL II will be more revolutionary than evolutionary. More sophistication, hardware changes, and added functions will necessitate almost total rewrite of software.

Some Lessons Learned

The program illustrates that an effective operational system was fielded in about five years, mainly using off-the-shelf hardware and an operational test bed as a tool for evolution. On the negative side, planing for EIFEL II indicates the transition to EIFEL II may be difficult and costly, because the original architectural framework did not provide for the capabilities and technology now envisioned in EIFEL II.

13. NORAD CHEYENNE MOUNTAIN IMPROVEMENT PROGRAM - PROGRAM 427M

The purpose of the 427M Program was to provide missile warning, space object detection and tracking, space defense, and air defense. The 427M Program provided communications handling, space object tracking and cataloging, missile warning event processing, and generation of displays.

Program requirements were established and documented in 1968 by NORAD. These operational requirements were translated into technical requirements by the MITRE Corporation. The technical requirements were theoretically derived and pushed the state-of-the-art at the time. An added requirement, imposed by JCS, was to employ WWMCCS computers. Acquisition was traditional in that it was designed to be a standard procurement against a set of detailed, rigid specifications. The total system was to be tested and turned over "in total" for operations. The ultimate user was heavily involved from the start, and even developed major portions of the software. Software costs were more than 50% of total system costs.

Delays and overruns were experienced during the first half of the program. A major contributor to the difficulties was the mandate to use WWMCCS computers. According to the GAO, WWMCCS standardization cost at least $100M and degraded mission capabilities. As the program progressed, and the computer problems were overcome, numerous management and contractual changes were made to better align systems engineering and management responsibilities. The program did not meet planned dates, and the requirements baseline changed dramatically with time. As a result, the necessary management changes shifted the program acquisition strategy from a completely traditional approach to a more evolutionary approach.

In order to achieve an operational date, in the changing requirements environment, an "Essential Operational Equivalence" (EOC) was defined which meant that the new system could be cut over with no loss of operational function. The new system would provide upgraded availability and the capacity to evolve more easily than the old one. In 1979, EOC was demonstrated. At that time, the user recognized that Final Operational Capability (FOC) could never be achieved in the system due to the dynamic nature of the requirements.

Since the 1979 EOC, the evolution of the 427M system, and the entire Cheyenne Mountain Complex has continued. A continuing series of software program modifications, managed by the user, have upgraded communications, processing, and display capabilities. An Off-Site Test Facility has been added to allow this process to continue, independent of operations. This latter portion of the 427M Program could be said to be responsive to user needs, timely, and a good example of C^2 system evolution.

<u>Some Lessons Learned</u>

Increments should be kept small, relative to a "total" requirement. Requirements should be kept within the state-of-the-art in order to achieve low-risk, on schedule initial implementation.

Direction to standardize on a computer resource can be disastrous. A wide variety of commercial products with very different capabilities are available and should be matched to the requirement to the extent that logistics and other factors allow.

An off-line test facility is essential to sensible evolution of an on-line C^2 system.

Requirements for C^2 systems <u>will</u> change. It is unrealistic to try to predict the future with great accuracy.

APPENDIX F

ABBREVIATIONS AND TERMS

APPENDIX F - ABBREVIATIONS & TERMS

ADP	automated data processing
ADVIS	advisor
AFCEA	Armed Forces Communications and Electronics Association
AFLC	Air Force Logistics Command
AFSC	Air Force Systems Command
AFTEC	Air Force Test & Evaluation Center
Architecture	See p V-1
ARTY	Artillery
ASAS	All-Source Analysis System (Army)
ASD	Assistant Secretary of Defense
ASARC	Army Systems Acquisition Review Council
ATC	USAF Air Training Command
ATOC	Allied Tactical Operations Center (NATO)
BDE	Brigade
BETA	Battlefield Exploitation & Target Acquisition System
BN	Battalion
C^2	command and control (see p I-9)
C^3I	communications, command, control & intelligence
C^3I^2	C^3I + information
C^3S	C^3 Systems
C^4I	C^3I + Computers
CAFMS	Computer-Assisted Force Management System (AF)
CCITT	Consulting Committee International for Telephone & Telegraph
CDR	Critical Design Review
CECOM	Army Communications Electronics Command
CG	Guided Missile Cruiser
CHOP	Change in Operational Control
CHRM	Chairman
CINC	Commanders of Unified & Specified Commands
CINCLANT	Commander in Chief, Atlantic
CINCLANTFLT	Commander in Chief, Atlantic Fleet
CM	Configuration Management
CMC	Cheyenne Mountain Complex - 427M C^2 System (AF)
COBOL	A computer programming language (COmmon Business Oriented Language)
COMWESTLANT	Commander, Western Atlantic
CONUS	Continental United States
"CORE"	See p I-15
CV	Aircraft carrier
CW	Constant Watch (AF)

DA	Department of the Army
DARPA	Defense Advance Research Project Agency
DCA	Defense Communications Agency
DIV	Division
DoD Components	Office of the Secretary of Defense, Depts of the Army, Navy, Air Force, Unified & Specified Commands, Defense Agencies
DoDI	Department of Defense Instruction
DODIIS	DoD Intelligence Information System
DSARC	Defense Systems Acquisition Review Council
DSB	Defense Science Board
DT	Development Testing
DUSD/AM	Deputy Under Secretary of Defense/Acquisition Management
EA	evolutionary acquisition (see p iii, p I-15)
EAC	Echelon Above Corps
EIA	Electronic Industries Association
ENSCE	Enemy Situation Correlation Element (AF)
ESD	US Air Force Electronic Systems Division
EUROTOS	TOS/Europe (see p E-3)
FMFLANT	Fleet Marine Force, Atlantic
FOC	Final Operational Capability
FRIENSIT	Friendly Situation information
FSED	Full-Scale Engineering Development
GACT	Graphic Analysis Control Terminal
GAO	General Accounting Office
HOL	high-order language (computer)
HQ	Headquarters (Hq Dept of the Army, Office of the Chief of Naval Operations, Hq USAF, Hq USMC)
I/O	Input/Output
ICNI	Integrated Communications, Navigation & Identification
"ilities"	reliability, maintainability, availability, safety, survivability, producability, interoperability, supportability, transportibility, human factors, trainability, etc.
ILS	Integrated Logistics Support
independent testers	OTEA, OPTEVFOR, AFTEC
IOC	Initial Operational Capability
ISAs	Instruction Set Architectures
ISO	International Standards Organization
IT&E	independent test & evaluation
I&W	indications & warning
JCS	Joint Chiefs of Staff
JINTACCS	Joint Interoperability of Tactical C^2 Systems
JCS Pub	JCS Publication
JTFP	Joint Tactical Fusion Program
JTFPMO	Joint Tactical Fusion Program Management Office
JTIDS	Joint Tactical Information Distribution System

lead user	see p I-13
LR	Letter Requirement (Army)
LOA	Letter of Agreement
MAF	Marine Amphibious Force
MAW	Marine Amphibious Wing
MENS	Mission Element Need Statement
MIFASS	Marine Integrated Fire & Aerial Support System
MILDEPs	Military Departments
MIS	Management information system
NASA	National Aeronautics & Space Administration
NATO	North Atlantic Treaty Organization
NAVEUR	US Navy, Europe
NAVELEX	Naval Electronic Systems Command
NCA	National Command Authorities
NORAD	North American Air Defense Command
OASIS	Operational Application of Special Intelligence Systems
ODT&E	Office of the Director of Test & Evaluation
OJCS	Organization of the Joint Chiefs of Staff
OJT	on-the-job training
OMB	Office of Management & Budget
OPNAV	Office of the Chief of Naval Operations
OPTEVFOR	Navy Operational Test & Evaluation Force
OR	Operational Requirement (Navy)
OS	Outlaw Shark (Navy)
OSD	Office of the Secretary of Defense
OSI	open system interconnect reference model
OTEA	Army Operational Test & Evaluation Agency
OTH/T	Over-the-Horizon Targeting
OUSDR&E/C^3I	Office of the Under Secretary of Defense Research & Engineering/C^3I Office
P^3I	Preplanned Product Improvement (see p I-22)
PACAF	US Pacific Air Forces
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PJH	PLRS/JTIDS Hybrid (Army)
PLRS	Position Location Reporting System
PPBS	Planning, Programming & Budgeting System
Provider	See p I-13
RADC	Rome Air Development Center(USAF)
RDJTF	Rapid Deployment Joint Task Force
RFP	Request for Proposal
ROC	Required Operational Capability (Army)
ROKAF	Republic of Korea Air Force
RRDC	Rapid Requirements Development Capability (See p V-14)
RSI	NATO Rationalization, Standardization Interoperability

6.2/6.3A	DoD Exploratory Development/Non-System Advanced Development
SAC	USAF Strategic Air Command
SACLANT	Supreme Allied Commander, Atlantic
SAGE	Semi-Automatic Ground Environment
SDDM	Secretary of Defense Decision Memorandum
SDR	System Design Review
SDSF	System Design/Support Facility
SIGMA	Automated Maneuver Control System (Army)
SPO	System Program Office
SON	Statement of Need (USAF)
TAC	USAF Tactical Air Command
TACC AUTO	Tactical Air Control Center Automation (AF)
TACFIRE	Tactical Fire Direction System (Army)
TAFIG	Tactical Air Forces Interoperability Group
TCS	Tactical Computer System (Army)
TCT	Tactical Computer Terminal (Army)
TFCC	Tactical Flag Command Center (Navy)
T&E	test & evaluation
TOS	Tactical Operations System (Army)
TOS/OS	Tactical Operations System/Operable Segment (Army)
traditional approach	See p III-28
TRADOC	Army Training & Doctrine Command
transportable application software	computer programs that can be moved from one host computer to another without (or with minimum) change
TSQ-73	Automated Air Defense C² System (Army)
II MAF	Second Marine Amphibious Force
USA	US Army
USAF	US Air Force
USAFE	US Air Forces Europe
USAREUR	US Army Europe
USD(P)	Under Secretary of Defense (Policy)
USDR&E	Under Secretary of Defense Research & Engineering
USMC	US Marine Corps
USN	US Navy
user	See p I-13
user surrogate	See p I-13
VII Corps	US Army Seventh Corps, Germany
WWMCCS	World Wide Military C² System

APPENDIX G

THE ROLE OF EVALUATION
IN THE
C^2 SYSTEM ACQUISITION PROCESS

BY

DR. DAVID S. ALBERTS
MITRE CORPORATION
JULY 1982

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

THE ROLE OF EVALUATION IN THE C^2 SYSTEM ACQUISITION PROCESS

1. INTRODUCTION

Numerous studies of government systems acquisition have tended to support and document commonly held beliefs that major systems take far too long to go from concept to the field, frequently experience significant cost overruns, and are found by users to be less than fully satisfactory. In short, these systems often offer too little, too late, and cost too much.

Many study recommendations have concentrated almost solely upon streamlining the acquisition process to get systems to the field sooner despite the fact that in the final analysis, the value of a system is, with few exceptions, more a function of the length of its useful life rather than the time taken from conception to implementation. To make matters even worse, the evaluation process is often a casualty of these streamlining efforts, despite the fact that evaluation plays an important part in helping to ensure a useful life for a system.

Thus, the net effect of an improperly conceived "streamlining" effort may be to get a system into the field sooner but accelerate its obsolescence. While a properly structured and executed evaluation process is no guarantee of a long, useful life for a system, the absence of one almost ensures an undesirable result.

A properly conceived and conducted evaluation function includes far more than post-deployment determination of performance, or even operational utility. It is a continuous process through the life cycle of the system. Its objective is to ensure that the system is conceived, designed, developed, and operated in a manner consistent with the mission(s) it supports and the environment in which it operates (or can be expected to operate).

The role which evaluation plays varies according to the phase of a system's life cycle in question. A well designed evaluation process takes this changing role into consideration, along with explicit considerations of "how much" evaluation is appropriate for the situation at hand.

The remainder of this paper will be devoted to discussing a concept for mission-oriented evaluation of C^2 systems, including the nature of the criteria which should be used, and how evaluation should be applied in the various phases of system acquisition.

Since the evaluation process is an integral part of system acquisition, the next section briefly discusses the "traditional" approach to acquisition and contrasts it with more recent approaches to C^2 system acquisition. The nature of the evaluation process needed to support such acquisitions is also discussed.

2. VIEWS OF C^2 SYSTEM ACQUISITION

The traditional view of the acquisition process is that this process is a well-mannered sequence of tasks, including a "test and evaluation phase" which progresses from concept development to design, from design to prototype development, and from prototype to production with go/no-go decisions and competition at key points in the process.

This well-behaved approach to design, development and production, which has served as the only role model for system acquisition until quite recently, rests upon a careful and detailed specification of system "requirements." The better the statement of requirements, the faster the development; the cheaper the price, the better the system; at least so went the acquisition folklore. "Freeze the requirements" has been the hue and cry of the system engineer. Change is anathema to the system developer, because change is perceived to lead to uncontrolled costs.

The freezing of C^2 system requirements and the use of performance measures related to these "requirements" as criteria for evaluation are actually antithetical to the interests of the user for whom these systems are intended to support. Change is not, as system developers often implied, a result of some mental laziness or lack of vision on the part of the user, but an unavoidable fact of life.

The "traditional" acquisition approach appears to be fatally flawed for systems for which change is inevitable. Since change is such a fundamental aspect of C^2 systems, these systems will require new design, acquisition and evaluation concepts.

Evolutionary acquisition, that is, an approach to the design and development of a system which ensures that the system can easily accommodate change, has been recommended for C^2 systems. While the objectives of the evaluation effort remain the same for this "new" type of acquisition approach, the mission-related evaluation measures employed must be augmented by a set of measures which specifically deal with the system's ability to accommodate change. Proper evaluation is even more crucial for system acquisition using an evolutionary approach than for those which employ a more traditional approach. This is because the future of the system will depend upon the results of a continuing evaluation effort.

Evolutionary acquisition, to reach its full potential, clearly must begin "at the beginning." However, many systems, particularly C^2 systems, are not replaced <u>in toto</u>, but are "augmented," "modernized," "enhanced" or "improved." Regarding the evaluation process, the assessment of these enhancements should be conducted as though the entire system is being acquired. To do any less is to guarantee that the evaluation effort will be too narrowly focused and that the resulting "new" system will continue to exhibit the problems of those acquired in the traditional manner. This does not mean that, given the constraints imposed by existing systems architectures and implementations, one can achieve the same end result as the one which could be achieved by a "new" start, but that given the constraints, the system will live out its days as gracefully and effectively as possible.

3. C^2 SYSTEM EVALUATION

The objective of the evaluation process is to insure that the system will be of value. Since system value is derived from its contribution to the success of some mission or function, C^2 systems have no intrinsic value. How well it does its "job" is, therefore, only of potential value. The degree to which this potential is reached depends upon many circumstances which are not within the control of system designers, developers or operators and, to a large extent, not even within the control of the commanders or decision makers (users). Therefore, system evaluation should encompass much more than how well a system "performs"; it should reach beyond the internal operations of the system to the contribution that the system makes to the operational tasks, functions and/or missions it was designed to support. This type of evaluation clearly cannot be done at arms length from the operational user. The user must be an integral part of this process. Evaluation which focuses upon a C^2 system's contribution to a specific mission has become known as mission-oriented evaluation.

The process of evaluation begins with the assessment of a system concept and is continuously applied to insure that the "value" of this concept is maintained as the system becomes further defined and specified. Evaluation is a formidable enough task if a system's operational context is stable, but given the changing nature of the threat and the users, the value of particular C^2 system characteristics and attributes will change over time and must, therefore, be factored into the evaluation (as well as the design and development) process. In this regard the evaluation process is very similar to a control process.

In order to accomplish this "control" objective, a suitable evaluation methodology is required. Such a methodology is discussed briefly below and in more depth in the attachment to this paper (p G-13) for those readers who are interested in exploring this subject further.

To be suitable, an evaluation methodology must be capable of relating the technical attributes of C^2 system components (or subsystems) to mission outcomes. The methodology discussed in the attachment "decomposes" this problem by formulating a set of measurement levels and linkage models which provide a means of inter-relating these levels.

The methodology identifies the following six levels of variables or measures:

- C^2 System Attributes
- C^2 Component System Technical Performance
- Information Quality
- C^2 Functional or Task Performance
- Decision Indicants
- Mission Outcome

A <u>C^2 System Attribute</u> may be descriptive of a system concept (e.g., distributed data bases) or a technical approach (e.g., frequency hopping). <u>C^2 Component System Technical Performance</u> measures relate directly to their capacity (memory size, band width); speed (band rate, response time, revisit time); coverage (range, spectrum); reliability or survivability (bit error rate, mean time to failure). The <u>first</u> of the "linkage" models called for by the methodology is designed to relate the impact that different technical attributes have on performance (e.g., impact of memory size of response time, impact of band width on error rate) given certain operating conditions or threat.

Since C^2 systems are primarily designed to collect, analyze, interpret and communicate information or instructions, the methodology calls for a measurement level which focuses upon the quality of this information. These measures include the currency of the information, its precision, correctness, completeness and degree of unambiguity (information content) as well as its ease of use. The second linkage model called for in the methodology is to relate C^2 Component System (Technical) performance to these measures of information quality, (e.g., impact bandwidth and response time on information currency).

The value of information quality is contextual; that is, it depends upon what it contributes to the performance of C^2 functions or tasks (e.g., detection, identification, classification), given certain conditions. The methodology calls for the development of a set of measures which reflect the degree to which these functions are accomplished. For example, the probability of detection as a function of time given the nature of the threat and a scenario could be used to measure this aspect of C^2 system function performance. Here, too, a linkage model is required to relate changes in information quality to these measures of functional performance (e.g., impact of information accuracy or completeness on probability of correct identification).

The successful and timely accomplishment of specific C^2 system functions (like detection) are necessary but not sufficient conditions for the success of the military missions which they support. To determine their value or utility, they must be looked at along with weapons, manpower and logistic systems. The evaluation methodology discussed in the attachment recognizes and tries to deal with this reality. Again, a linkage model is required, this one capable of relating C^2 functional performance to mission accomplishment as measured by variables appropriate to the mission. When dealing with the determination of the mission measures, it is important to scope the problem sensibly to prevent sub-optimization. For instance, in the case of air defense, a measure of enemy killed or even enemy/friend air casualty ratios is too narrow to be very useful. Damage to friendly ground forces defended by our air defense system and the ability to use the air space for friendly assets must also be included in a formulation of the objective function. Also, there must be a set of constraints which reflect the role of a particular mission as it relates to the overall military situation (e.g., resource limitations).

Given the formulation of a set of measures at each level and linkage models to relate one level to another, a baseline measure of mission accomplishment could be derived (from the technical attributes and performance of current system) and used to provide a basis with which to evaluate proposed system improvements.

The methodology also proposes the use of decision indicants as a way of representing critical C^2 system functions for systems which are designed to provide support decision makers. Measures are proposed which deal with the important aspects of decision making, particularly option generation and assessment, and the determination of decision criteria.

While the focus of C^2 system evaluation is primarily fixed upon the mission-related value of a C^2 system or component, the existing DoD acquisition process seems to have been designed primarily to monitor costs and schedules. These three measures [value (derived from performance), cost and schedule] are interdependent and when a program gets into trouble on one, the "fix" usually involves sacrificing program objectives of one or both of the other two. The role of evaluation in the acquisition process should be designed to insure that: (a) reductions in system performance or capability objectives which "need" to be made to meet budget and schedule targets (or diminish overruns and delays) do not result in a disproportional reduction in the value (mission contribution) of the C^2 system, and (b) changes in desired system capability which are due to a changed threat or environment are promptly identified.

These "continuous" trade-off analyses can only be accomplished if the evaluation process is continuous and well integrated into the acquisition process and its decisions. Without the capability to judge the impact of fielding a C^2 system which does not meet its original design specifications or where its original specification is no longer valid, there is little to insure that the system which is expected to be delivered will still be cost effective or even necessary.

4. THE CHANGING ROLE OF EVALUATION IN THE ACQUISITION PROCESS

As the system proceeds through the various phases of the life cycle, both the nature of the evaluation activities and the organization conducting the evaluation change. The focus of evaluation at a given point in the life of a system is a function of the decision(s) which need to be made and the data which can be obtained. At different points in time, the evaluation effort will involve addressing one or more of the following questions:

1. If the C^2 system achieves its stated objectives, is it worth the cost?
2. How important is it that each of the capabilities is achieved? To what degree?
3. Is the architecture being proposed the best (a good) framework for the system?
4. Is the system design consistent with the capabilities being sought?
5. Has the design been properly specified?
6. Does the system as built meet the specifications (design, performance, etc.)?
7. Have the contractors fulfilled their obligations?
8. Are the system concept and capabilities still relevant?
9. What changes are desirable? At what cost?

For the purpose of this discussion, let us break down a system's life into the following four major phases: (a) mission analysis; (b) architectural analysis and system design; (c) development and testing; and (d) change control. The remainder of this section will be devoted to discussion of what questions are addressed, who should address them, and how the measurement continuum and linkage models called for in the evaluation methodology relate to these major system phases.

a. Mission Analysis

Mission analysis should not merely provide a broad based justification, but should be able to: (a) determine an upper boundary on the value of a C^2 system which supports it; and (b) identify and bound the impacts of both anticipated and unanticipated change with which the system must cope. For all practical purposes, a baseline of sorts always exists, and the maximum value of a system represents the value-added a new or enhanced system brings to the mission. The estimation, even roughly, of an upper boundary, and the expected impacts of change are very important since they can profoundly influence the selection of a system concept, and help determine the nature of the resources which should be devoted to flexibility. In terms of the measurement continuum presented earlier (six levels--p G-5), outputs from a mission analysis are required to:

1. establish the relationship between the potential value of a system and its expected value

2. contribute to a determination of the nature and extent of the operational use of the system (these operational measures serve to modify the theoretical or maximum potential value of the system)

3. provide inputs to developing operational definitions of the decision indicants (i.e., what constitutes a "complete" set of options)

4. identify the "key" decisions upon which the evaluation should be based.

b. <u>Architectural Analysis and System Design</u>

Architectural analysis and system design involve the development of a system concept and the basic structure which will guide its development. Outputs from the mission analysis stage serve to provide an upper bound on costs, as well as determining reasonable ranges for system characteristics, such as survivability, connectivity, functionality and flexibility. Given these rough parameters and a knowledge of existing and emerging technology, a design concept along with a range of implementation options and corresponding performance and cost estimates are developed.

These performance estimates, in conjunction with the outputs from the mission analysis, establish for the first time at least a rough quantitative link from system performance to the nature of the information which could be provided and the values of the decision indicants (for key decisions). Previously, estimates linking the decision indicants to potential and thus expected value were accomplished in the mission analysis phase. Based upon an examination of these links, a particular architectural concept could be selected for implementation, or to guide the development of a facility which can be used to "mock-up" or simulate for the user alternative system concepts, capabilities and features. Experience has shown that without a tangible vehicle to use, users have great difficulty envisioning how a proposed C^2 system would work, or impact on their Operational Procedures. This concept of a Rapid Requirements Definition Capability (RRDC) is discussed in Chapter V of this report.

c. <u>Development and Test</u>

The development and test phases of a system's life cycle are reasonably well understood compared to what has been envisioned for the mission and architectural analyses and the RRDC. However, the notion that requirements are changing/evolving rapidly compared to the design/development time frames requires that the system's ability to change must not only have been determined in the mission analysis phase and incorporated in the architectural and design analysis, but must be carried through into the system development and test phases.

There are two basic aspects of change--ease and time. Ease is an embodiment of cost, disruption to the system, error effects and the like. Time is relative to the dynamics of the environment. Self-correcting/adjusting system features, such as dynamic alternate path routing, as well as more traditional enhancement or replacement techniques, will all play a part in providing the needed flexibility (or adaptiveness).

Another point worth mentioning is that, when closely correlated with an internally consistent measure continuum like the one presented here, the design verification and system testing procedures are employing measures which are traceable back to the original mission-oriented measures of value. This means that if for any reason (technology breakthrough, mission alterations, budget reduction or increase) a significant change in design or implementation is dictated or considered, the impact which it will have on the "value" can be traced and provide an input to the decision making process.

d. System Evolution and Change Control

Change control normally is incorporated in the activities associated with maintenance or operations. Evolutionary systems, which are the wave of the future, will not be "maintained" in the same sense that existing systems are. In a sense, they will be managed either explicitly by micro-mission analyses, etc., or implicitly by their use as fully adaptive and learning systems.

The development, then, of a mission-oriented set of measures (embodied in an RRDC, as augmented by a System Design/Support Facility (see Chapter V)) which can be employed through the life cycle of a system (not merely during development and test) is of paramount importance. Of equal importance is the notion of designing for change and the incorporation of this concept into the evaluation and selection process that leads to a C^2 system's specifications. If progress can be made in introducing these concepts into widespread practice, C^2 systems will be far more responsive to users and, hence, better received and used.

5. SUMMARY AND CONCLUSIONS

Three basic concepts which have been presented in this appendix are central to a coordinated evaluation/acquisition process. The first is the notion that for many systems, rapidly changing user requirements are the norm. The second is the notion that there is a set of evaluation measures (which contain at least three types of measures--those related to mission performance, those related to C^2 functional performance, and those related to C^2 component performance) which can be linked together by testable hypothesized relationships. The third is the notion that the evaluation process is continuous, particularly for evolutionary systems.

Conclusion 1:

A mission-oriented evaluation process must be the driving force in both the development of an initial set of system requirements and in subsequent analyses of changes in anticipated or designed capabilities.

Conclusion 2:

To facilitate the necessary interaction between user/developer in the requirements/design process, a Rapid Requirements Definition Capability, later augmented by a System Design/Support Facility, is required. This facility should be capable of providing the link between component system characteristics, functional performance, and contribution to mission.

Conclusion 3:

Evaluation of the contractor should be separated from evaluation of the mission-related value of the system.

ATTACHMENT
MISSION-ORIENTED EVALUATION METHODOLOGY

The evaluation methodology described in this attachment has become known as mission-oriented evaluation since an explicit attempt is made to relate "traditional" measures of system performance to mission-related measures of value. The methodology is based on a multi-level measurement structure, with each layer "related" to the next succeeding layer by a set of linkage "models" or hypotheses.

Figure 1 depicts the three types of linkages involved in going from a measure of system performance to a measure of the value of the system as expressed in terms of its impact on mission-related outcome measures. System performance measures, (e.g., response time) require substantial additional analysis before the extent to which their potential value is achieved can be determined. For example, three analytical steps are required to determine the impact of improved response time on mission outcome: First, response time must be related to the currency of the information reaching the commander, (that is, how long ago was the enemy spotted in this particular location?). Second, information currency, a measure of "information quality" must, in turn, be related to better decisions; that is, in this case, the determination of a target's priority and the assignment of weapons to the target. Information accuracy--in this case the location and description of the target--is clearly important in terms of getting munitions to the right place quickly. Third, the contribution to a military mission, say Air Defense, that the ability to destroy a target or group of targets (or conversely, the failure to destroy), must be ascertained.

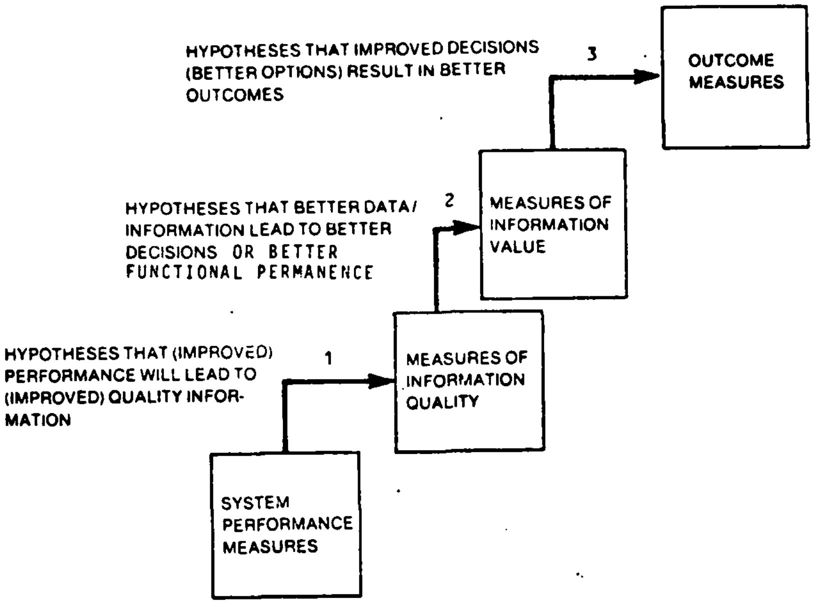


Figure 1. Linkage Hypotheses

These three steps in general can be posed (as in Figure 1) as linkage hypotheses or formulated as models. Information collected in the evaluation process, therefore, needs to contribute not only to determining the level of system performance, information quality and value in particular instances, but must also contribute to the development of a mission-oriented model which can link adjacent levels to one another.

Decision Indicants

In the case of determining a system's contribution to a reasonably straight-forward mission-related measure, like the single-shot probability of kill, one can make a direct evaluative argument using (specifying) the linkage hypotheses (1, 2, and 3) depicted in Figure 1. In the case of a system's contribution to strategic planning or resource management, and in determining a system's overall contribution to a specific mission (over a wide-ranging set of scenarios and circumstances), a different, more tractable approach is needed. This is because the approach outlined above would require "individual" analysis for most likely scenarios and conditions. Therefore, a way is needed to "abstract" the underlying qualities such a system should have to be of value, rather than "sum" and weight analysis results for many specific scenarios and conditions.

The use of decision or functional indicants (see Figure 2) which: (a) can reasonably be related to a system's value or utility, and (b) can be directly related to parameters of system performance or capability, is such an approach.

A decision indicant is an attempt to measure the contribution of a system to the decision making process.

In a command and control system, the distinction between tasks and decisions blurs. For example, target assignment, detection, and identification could be called tasks, functions or decisions. By setting up a methodology which includes the use of indicants, an attempt is being made to minimize the use of linkage models of types 1, 2, and 3, which, while often being conceptually straightforward, are difficult and costly to establish empirically for a large set of tasks/missions. Instead, it is proposed to deal with linkage models that, rather than dealing with events

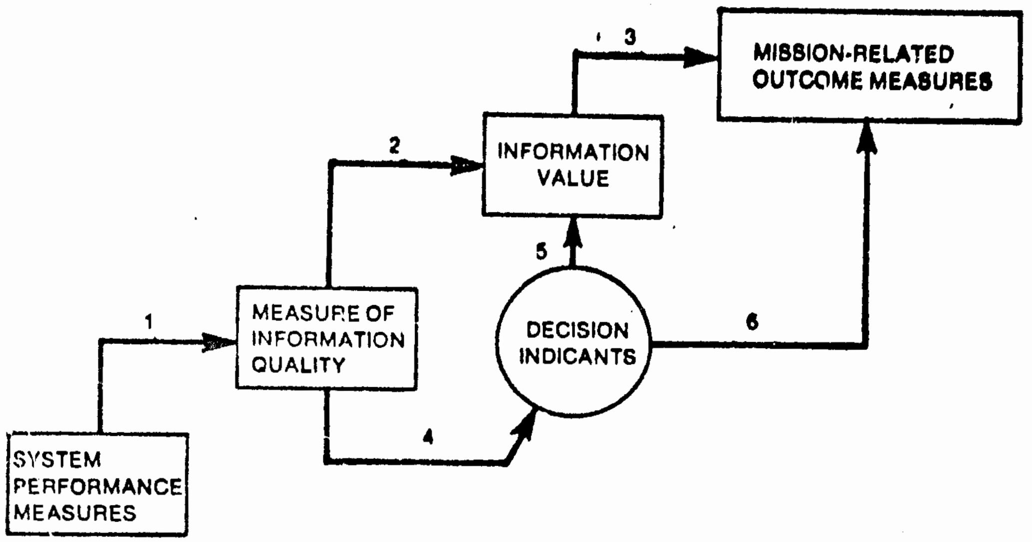


Figure 2. Decision Indicants

or specific decisions, attempt to reflect the key underlying characteristics of the decision-making process--linkage types 4, 5, and 6. These linkage hypotheses will be discussed after candidate decision indicants have been developed. The basis for the development of such indicants lies in the research work that has been done on the decision process.

The nature of the decision making process has been studied by students of various disciplines. A considerable amount of theoretical work and informed conjecture has been focused upon how individuals/groups for cognitive models of the "problems" they face; the nature of these models; and the way in which decision makers formulate, assess and select among options. The influence of time and data constraints and the criticality of the decision have also been favored subjects of inquiry.

Whether a decision maker approaches the problem by "satisficing," that is, by selecting the first feasible option (one which meets some minimum standards) or takes a "maximin" approach which involves selecting the option which is felt to have the smallest downside risk or uses an expected value calculation, which involves weighing various potential outcomes by the probability of occurrence, is a matter of individual style and the extingencies of the situation.

An information or command and control system should be able to provide support to decision makers who employ various approaches. Accordingly, an evaluation methodology should be capable of assessing the potential of a system to provide this support.

Although each group has brought its own perspective, experience and issues to their investigations, certain aspects, elements or parameters of the problem seem to be universal. In their own jargon, these disciplines describe the decision making process as being comprised of the following essential steps, although attaching differing degrees of importance/emphasis upon them.

Step 1. Determination of Goals, Objectives, Desired Outcome(s), Criteria and/or Value Metric.

Step 2. Development/Identification of Feasible Alternative Actions/Options or Controllable Variables.

Step 3. Identification of Environmental Factors, Scenarios or Uncontrollable Variables.

Step 4. Determination of Value or Utility of Alternatives Conditional upon Scenarios and Environmental Factors.

Step 5. Selection of Alternatives based upon Likelihood or Risk Assessment.

The determination of goals and associated decision criteria and their respective weights is an exceedingly challenging problem for decision makers and gets more difficult as one moves up in the organization. In the case of a tactical commander, the criteria encompasses not only variables directly associated with target damage, but also include such things as different as deterrence, enemy perceptions and resource utilization.

The identification of a "good" set of options from which to choose can easily be the key to satisfactory performance. A fire control officer, who knows the forces at hand in his domain, can "formulate" a set of options which, in essence, pre-screen those which are not feasible. Given the location of his resources, he further can quickly screen out those which are less desirable. Given an ability to communicate with, and/or knowledge about other functional areas and their status and intentions, the options set available may increase, or the choice may be altered.

The ability to understand the situation faced quickly and accurately is usually the first priority a manager or commander has. For example, a system must be able to help a commander distinguish among duplicate reports, and a mounting enemy thrust, among a false alarm, an ambush and a real call for help; between a high priority call and a routine (delayed or off-line) assignment.

In selecting from the options available, a decision maker must be made aware of how the situation may impact the effectiveness of the alternatives. For a commander, a knowledge of the specific individual officers and their characteristics, such as leadership skills, ability to deal with stress, and experience are often critical.

In other words, a decision maker basically develops, refines and chooses from a set of alternatives. In order to select the "best" options, he must assess the likely result of applying each option under different potential environments and determine the value or utility of the possible outcomes.

Hence, an information or C^2 system can be of value to a decision maker in one of four ways:

- First, it may contribute to the knowledge of, and understanding of, the options.
- Second, it may contribute to the ability to assess the situation or environment.
- Third, it may help to project the outcome which would result from a given option/situation pairing.
- Fourth, it may help evaluate potential outcomes in terms of a variety of criteria.

Linkage Hypothesis Related to Decision Indicants

Each of the four types of decision indicants developed above (option, situation, outcome, value) have some face validity, that is, they generate a degree of confidence that if improvements in them can be achieved, results of value will follow. In Figure 3, the decision indicants are linked in four ways to measures of information quality (attributes) and to measures of value. Each of these linkages is illustrated below.

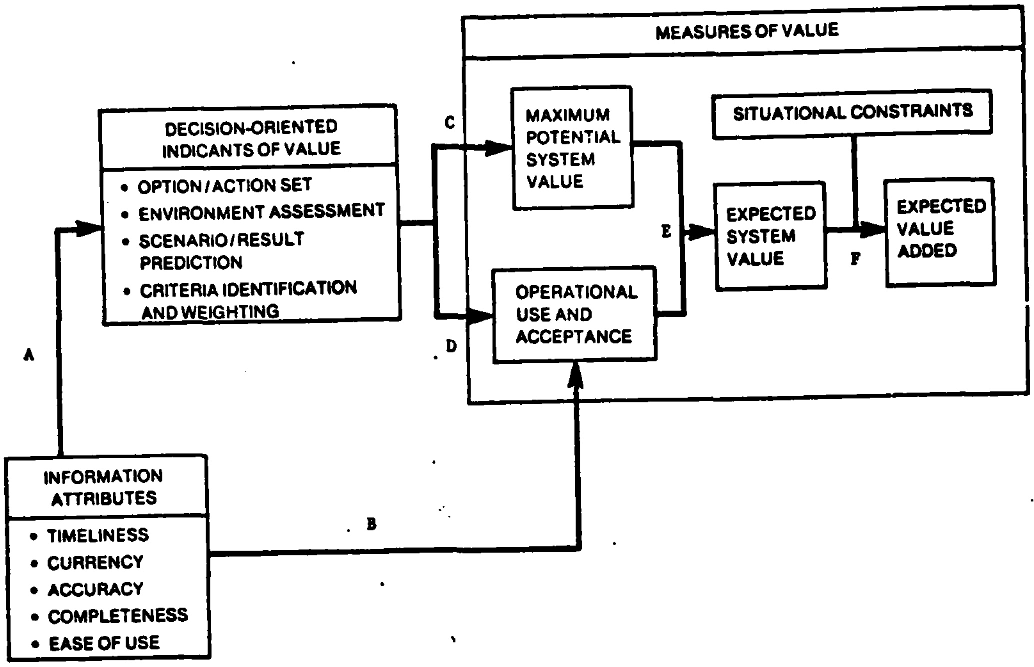


Figure 3. Decision Indicants and *Linkage Hypotheses*

Linkage type A involves the conjecture/assumption that an improvement in information quality, say currency of information, will be related to a more complete or better formulated option set. If, for example, mobile enemy and friendly locations can be digitally updated on a 15-minute basis, this situation could be compared to the commander having only the positions from the beginning of the day. Whether or not, and how the commander incorporates these in the decision making process is the focus of linkage hypotheses of types B and D.

Linkage C would, in this case, be the connection between the addition of new options, which might incorporate resources whose existence may not have been known previously, and the likelihood (frequency) of the new option being the "best" in a situation. If it appears that options which are "added" are always inferior, then their incorporation into the decision process has no (or even a negative) value, that is, the "maximum potential system value" is zero. On the other hand, if their selection would be appropriate in a fair number of cases, this would enable the commander to do a better job and the maximum potential value would be some positive number. The difference between maximum potential value and expected value (E) incorporates the realities of an operational situation with real decision makers. Decision makers often do not make "correct" decisions because of error and lack of confidence in the system, and this impact must be considered.

Finally, the linkage between expected system value and expected value-added (F) reflects the real impact of the decision-maker doing a better job. For example, if a "new" option is "better" because it gets a weapon fired ten seconds faster than the best previous option but the target has already moved away, the value of better decision making is nil. Conversely, if the ten seconds does, with some probability, increase the chance of interdiction and/or kill, a link to mission related value has been established.

The breakdown of the Value Measures in Figure 3 serves to illustrate two major aspects of the C^2 system evaluation problem. The first is the contextual orientation of a measure of utility; the second, the contribution of the individual decision maker to the value structure.

The contextual aspects of the problem involve the transition from the expected <u>potential</u> value of the system to this expectation (Linkage F). This translation of a potential into a reality is dependent wholly upon factors <u>outside</u> the system itself. In Figure 3, this is depicted simply by the addition of a box representing situational constraints or environment factors. The system supporting the commander can be perfect in every way (e.g., having no delays and complete and accurate information), but if weather is bad, or equipment breaks down, or enemy forces are overwhelming, the realized value with the system falls substantially below its potential. This principle has been captured in the proverb about the weakest link in a chain.

The characteristics of a decision maker and the impact of how, or even when he uses the system is represented by the depiction of the expected potential being comprised of two parts--a maximum theoretical value and a modifier consisting of operational use variables (Linkage E). If the decision maker uses the system (and the information provided) perfectly, the maximum value is what is achieved. However, in reality the human condition is less than perfect and how the system works with people must be factored in explicitly.

APPENDIX H

INFORMATION SYSTEMS ARCHITECTURE
IN
COMMAND AND CONTROL

BY

HAROLD M. WISHNOW
PLANNING RESEARCH CORPORATION

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

INFORMATION SYSTEMS ARCHITECTURE IN C^2

A. INTRODUCTION

What is Information Systems Architecture? What are the perspectives developed by information systems architects, and what are the implications of this thinking for the command and control community? What specific initiatives should the command and control community undertake at this time to improve productivity of future system developments and to position itself to better exploit emerging technology? What should the command and control community do now?

These are basic, down-to-earth questions. They are difficult, so it is not surprising that they evoke many different answers, reflecting widely different opinions and viewpoints. This paper is intended to provide an initial baseline for discussion of the role that Information Systems Architecture plays in the acquisition of command and control (C^2) systems. To develop and field C^2 systems that can survive the test of time requires an architectural framework that will encourage technological advances while introducing functional components in a timely manner.

To provide a basis for departure, this paper describes the growth of Information Systems Architecture concepts and develops a current definition of the terminology; examines the C^2 requirement for Information Systems Architecture; looks at technology trends that can be anticipated in the near future; considers the problem of architecture's role in evolutionary acquisition; and, finally proposes some steps toward introduction of Information System Architecture concepts into C^2 system development.

Many thanks go to Mr. Hans I. Johannson for his extensive contribution to this paper.

B. INFORMATION SYSTEMS ARCHITECTURE

1. Historical Perspective

Since the concepts of "computer architecture," "computer systems architecture," and "information systems architecture" are relatively new, these terms are often used in the current literature with very different meanings. The terms "architect" and "architecture" are borrowed from construction, where the terminology is well understood to mean the science, art and profession of planning, designing and creating buildings, dams and similar civil structures. The work performed by construction architects and the products they produce are well understood because the profession has matured over a long period of time. Similarly, the work performed by and responsibilities of civil engineers and by contractors in building civil structures is well established by common practice. The word "architecture" is also used with a more philosophic meaning to connote the design and drawing up plans for any systematic structure or framework. For example, in this latter sense, we speak of "architects" of the Constitution.

Civil engineering architects make extensive use of standards in planning and building different structures. Steel I-beams can be specified by selecting from among different sizes in a catalog; a wide variety of wooden beams are available in different shapes and sizes, such as the most basic building material, the 2" x 4" wall stud. But in spite of this standardization, practically no two major buildings (bridges, dams) are the same. Of course, there are lots of "row" houses, but a stroll down any principal street of any city will not turn up many identical buildings.

This implies that the practice of construction architecture has discovered places where standardization pays good dividends and places where it does not. Through thousands of years of practice and many different stages and styles--Egyptian, Greek, Roman, Gothic, Baroque, Eclectic, and Modern--construction architecture has discovered certain principles, certain areas where standardization has higher productivity payoff, and other areas where customized approaches are more suitable. The "principles" discovered by civil engineering architects, and embodied in strength of materials and allied disciplines, are few and basic, in contrast to the many passing styles and design trade-offs.

Use of the term "architecture" in connection with computer-based information systems was popularized in the early 1960's by a group of computer designers, headed by Amdahl, at IBM. These designers used the term to describe the common attributes of the then-new IBM 360 series family of computers. The common attributes of this family were expressed in terms of the assembly-level instruction set (and some I/O connection conventions). The machines in the IBM 360 family were designed by different engineers, with different speed and cost constraints. However, these hardware details were deliberately made invisible to the using programmer, who followed the overall blueprint or block diagrams of the system as a whole, and built programs executable by any one of the class of machines.

Various authors' definitions of computer architecture have been postulated and compiled by S.S. Reddi and E.A. Feustel.[1/] The passages below are representative.

- According to Brooks, "The computer architect designs the external specifications, gross data flow and gross sequencing of a system. He is, like the building architect, the user's advocate. He must balance the conflicting demands of engineer (cost, speed), programmer (function, ease of use) and marketing (function, speed, cost) to yield the machine of greatest true value to the user..."

- Foster, in Computer Architecture, introduces the architect as follows: "The computer architect...is unconcerned with the insides of an adder or a shift register. His job is to assemble the units turned out by the logical designer into a useful, flexible tool that is called a computer." "The field of computer

[1/] S. S. Reddi and E. A. Fuestel. "A Conceptual Framework for Computer Architecture," ACM Computing Surveys, Vol. 8, No. 2 (June, 1976), 278-279.

architecture, or 'the art of designing a machine that will be a pleasure to work with' is only gradually receiving the recognition it deserves. This art (one cannot call it a science), is one step more abstract than that of a logical designer, which in turn is abstracted from the study of electronic circuits." Foster also suggests: "Computer Architecture is the profession of adopting present-day technology to the solution of current computing problems and of dreaming about the future of the field in such a way as to influence it for the better."

- Beizer describes the architect's job as"...the design of a hardware/software complex, subject to realistic technical, economic, operational and social constraints such that it: 1) works, 2) is optimum and 3) survives." He summarizes the architect's role by stating that "it is synthetical, catalytic and translative. His design is a synthesis of the substance of subordinate disciplines."

- Abrams and Stein discuss the architect's duties: "The job of the computer system architect is to develop an overall concept of a machine--what it can do and how that solves the problem for which the machine is intended. Just as an architect who designs houses must consider utility, appearance, and compatibility with the neighborhood, so must the computer designer balance requirements, user interface, and costs to make a viable design."

- Finally, the term "architecture" is used by Amdahl et al, in their description previously discussed "to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design, and the physical implementation."

Note the implicit assumptions in these early definitions of computer systems architecture. What are the essential tasks of the software engineer and the hardware engineer? The essential job of the software engineer is to build algorithms that solve some problem or set of problems. Implicitly, a software engineer was not to be concerned with throughput performance. The throughput performance of a computing system was a function of such factors as speed of instruction set and size of memory. These factors, in turn, were a function of overall performance of a machine in a family of compatible machines, which in turn was a function of price. All of these latter considerations were the province of the hardware engineer. The legacy of this early thinking is still with us, promulgated throughout DoD by existing policy directives.

Over the last few years, the Military Computer Family Industry Advisory Committee, recently renamed the Computer Architecture Study Committee, has assisted the Government in establishing a "standard" set of computer architectures in DoD instruction.[2] It defines computer architecture as a:

> "precise description of computer attributes as seen by the assembly language programmer. Architecture is, therefore, the conceptual structure and functional behavior of the computer, as distinct from the organization of data flow and controls, logic design, and/or physical implementation. Defining attributes of a computer architecture include instruction set, number and type of registers, input/output protocols; it does not include or infer a stated level of component selection, internal computer partitioning, manufacturing technology, or any other vendor specific parameters."

Note the similarities between this definition and that provided by Amdahl in the 1960s. Based on this definition, some computer architecture families have been approved for DoD. These are:

Family	Controlled within DoD by
AN/UYK-7, AN/UYK-43	Navy
AN/UYK-19	Army
AN/UYK-20, AN/UYK-44	Navy
AN/GYK-12	Army
AN/GYQ-21	Army

One might wish that information systems architectures were as highly developed as their construction relative. However, the reality is that Information Systems Architecture continually finds itself on the horns of a "standardization" dilemma. On the one hand, it can be said that establishment and promulgation of standards is a measure of the maturity of a profession and a society. On the other hand, premature establishment of standards, or selection of arbitrary and insensitive standards, or standardization of the wrong thing, only acts to stifle the initiative and

[2] Department of Defense Instruction. Draft List of DoD Approved Computer Architectures, Instruction Number 5000.XXA, April 19, 1979.

innovation that are the vital forces that fuel productivity improvement in today's rapidly changing technology.

Developing a command and control system can be likened to the building of a house. While many of the subsystems may be viewed as building blocks and may in the future be standardized, much as plumbing has been in the building trades, the arrangement of these subsystems into systems must allow flexibility for the user. The people who inhabit houses are different, and a particular house reflects the flexibility and style of the residents. C^2 systems must be designed in the same manner: flexible enough to accommodate the command style of the user but based on accepted and established system building techniques.

A brief review of some major accomplishments of the computer-communications industries is useful. Figure 1 below lists some of the major evolutionary developments in communications, computer, and data

	1920's	1930's	1940's	1950's	1960's	1970's	1980's
Communications Technology	Inductive Loading Passive Lumped Electrical Networks Filters	Network Response to Time-Varying Signals Negative Feedback	System Stability Active Network Design Frequency Division Multiplexing	Information Science Channel Speed, Noise and Power Interrelated	Circuit Switching Electronic Switching, Electronic Translation	Packet Switching Layered Communication Protocols for Open Systems Architecture Time Division Multiplexing	Broad Band Satellite Communication Utilities Demand Assignment Multiple Access Millimetric Frequencies
Computing Technology				Computer Engineering/ Manufacturing Feasible	Operating Systems, Programming Languages	Computer Communications	Applicative State Transition Systems
Data Base Technology				Report Generators	Inverted Files	Hierarchical and Networked Systems	Data Base Speed, Noise and Power Interrelated

Figure 1. Communications, Computer and Data Base Technology

base technology over the years. A glance at this figure reveals that communications technology is much older than computing and data base technologies. The chart suggests that the current practice of communications engineering can be traced back to developments in the 1920s, and communications principles can be traced back even further, to Maxwell's discovery of the relationships in electromagnetic fields, or Alexander Graham Bell's invention of the telephone, for example. In any case, communications technology is much more mature than either computing technology or data base technology. In fact, as is shown in the figure, communications technology is sufficiently mature so that it gave rise to the fundamental notions of system stability with negative feedback in the 1940s, and the basic ideas of information science as a measure of transmission efficiency in the 1950s, that are the elements of the new, emerging profession of Information Systems Architecture in the 1980s.

The transfer of a well- developed body of difficult mathematics for analysis of dynamic systems (and incorporating synthetic design techniques) from one technology area to another apparently-unrelated technology area, was a revolutionary development. It was important because the body of mathematical development transferred was sophisticated, being shaped by many scientists and engineers over a long period of time, and represented a significant intellectual investment. Today, we often honor this signal accomplishment, calling attention to the benefits of technology transfer (there is even a professional engineering society by this name) and seek to emulate the achievement whenever possible.

Communications technology is mature enough to boast achievements of the first rank, as described above. Therefore, it seems sensible to look toward this technology area for guidance in development of crucially important standards. The importance of a solid foundation for Information Systems Architecture cannot be overstated. What better place to look for that solid foundation than a mature communications technology?

2. Recent Architectural Concepts

Many of the concepts and approaches which were valid twenty years ago must give way to improvements achieved by technology initiatives. The practice of Information Systems Architecture today bears little resemblance to the classical, hardware-oriented discussions above. Today, it is clear that system performance parameters are the realm of neither the hardware engineer nor the software engineer to the exclusion of the other. Performance is the performance observed by the user, and the user is disinterested in aesthetic considerations of hardware, software, firmware of any kind of "ware." The user is interested as he should be in holistic cost-performance trade-offs.

The International Standards Organization (ISO), in its Reference Model of Open Systems Interconnection (OSI), has made a major contribution in the development of principles for network architecture. This model, through its definition of seven layers, and their attendant protocols and interfaces, has provided a framework in which modern systems architects may communicate. This model, though originating in the communications world, represents a basic departure point for the development of C^2 architecture. As has been often recognized, without communications there is no command and control; hence C^3. The basic question of how to communicate is integral to system development and must accompany (preferably precede) the "what" or substantive nature of information systems design. It is in the nature of this "flow" that many of the current problems of interoperability, security, survivability, and networking can be broached in a logical manner.

a. Description of the ISO Protocol Model

The International Standards Organization (ISO) reference model for Information Systems Architecture is shown in Figure 2. Advanced information systems have architectural structures, in which different functions are realized in different layers or levels of that structure.

The structure can be visualized in terms of layers of concentric ellipses, as in the figure. The layers can be thought of as the layered skins of an onion. Each layer improves the generality of function and the usefulness of the components of the architecture. In telecommunications, distant machines are interconnected. The use of communication links in information systems has heavily influenced the arrangement of the structure, and affected designs intended to make the work of users easier.

The ISO model has seven layers. Following the ISO terminology, we refer to these levels as 1, 2, 3, etc. Many commercial computer networks for distributed processing contain a subset of these levels, although details and names differ from system to system. Standards organizations have formalized the structures of levels 1, 2, and 3 with growing acceptance by the using Community.

The layers shown in the figure are described in the following sections.

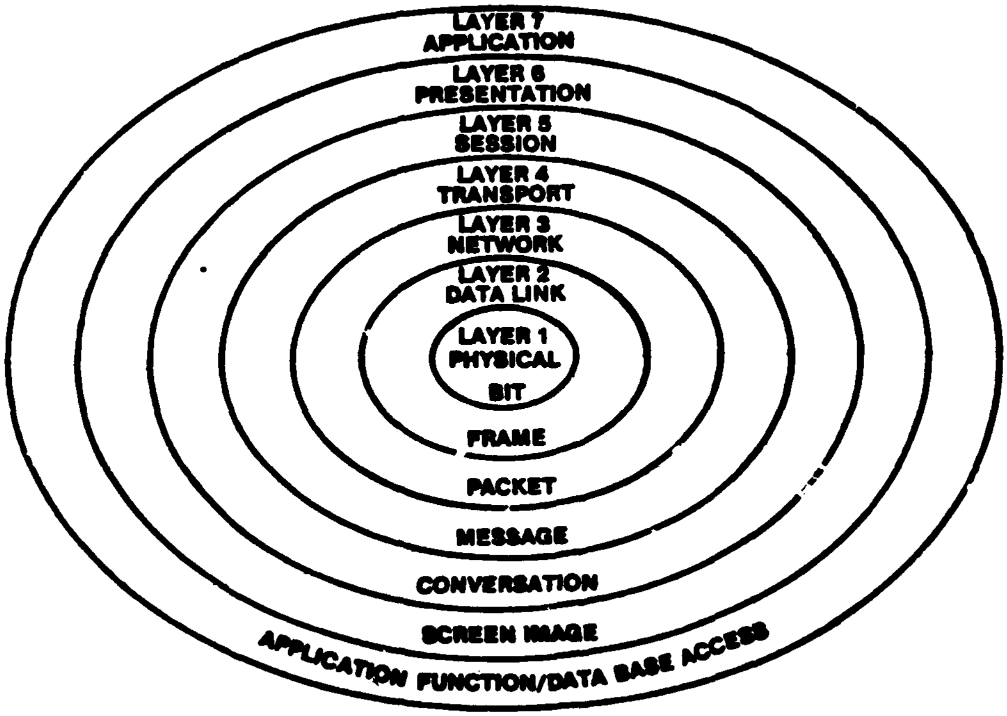


Figure 2. Open Systems Architecture

1) Physical Layer

It takes two to communicate. The layers occur in pairs (or tuples). Each layer has a complementary layer at some remote place which actively cooperates reciprocally with it. The Physical layer, the lowest layer, is concerned with the physical, electrical, functional and procedural characteristics to establish, maintain and disconnect the link.

The Physical layer is concerned with the flow of 0 and 1 bits back and forth between computers, and between computers and terminals and other devices. The electrical and mechanical characteristics of the bit string are determined in the physical layer. For example, the voltage levels representing 0s and 1s, the speed of transmission in baud or Hertz and its directionality (simplex or duplex) are specified in the Physical layer. The nature of the physical communications facility, such as the number of leads in a cable, their transmission and signaling functions, and the standardized assignment of network connector pins to those leads, are also specified in the Physical layer.

The two standards most frequently used to describe the Physical layer are the EIA RS 232-C modem interface standard in the United States, and its European equivalent CCITT (Consulting Committee International for Telephone and Telegraph) standard V.24 for analog links; and the new CCITT standard X.21 for digital links.

2) Data Link Layer

The Physical layer transmits and receives individual bits that have no meaning or structure beyond the fact that each bit is either 0 or 1. The Data Link layer accumulates bits together into a frame, the basic unit of information exchanged between any two nodes of the network. The purpose of the Data Link layer is to provide mechanisms so that the starting and ending points of the frame are well defined; transmission errors can be detected and corrected; destination addresses are provided; and sequence of transmission and receipt controlled so that multiple nodes can share the communications utility. Transmission and receipt of a frame implies all of these functions, and their realization is a physical data link control procedure.

Early line links were asynchronous and start-stop bits were used to establish the beginning and end of a character. More sophisticated and faster transmission techniques evolved, employing synchronous transmission; that is, where two or more clocks at different points in a network are synchronized to effect frame exchange. Binary synchronous (BISYNCH) line control is a popular IBM version of synchronous link control. Special start-of-header, start-of-text, and end-of-text unique bit sequences were used to signal the beginning and end of frame conditions. Today's frames employ a special unique character called flag bits to establish transmission boundaries. Frames also include address bits, error detection bits, and other control bits grouped at the beginning and end of a frame in a "header" and "trailer." A frame consists of these control mechanisms, plus a packet embedded between the header and trailer. A packet is a bit string with a structure meaningful to Level Three (and higher levels) of the protocol. From the point of view of the Data Link layer, only the header and trailer bits have meaningful structure; the packet bits do not. The Data Link layer administers the physical link, using the header and trailer bits as control mechanisms to perform its functions. See Figure 3 for a summary of the discussion above.

The Data Link layer sends frames back and forth between nodes, both those containing substantive information as messages, and those which simply acknowledge proper (improper) receipt of substantive information.

From the point of level three (the Network layer) all transmissions occur flawlessly. In reality, noise can destroy a frame. It is the responsibility of the Data Link layer to recognize this and take remedial action. The Data Link layer uses mechanisms such as cyclic

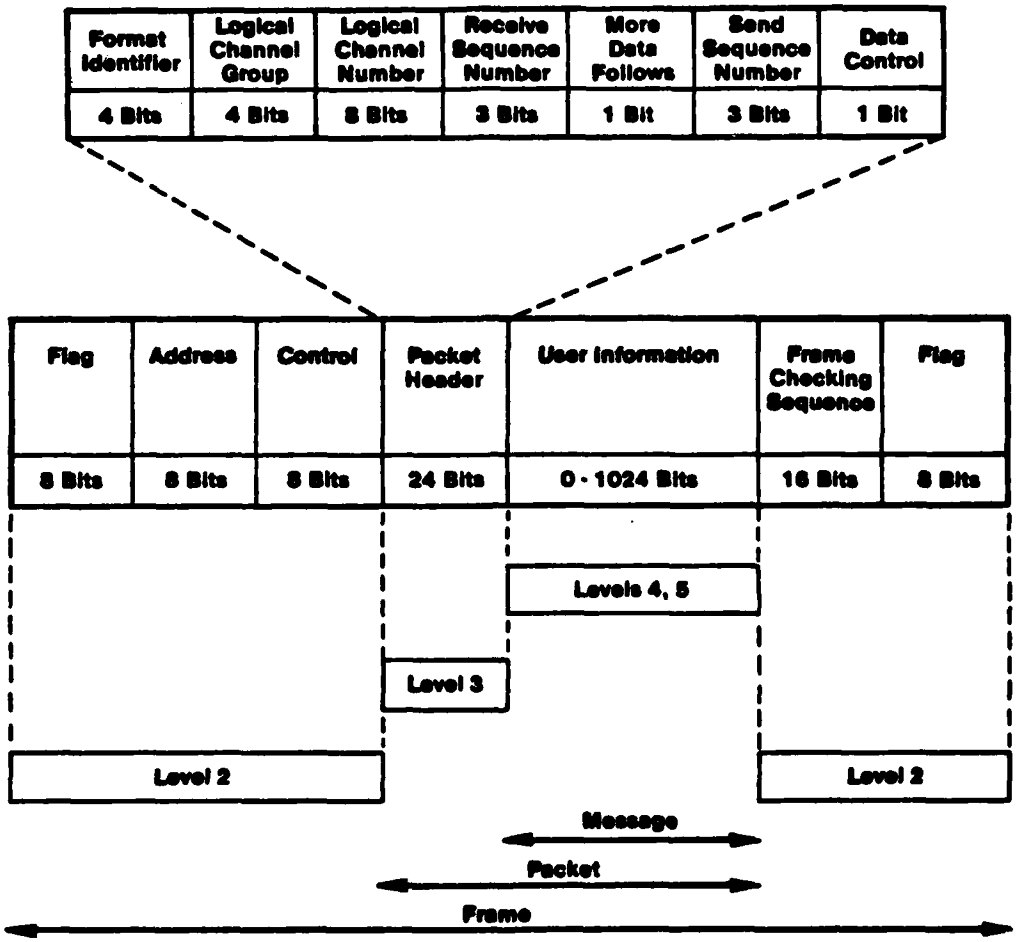


Figure 3. Control Level Mechanisms

redundancy check bits to verify that noise did not destroy one or more bits in a frame. If this happens, the frame is retransmitted until it is correct. If noise persistently perturbs transmissions, the line handler downs the line until it an be repaired, and traffic over that particular link is terminated. Queued frames waiting for transmission find other paths through the network. Level Three considers that it is working with error-free lines, since Level Two passes only correct packets to Level Three. An error-free line is often called a virtual line (or a virtual circuit) because no real line behaves as well.

Specific implementations of Level Two (Data Link) protocols can provide additional services for Level Three, each increase in quality of service at a price. A satellite link justifies greater handling expense than a short twisted wire pair. Examples of services beyond the basic ones described above include speed and code conversion, static and dynamic packet buffering, error correction bits, duplex transmission, etc.

The ISO recommendation for the Level Two protocol standard is the High-Level Data Link Control (HDLC) procedure. HDLC is similar to another widely implemented protocol, the Synchronous Data Link Control (SDLC) procedure, designed by IBM.

3) Network Layer

Before 1975, access of a centralized computer to a population of remote terminals could be accomplished by the mechanisms of the first two layers by themselves. With the injection of microprocessors into terminals and all the other devices that constitute computing networks, increasingly sophisticated and complex distributed processing control was required, and additional layers of protocol were created, beginning with the Network layer.

The Network layer routes packets within a network (sub-network). The Data Link layer strips the header and trailer bits from the frame, presenting just the packet portion of the frame to the Network layer. The Network parses the header of the packet from the remainder, leaving just the message portion. The structure of the message portion of the packet is meaningless to the Network layer. Its structure is meaningful only to higher levels of the hierarchy. The Network layer control mechanisms use the packet header to route the message to its proper destination in the proper sequence, just as the Data Link layer uses its frame header to do its work. The packet header contains logical channel address, send and receive sequence numbers and other control information. An illustrative packet header format is shown in Figure 3.

The essential task of the Network layer is routing of packets through the network. There are many different degrees of sophistication of routing algorithms which have been implemented in various networks, ranging from fixed, static table look-up, to dynamic routing as a function of instantaneous traffic loads with local congestion monitoring. accounting and costing of traffic charges are also part of the functions of this layer.

The Network layer manages virtual links (sometimes called virtual lines), virtual circuits, or alternately, logical lines and logical circuits. As the name implies, these facilities do not exist in reality, although they are mapped onto the real physical facilities of Level Two. From the point of view of Level Four and higher levels of the protocol hierarchy requesting services from the Network layer, the virtual links are "real" enough. However, the physical links may be considerably different from the virtual links. For example the physical link may consist of several different segments, while the virtual link consists of only one. The different segments may have different characteristics, consisting of both terrestial and satellite portions. This is what distributed processing means.

In a dynamic network, the multiple packets which contain an integral message can take different routes because packet routing is a function of load, which varies from instant to instant. The pieces of data in various packets have to be assembled together to form the complete message at the destination. The reassembly process includes resequencing of the packets. Since the packets have potentially travelled different routes (of different lengths), arrival out of sequence occurs frequently. The Network layer (the third level) performs these services for its clients, the higher levels.

Many PTT (Post Office, Telephone and Telegraph) networks, such as TELNET, TRANSPAC, DATAPAC and EURONET use the CCITT X.25 network layer protocol as standard. This standard is finding increasing acceptance on a world-wide basis for global telecommunication networks.

4) <u>Transport Layer</u>

The Transport layer was the last layer to emerge from architectural studies of layered protocols, even later than the layers which are "higher" in the hierarchies. In early designs, such as ARPANET by the Defense Advanced Research Projects Agency (DARPA), DECNET by the Digital Equipment Corporation (DEC), Systems Network Architecture (SNA) by IBM, AUTODIN I by the DoD, the initial protocol specifications promulgated by the Consulting Committee International for Telephone and Telegraph (CCITT) in Europe, and many others, the emphasis was on specification of a system-wide Network protocol. This basic approach is natural enough, but it ignores the realities that there are many large institutionalized networks (some with large institutionalized data bases) which pre-date formal protocol definition.

Real networks fall short of the theoretical ideal implied by the older models cited above. Real networks can be partitioned into sub-networks, each with its own set of peculiarities. A large computing system with multiple distributed computers and long range communications capabilities normally speaks several languages. This is because as a practical matter, commercial systems use more than one type of link, more than one type of terminal, and more than one type of computer, each with different characteristics. A typical network grows in time, adding new users with new equipment and new requirements. Occasionally, different networks are married to realize economies of scale and for other reasons. Therefore, a typical network really consists of interconnected sub-networks, each sub-network with different communication links with different sets of control requirements. The Transport layer handles translations between sub-networks. The latest protocol standards, such at the ISO model described here, and AUTODIN II, include an explicit Transport layer, whose function is to provide the same sort of network management capability on an inter-network basis that the Network layer provides an intra-network basis.

The Transport layer provides service to the Session layer above it. Again, the degree of sophistication can vary widely. The simplest service is end-to-end point-to-point channel management. In addition to this basic function, there are more advanced transport functions, such as: broadcast to multiple receivers, multiplexed multiple connections to serve a single session client with high bandwidth requirements, and the opposite, multiplexing several sessions clients onto a single expensive inter-network link resource, etc.

The Transport layer often is implemented in a large host, and can be thought of as part of the resident operating system or control program. In this sense, the Transport layer provides (or causes the operating system to provide) multiplexing of several simultaneous message streams to and from the Session layer. In contrast, the Network layer usually is implemented as a link driver extension to the operating system, with close ties to the inter-network link, and without multiplexing capabilities, or with limited multiplexing capabilities.

The first three layers have implementation disciplines which are well understood and widely observed on an international basis. The Fourth, or Transport layer, has many diverse implementations, no one of which can be said to enjoy sufficiently wide recognition to merit serious claim as a standard today.

5) Session Layer

The Session layer negotiates a connection between one user (or an applications program acting as surrogate for the end user) and another user (or another software process on some other computer). The connection between two users (more specifically, between two Presentation layer software processes) is a session.

As the name of the layer suggests, the Session layer is concerned with the establishment, maintenance, and termination of services related to communications between two (more more) clients. One client at one end of a session can be a thoughtful human being, thinking at a terminal and pausing to consider his next action, while the other client at the other end of a session can be a data base manager, concerned with optimizing response times in response to ad hoc query. It is apparent that the Session layer construction is sensitive to the context of the on-going dialogue.

Consider, for example, a line printer connected to a communications link. It is the responsibility of the two communicating Session layers to be aware of the performance characteristics of the line printer (lines per minute capacity) so that the device is driven at its rated speed, not faster, and not slower. The active cooperation between two otherwise-independent Session layer software packages to accomplish some common mission, such as provided by this example, is called "binding" of the two processes.

Another example of Session layer functions is provided by transactions directed toward a data base which results in modifications to data or structure of the data base. If there is any failure of the session for any reason, the stimulus to the data base cannot be aborted halfway through a process so that the data base is left in an inconsistent and non-recoverable state. Application-sensitive transactions are bracketed, so that recovery from failure during a processing sequence can be effected.

In classical large main frame systems supporting interactive COBOL task programs and administered by sophisticated multiprogramming operating systems, core limitations often constrain the degree of concurrency and response times that can be obtained. Session layer functions are *sometimes especially designed to minimize* core residence requirements during the long delays normally encountered in interactive situations because of human think and reaction times. Significant improvements over classical system performance can be accomplished in this way with relatively little effort.

6) Presentation Layer

The Presentation layer, as its name implies, is concerned with the format of data presented to the user (or Application layer programs acting as surrogate for the user). The Presentation layer is responsible for the appearance of the data to the user. Data compression-decompression provides an example of a Presentation layer function. Much message, tabular and pictorial information contains a great deal of entropy, to use information-theoretic language. Efficient encoding of information at the source and decoding at the using end can effect significant savings in Transport bandwidth. Of course, the encoding-decoding mechanisms in the Presentation layer have their own expense associated with them. The consideration of data compression-decompression leads immediately to another possible Presentation layer service, that of encryption-decryption to provide security. In information-theoretic terms, encryption-decryption can be viewed as a special case of encoding-decoding compression-decompression techniques.

In general, different computing systems and their users have incompatible file formats, incompatible CRT screen formats, and incompatible line and character formats. Transformations to resolve these differences, so that the end result is meaningful and attractive to the user, are the responsibility of the Presentation layer.

7) Application Layer

The Application layer is a catch-all name which designates any application programs written by a user to accomplish any tasks. In a distributed-processing environment, the objective of establishing an Application layer in an architectural framework is to emphasize the fact that few application programs are stand-alone. Typically, application programs operate in cooperation with a communicating application program in one or more other machines, and it effects that communication by some agreed-on rules called protocols. It does not matter so much what those protocols are; what matters is that there be some clear, easily-understood protocol. The set of protocols, together with the set of application programs, constitutes the Application layer.

It is at this level that much of the assumed architecture of today's C^2 systems has occurred. Systems have been developed to satisfy a given application on a given set of hardware with little attention paid to the ability to communicate physically or substantively with other applications. The definition of this Seventh Layer directly relates to the problems articulated in other portions of this study; e.g., requirements definition, system interaction, etc. This Seventh Layer, however, must be defined only in the context of solution of the other layers.

3. Information Systems Architecture Concepts

Information Systems Architecture may be defined as the logical structure of hardware, software and communications needed to facilitate information exchange between user nodes. This structure should permit and encourage the introduction of new technology within each sub-element (hardware, software, and communications); provide the flexibility for system reconfiguration; and allow for continued expansion as requirements evolve.

An architecturally sound information structure is sufficiently flexible so that it admits introduction of new technology into existing ADP systems without radical and wholesale disruption of existing operations. This development philosophy and architectural viewpoint requires continuing restatement and validation in terms of emerging client requirements.

Figure 4 shows a classical view of individual and specialized command-oriented ADP development. Each project "reinvents the wheel" at each level. This approach results in duplication of functional development. Even more important, the individualized, fragmented approach almost guarantees that individual nodes of the command and control network will not work together harmoniously. If it is desirable to promote the goals of interconnectivity, interoperability and survivability of function across a self-healing network, then certain elements, namely those elements supporting communications, must be standardized. More specifically, the first three levels of the ISO model for Information Systems Architecture need to be standardized. These concepts are suggested graphically by Figure 5.

There are both specialized command-unique programs and application programs with sufficient generality to fit across multiple command locations. Figure 6 illustrates graphically that there is room for both types of project development in the architectural reference framework provided by the ISO model.

There are many ways of looking at current trends in computation and communications. While today's technology trends provide hints of much more powerful approaches to problem definition, algorithm design, and program computation, only the outlines of these thrusts are clear at this time. New results are just now emerging from current research. Research into information systems to support C^2 is not an accomplished and finished thing. Nor is it likely that it will ever be an accomplished and finished thing.

In the final analysis, the holistic concept of an Information System Architecture is of vital importance because it provides the flexible framework that permits introduction of new technology (and there will always be new technology) into existing ADP systems, without radical and wholesale disruption of existing operations. Information Systems Architecture means creation of sufficiently comprehensive information structures, built upon computing-communications disciplines, that progress in one area is not made at the expense of another area. Figure 7 suggests that the role of Information Systems Architecture is to "put it all together," to create a framework uniting the vital technology components of information systems--communications, processors, processes, data bases, the human users--into a common structure, just the way that construction architects combine the different parts of civil structures--communications, hardware, air, electric and gas utility flows--with their human users.

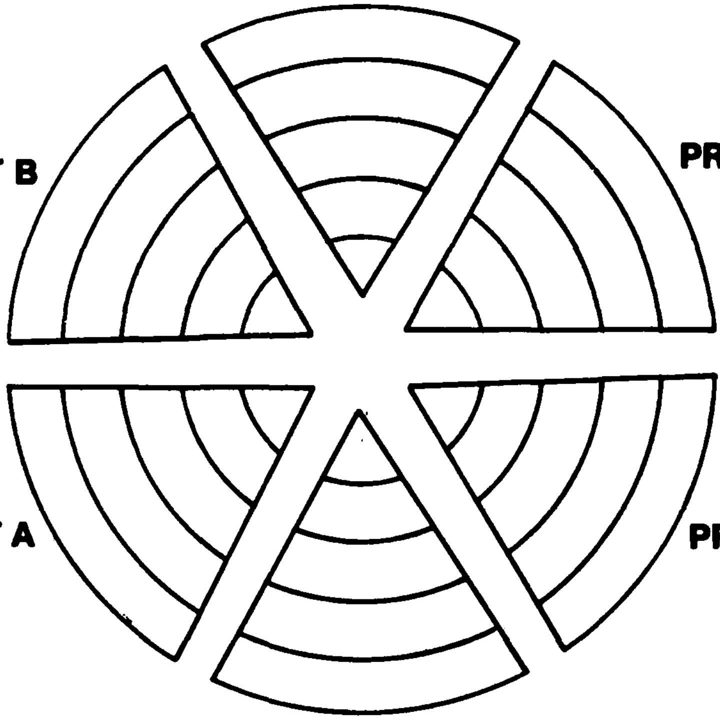


Figure 4. Without Information Systems Architecture - Reinventing the Wheel

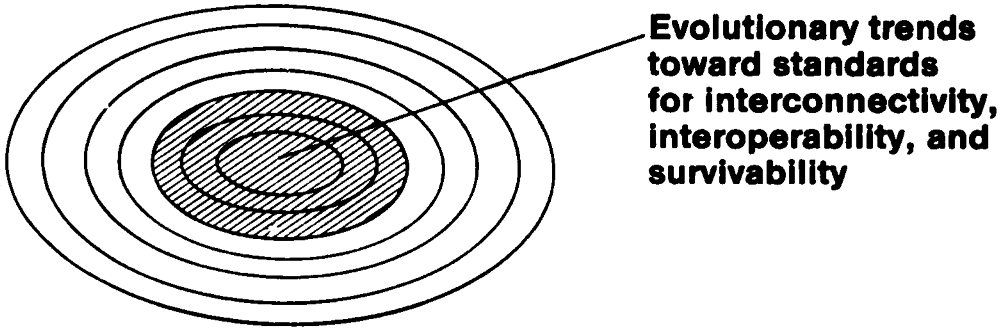


Figure 5. Information Systems Architecture Conceptual Framework and Basic Vocabulary

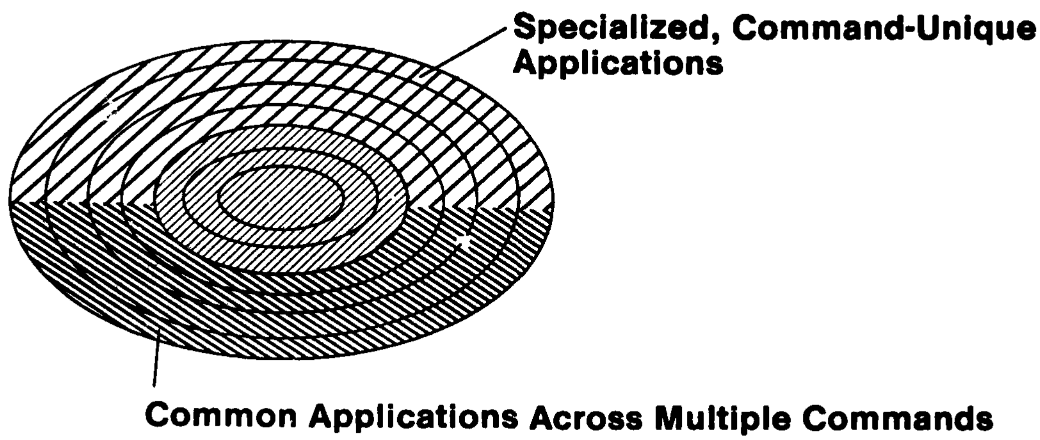


Figure 6. Information Systems Architecture the Basis for a Variety of Applications

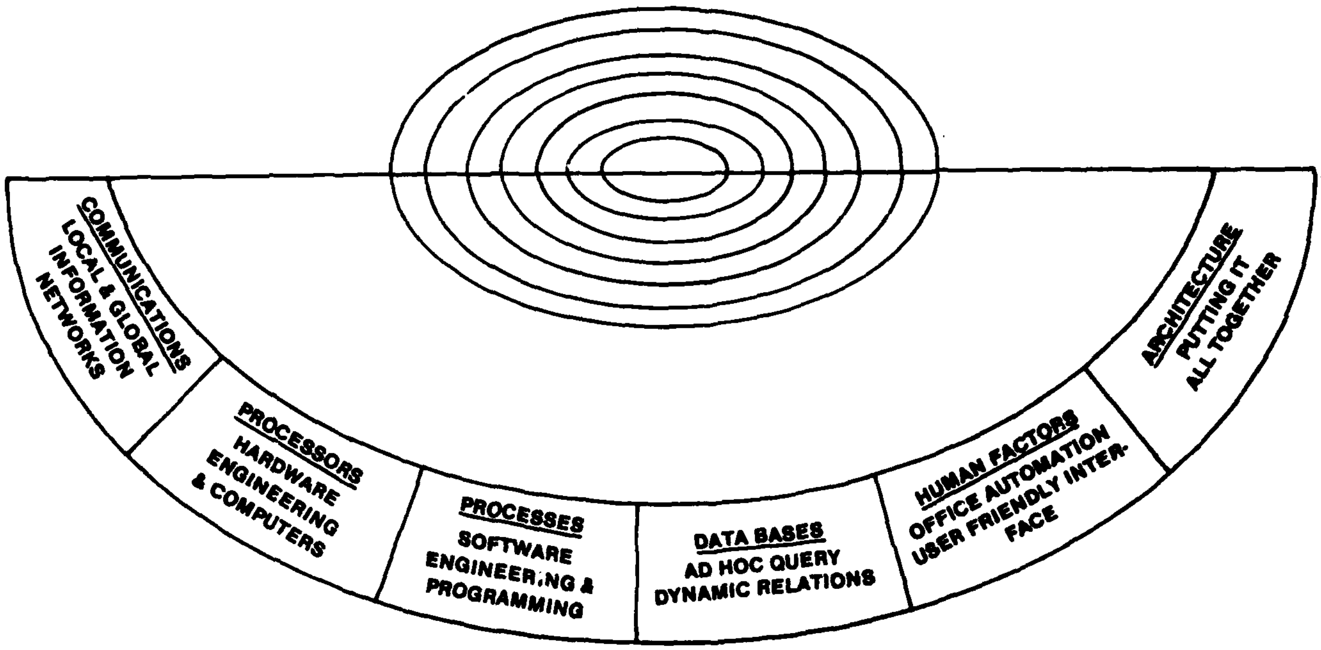


Figure 7. Information System Architecture The Vital Technology Components

C. THE REQUIREMENTS FOR INFORMATION SYSTEMS ARCHITECTURE IN COMMAND AND CONTROL

1. Command and Control Perspectives

Early automated command and control systems were oriented toward a nuclear holocaust environment. They were oriented toward detection of hostile aircraft, interdiction, and counter offensive measures. The time constraints in those early days were beyond the means of existing manual command and control systems and, therefore, automated aids were developed. These time constraints became more stringent with the advent of missile delivery capability. The development of command and control systems was aimed at meeting this threat, which can be typified as a short, violent nuclear exchange of a one- or two-strike character. The information requirements called for immediate threat assessment decision making at the NCA level, commitment of resources and subsequent delegation of operations.

As time progressed, it became evident that command and control systems must be developed to deal with a much greater variety of combat situations. Viet Nam showed that the US must be able to deal with limited, non-nuclear confrontations more akin to a World War II environment, but utilizing technology advances in weapon systems and communications. It further displayed the extreme need for interoperability between DoD Component systems. The information requirements for this type of conflict were typified by rapid threat assessment, continual force deployment and status information, and long-term logistics planning. The level of commitment authority was elevated due to the political environment and the increase in communications capability.

As it became evident that there existed both the threat of all-out nuclear war and limited non-nuclear confrontations, planners also realized that a need existed to be able to deal with "limited" nuclear war concepts. This environment dictated the need for more enduring command and control systems, whose capabilities provided prolonged nuclear crisis support. Information requirements take the form of rapid multi-threat assessment, communication and information exchange among highly-mobile, dispersed activities, and delegation of operational control.

The three scenarios depicted above have evolved over time and have defined a command and control systems requirement of rapid-to-immediate threat assessment, communication from the battlefield to the NCA, operational direction from varying levels in the command structure, constant monitoring of operational and resource allocations, and transmission of decisions to a variety of command echelons.

2. The Command and Control Systems Requirement

The description above depicts a many-faceted problem further complicated by geographic, political and environmental considerations. The United States cannot count on having numerical superiority in military personnel and weapon systems. This puts premium value on any characteristic of C^2 systems that acts as a force multiplier to reduce this disadvantage. United States force endurance and survivability can be improved by C^2 systems with improved endurance and survivability. C^2 ADP support systems designed for modularity and redundancy within a self-healing network are perceived to be a means of accomplishing that goal. Recognition that prolonged nuclear or non-nuclear crisis is realistic makes justifiable the requirement for survivable C^2 systems that enable commanders to observe the enemy with more timely and more sustained data collection over longer times and larger areas. C^2 developments that have, as design objectives, continued systems life during prolonged nuclear crisis and loss of individual nodes are intended to enable US commanders to observe the enemy and assess operational patterns in order to exploit enemy vulnerabilities and endurance breakdowns. The objective is to provide the commander with support tools to enable him to outplan and out-maneuver the enemy under prolonged crisis conditions.

To summarize, there is growing recognition that C^2 ADP systems with improved warning, strike support and force management before, during and after prolonged crisis periods can be justified in terms of their force multiplier potential.

3. Networks for Interoperability, Connectivity and Survivability

The management aspects of this subject are emphasized in this discussion, although reference to the technical underpinings is included to assist the discussion.

a. The Architectural Problem

As a practical manner, many of today's large information systems are clusters of computing elements developed by different people with different perspectives at different times, which are then connected together to extend the range of the individual computing elements. The interconnection of disparate computing elements in a "marriage of convenience" often is motivated by the desire to extend the range of the individual capabilities developed at one node in a network (often at considerable expense) to other nodes in the network.

b. The Architectural Solution

In designing practical (rather than monumental) buildings and other civil structures, no architect sets out to create a design "from scratch." Instead, the architect selects from among the designs of buildings and structures erected for similar purposes which have proven effective in the past. Having chosen one which satisfies a current need, the architect customizes it as little as possible.

Similarly, a construction engineer does not fire bricks nor mill steel to his specification. He does not commission the manufacture of custom elevators, mail chutes or anything else, if he can help it. He obtains these items, made to standard specifications, from proven manufacturers.

Architects and construction engineers are able to obtain the considerable savings and work productivity which the use of mass-produced materials and components permit because the construction industry understands the virtues of standardization, because material and component designs are created with general-purpose use in mind, and because plans and specifications are sufficiently standardized so that they are widely understood and accepted.

Since Information Systems Architecture is a less-mature discipline than its civil engineering counterpart, the realization of these basic ideas is not as well developed. Applied computer science has not matured to the point where there are clean definitions of terminology and well-accepted divisions of responsibilities between practitioners at different levels of design and implementation responsibilities.

There are no universally-accepted definitions of the various technical products generated in ADP application, such as "requirements definition," "functional description," "performance specification," "system specification," "subsystem specification," "program specification," "maintenance specification," etc.

The absence of well-understood and widely-observed professional discipline in specific technical areas also is reflected in program management planning. Realistically, then, the idea of a DoD Information Systems Architecture, well integrated by an architect in accordance with a carefully conceived program management plan, is a conceptual goal and objective, rather than an undertaking to be implemented "today" with a well-understood and predictable outcome. This means that existing systems, with their limitations and imperfections, are "bricks" in an Information System structure. Pragmatic judgements (and compromises) must be made regarding which "bricks" should be "standardized" in order to deliver essential services at least cost and least risk.

The basic intent of Information System Architectural planning is to specify an overall design by isolating and exploiting ADP commonalities (such as tasks, procedures, data structures, data) between specific, command-oriented functional areas which lead to specific threat awareness. Motivating reasons for looking for such commonalities include economy and development risk minimization. However, the most compelling reason for looking for commonalities is to insure interconnectivity and interoperability of the nodes in the network, so that there can be efficient data interchange in the delegated production environment.

It is becoming increasingly obvious that in wartime, distinctions between civil and military communications capabilities are irrelevant. In wartime, civil communications needs must be subordinate to military communication needs. Based on this perspective, interconnectivity and interoperability assumes even greater and more universal significance. One system-wide way of looking at interconnectivity and interoperability is survivability of the network as a whole. Redundancy of capabilities at different nodes enables recovery when one or more nodes are lost for any reason, when the remaining nodes can be configured so as to adapt to the loss and take over the lost function. A self-healing network is the logical "last step" of interconnectivity and interoperability in a delegated production environment.

D. EFFECT OF EVOLUTIONARY ACQUISITION ON INFORMATION SYSTEMS ARCHITECTURE

It is difficult to determine whether or not evolutionary acquisition is new to information systems or just that DoD's recognition of an on-going-process is new. To date, information system development policy has followed the classic process of: (1) concept formulation; (2) system design; (3) implementation; (4) test and evaluation; and (5) installation and operation/maintenance--all in a serial fashion. While DoD acquisition policy provides for this serial process, experience has shown that actual development of C^2 systems should be iterative. C^2 system specifications tend not to be static but rather change as implementation progresses. The rationale for these changes is not subject of this appendix, but their existence is important to the development of a C^2 systems architecture strategy. This strategy must provide the latitude for introduction of change, whether it be technological or functional.

The traditional view of system acquisition as a serial, non-iterative, "arms-length" process has led to the conclusion that C^2 systems are most-often behind schedule and usually overrun in cost. To avoid these criticisms, many systems have "buried" system iterations in latter phases the life cycle. Such terms as "enhancements," "releases," or "improvements," buried in the maintenance phase oftentimes are, in reality, major system developments that can, in fact, reflect changes in the original specification of requirements.

It is incumbent upon any C^2 system architecture to recognize and to provide an environment for evolving systems. An architecture that is not flexible enough to accommodate changing needs cannot exist successfully in the C^2 world. Additionally, DoD must recognize the iterative nature of requirements statements and system implementations thereby planning realistically for systems introduction.

User involvement in the development of an Information Systems Architecture is of paramount importance. Evolutionary acquisition will result in varying responsibilities between the developer (provider) and user, dependent upon the nature of each individual system. In some cases, the user will play the dominant role in system development from requirements definition through physical implementation. In other cases, the user may be more heavily involved in the requirements definition, with little or no involvement in development. The meshing of these various systems will occur with the user and it is here that operational effectiveness truly can be measured. The using command must, therefore, have a plan for introduction of new capabilities--an architectural design. Further, evolutionary acquisition connotes the continual upgrading of existing capabilities. The lack of a plan for this evolution can lead to utter chaos for the user.

The user is an integral part of architectural design and, therefore, must either provide or be provided the talent to accomplish the necessary tasks. In this environment, there is a natural hierarchy of users that must be considered. For example, if we look at the peacetime operational chain of command for Europe we note the following:

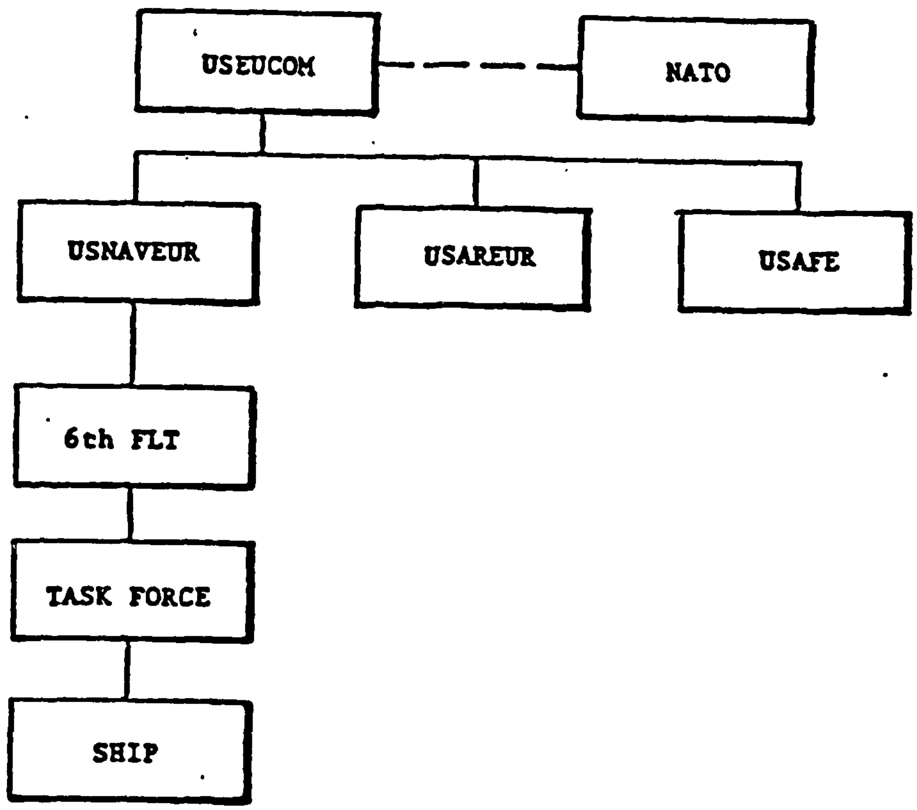


Figure 8. Sample Operational Chain of Command for US Forces in Europe

While this represents a simplification of command interaction, it highlights the problem of having to accommodate a hierarchical chain of requirements, even though development may occur outside the chain. USEUCOM has a requirement for exchanging information with its components; for example USNAVEUR, who in turn exchanges information with the Task Force who, in turn, interfaces with the individual combatants.

Each of these echelons must be capable of specifying its information needs in terms that recognize the requirements of other portions of the chain. Each integral part of this chain must be able to introduce new systems that recognize the presence of this hierarchy and interoperate within it. Each user level requires a definition and plan for introduction of information systems capability. In this example, HQ USEUCOM preferably, would define needs for the European theater and the concomitant responsibilities for the Component commands. Subsequently, the Component commands would develop their plans in concert with this master plan and based on projected systems introduction dates. These plans then would become the basis for an architectural design at each level of command.

Summarizing: What, then, should the Command and Control community do to advance its objectives? It should endorse the first three levels of the ISO model and take a firm stand on a single standard for each of these levels. Further, DoD should take a lead role in establishing interface and protocol standards for the higher levels in the hierarchy.

APPENDIX I

DEFENSE ACQUISITION MANAGEMENT ORGANIZATION

DEHNER, MATHIAS, MCILVAINE & TIMMONS
(REPRINTED FROM PROGRAM MANAGER, MAR-APR 82)

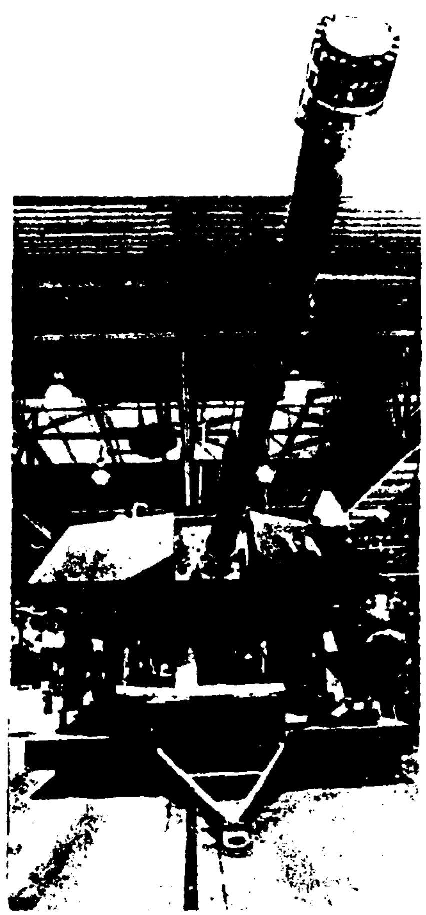
DEFENSE ACQUISITION MANAGEMENT ORGANIZATION

Compiled by Major Frederick T. Dehner, USAF; John R. Mathias; Paul J. McIlvaine; and Commander David R. Timmons, USN

Program management can be described as the timely, systematic, and intensive integration of diverse functional activities to achieve a coordinated concentration of resources on the objectives of a specific task. Within the Department of Defense, major categories of tasks to which program management techniques are applied are the development, acquisition, and logistic support of weapon systems and subsystems. These tasks are accomplished within the military departments in complex and detailed organizational scenarios.

This article provides a summary of how each individual military service has implemented DOD systems acquisition policy and guidelines—including key organizations and participants, functions, documentation, and management review procedures. It quickly portrays the similarities and differences between the Army, Navy, Air Force, and Marine Corps. It attempts to remove the cumbersome detail normally associated with bureaucratic organization charts. Hopefully, the reader will be able to focus on the essential elements of information about the acquisition organizational environment.

The selected organizational element charts for the individual services graphically depict variations between services concerning the approach to acquisition management. The U.S. Army has one major command, the Materiel Development and Readiness Command (DARCOM), accomplishing most of its development, acquisition, and logistic support functions. However, within DARCOM, materiel acquisition is accomplished primarily at the major subordinate command (MSC) level, generally identified as either research and development (R&D) commands, materiel readiness (MR) commands, or in combined commands where both the R&D and MR responsibilities for selected materiel are assigned to a single command.



The U.S. Air Force has two major commands involved in its system acquisition process. The Air Force Systems Command is responsible for development and acquisition of weapon systems/subsystems. Air Force Logistics Command provides logistics support for the weapon systems/subsystems. The U.S. Navy has one major command dedicated to the system acquisition process, Naval Material Command (NAVMAT). Within NAVMAT, there are subordinate systems command (SYSCOMs) that accomplish the development, acquisition, and logistics support for Navy systems/subsystems.

The U.S. Marine Corps predominantly obtains its weapon systems/subsystems through the Army and Navy and, in some instances, the Air Force. Within the Marine Corps, the Marine Corps Development and Education Command (MCDEC) at Quantico, Va., manages the development and Headquarters, Marine Corps, manages the acquisition and logistics support functions. Since the Marine Corps is dependent on the other services for most of its weaponry, it has established an Acquisition Coordinating Group at Headquarters U.S. Marine Corps level to perform the function of program management and to assure that Marine Corps weapon system program needs are satisfied.

The last two matrix charts provided in this article are a presentation and comparison of selected key management features, practices, and focal points within the military departments for acquisition management.

The compilers of this supplement are professors in the Defense Systems Management College School of Systems Acquisition Education. Major Dehner, Mr. Mathias, and Commander Timmons teach in the Policy and Organization Management Department. Mr. McIlvaine teaches in the Technical Management Department.

Figure 1 **PRINCIPAL MATERIAL ORGANIZATIONS (MAJOR COMMAND LEVEL)**

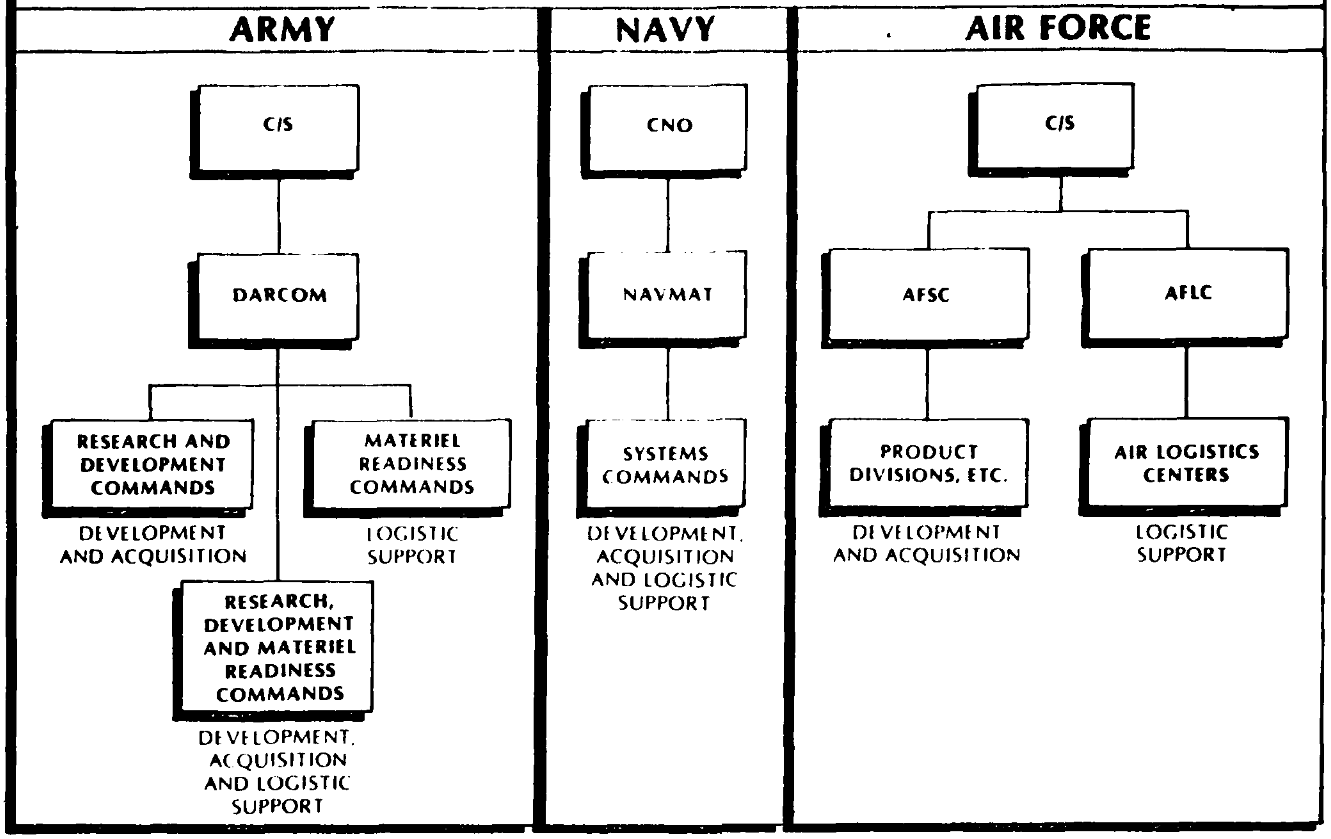


Figure 2 SELECTED ORGANIZATIONAL ELEMENTS
U.S. ARMY ACQUISITION PROCESS

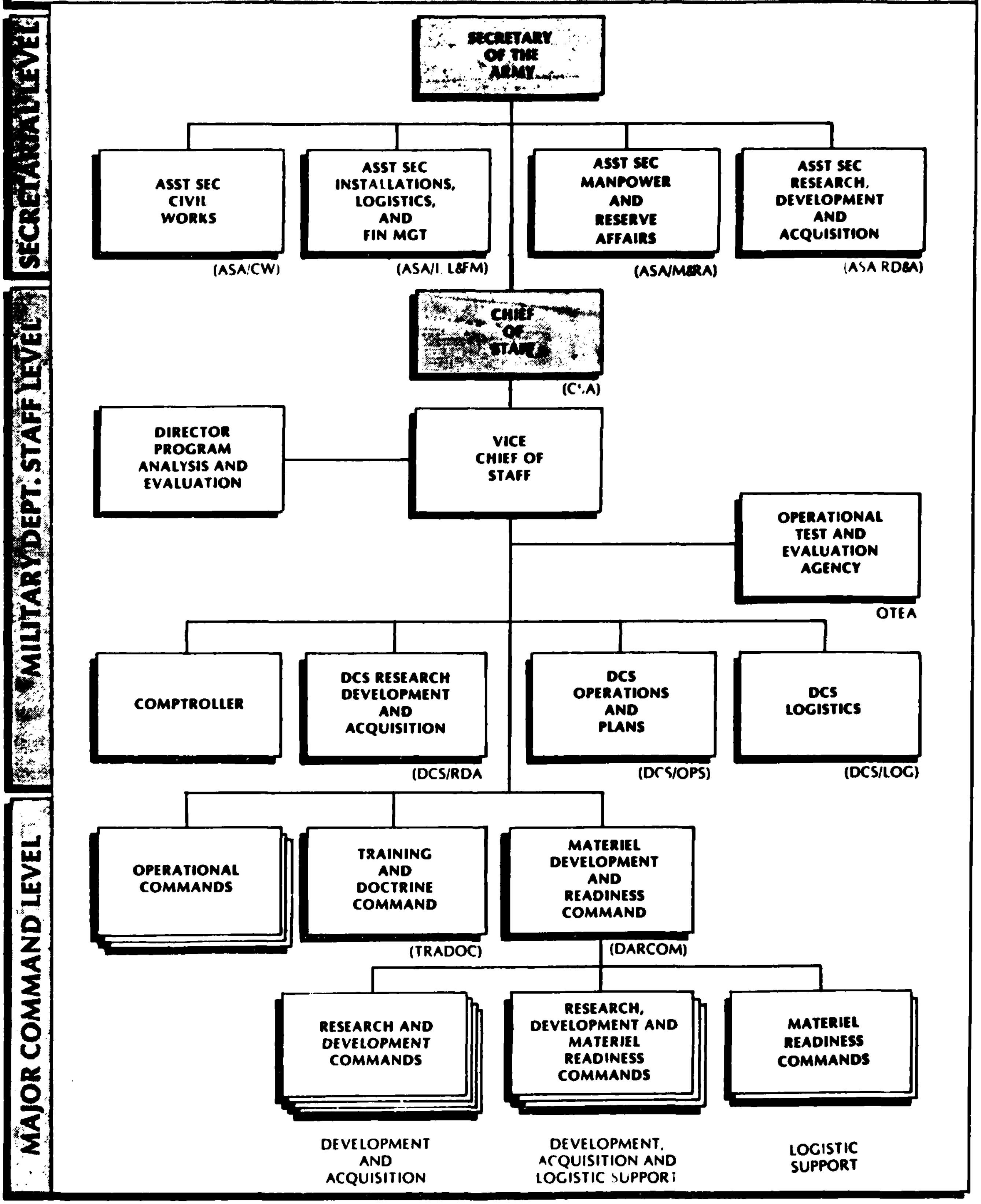


Figure 3 SELECTED ORGANIZATIONAL ELEMENTS
U.S. NAVY ACQUISITION PROCESS

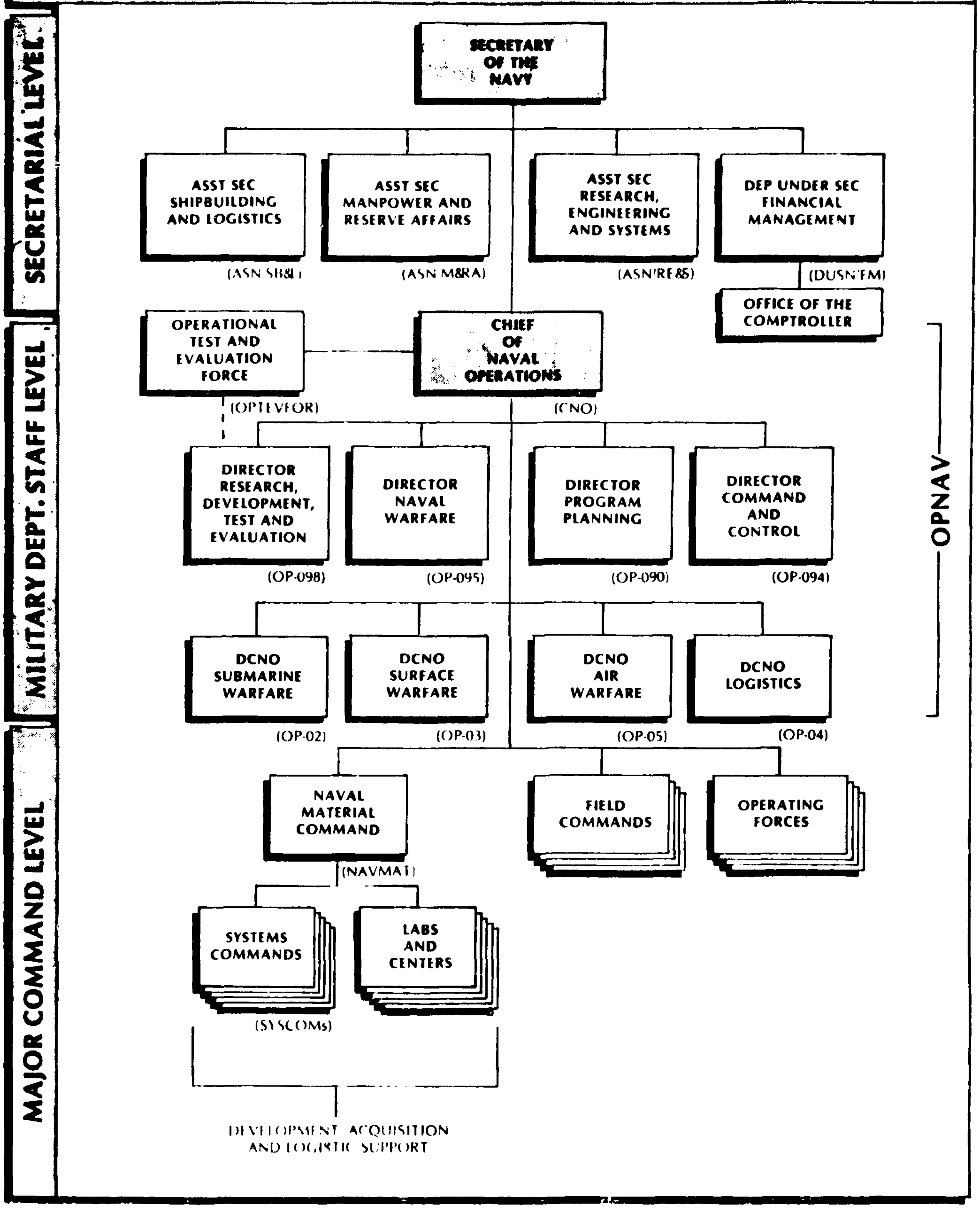


Figure 4 SELECTED ORGANIZATIONAL ELEMENTS
U.S. AIR FORCE ACQUISITION PROCESS

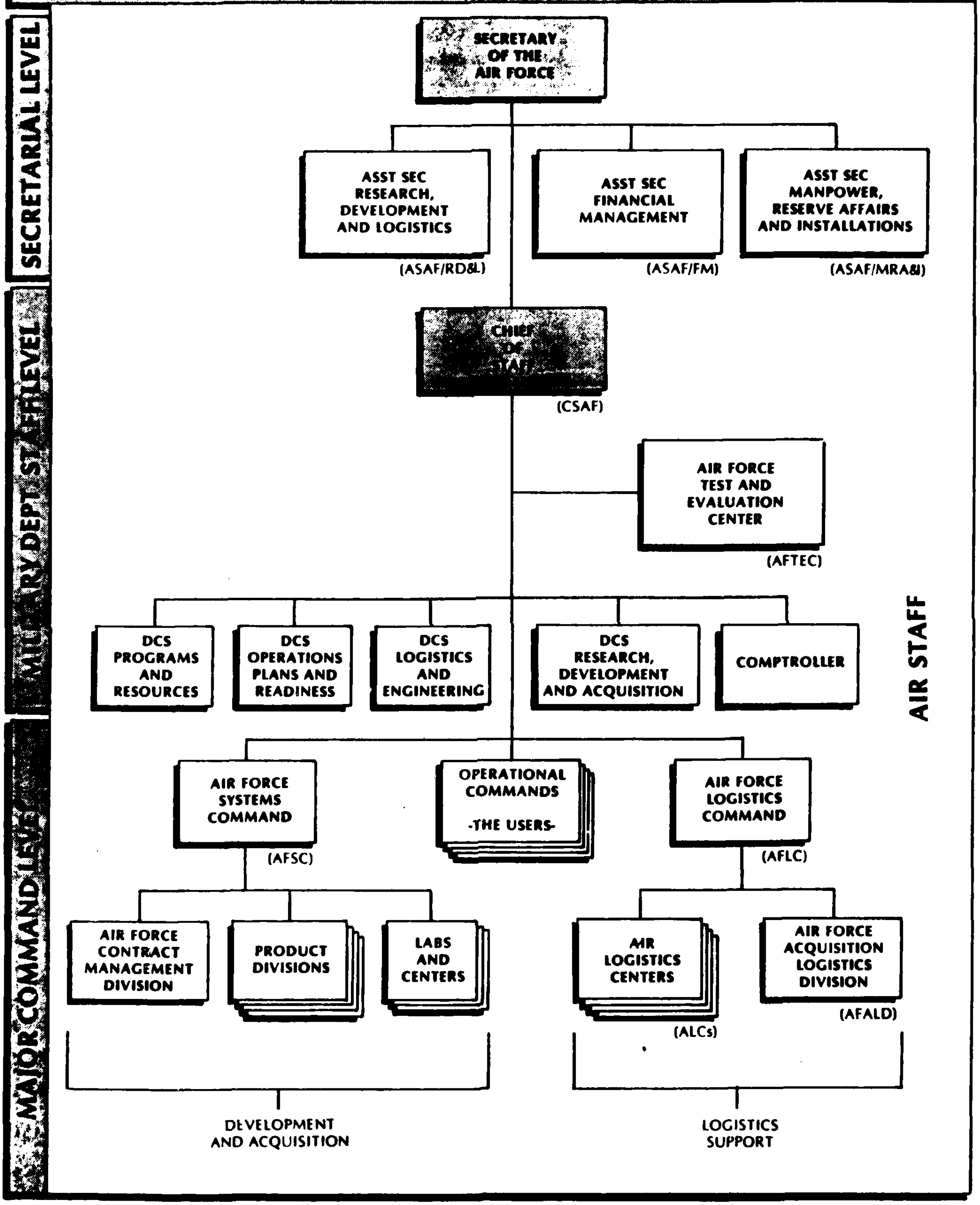
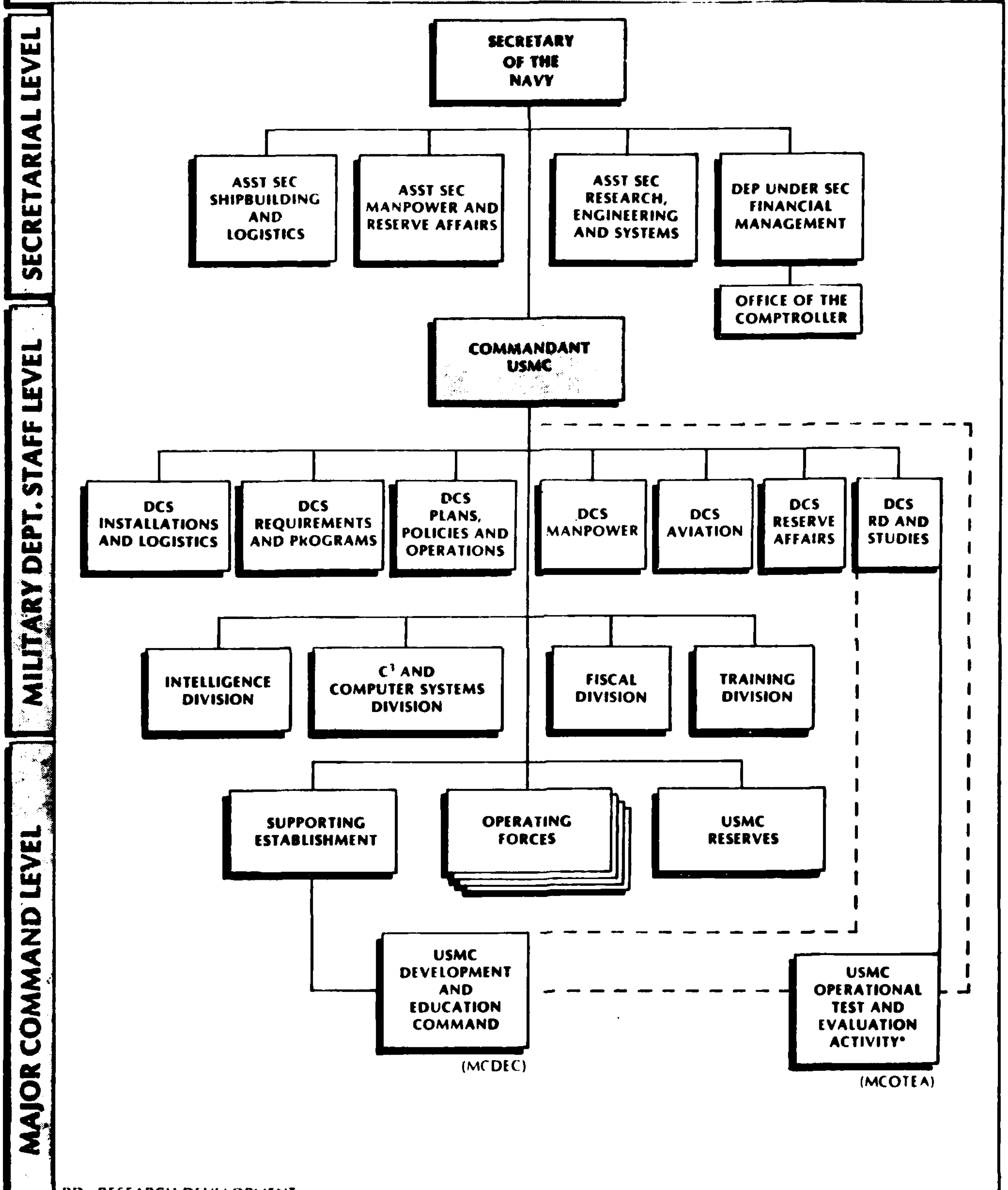


Figure 5 **SELECTED ORGANIZATIONAL ELEMENTS U.S. MARINE CORPS ACQUISITION PROCESS**



RD - RESEARCH DEVELOPMENT
C¹ - COMMAND, CONTROL, COMMUNICATIONS

*A HQ USMC UNIT LOCATED WITH MCDEC AT QUANTICO

Figure 6 U.S. MARINE CORPS ACQUISITION APPROACH

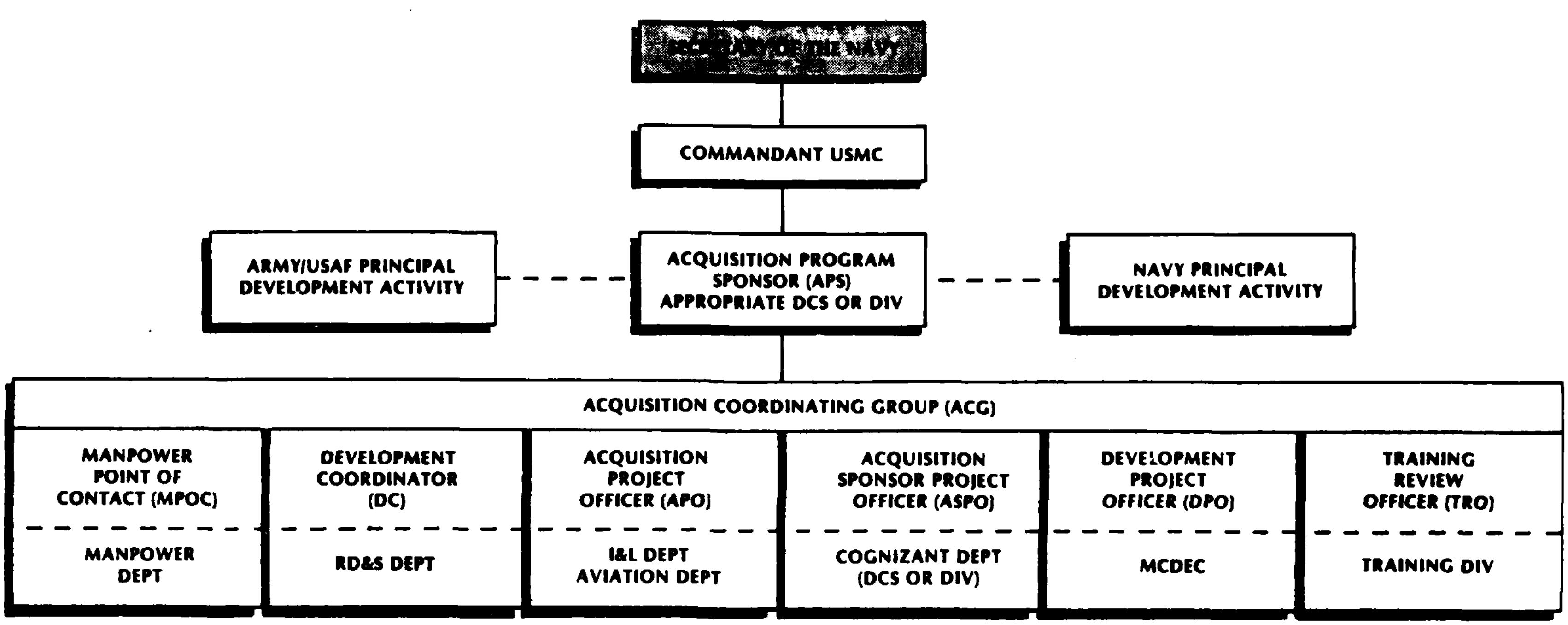


Figure 7 SUMMARY COMPARISON

	ARMY	NAVY	AIR FORCE	MARINE CORPS
SERVICE ACQUISITION EXECUTIVE	ASA (RD&A)	ASN (RE&S) - EVERYTHING BUT SHIP/SHIP CONVERSION UP TO PRODUCTION DECISION ASN (SB&L) - SHIP/SHIP CONVERSION THROUGHOUT LIFE CYCLE AND ALL POST PRODUCTION DECISION ACTIVITIES	ASAF (RD&L)	ASSISTANT CMC/CHIEF OF STAFF ASSISTED BY: DCS, RD&S (DURING RDT&E PHASE) DCS, I&L (DURING PRODUCTION AND O&S PHASES)
(S)SARC CHAIRMAN	VICE CHIEF OF STAFF	COGNIZANT ASN EITHER ASN (RE&S) OR ASN (SB&L)	ASAF (RD&L)	ASSISTANT CMC/CHIEF OF STAFF
(S)SARC MEMBERS (PER REGULATION)	UNDER SEC ARMY ASA (RD&A) ASA (IL&FM) ASA (M&RA) DUSA (OR) GEN COUNSEL CG DARCOM CG TRADOC CG OTEA DCS/RDA DCS/OPS DCS/PER DCS/LOG COMPTROLLER DIRECTOR, PA&E ACS/INTEL CHIEF OF ENGRS CHIEF, ARMY FORCE MODERNIZATION COORD OFFICE	SECNAV UNDERSECNAV ASN (RE&S) ASN (M&RA) ASN (SB&L) DUSN (FM) CNO COMMANDANT USMC CHIEF OF NAVAL MATERIAL	ASAF (FM) ASAF (MRA&I) VICE CS COMPTROLLER DCS/PROGRAMS & RESOURCES DCS/RD&A DCS/LOG & ENGR DCS/OPERATIONS PLANS & READINESS DCS/MANPOWER & PERSONNEL	DCS PLANS, POLICIES & OPERATIONS DCS REQUIREMENTS & PROGRAMS DCS MANPOWER DCS INSTALLATIONS & LOGISTICS DCS AVIATION DCS RD&S DCS RESERVE AFFAIRS DIRECTOR, C^3 AND COMPUTER SYSTEMS DIV DIRECTOR, INTEL DIV DIRECTOR, TRAINING DIV FISCAL DIRECTOR CG, MCDEC
EXECUTIVE SECRETARY	DCS/RDA	DIRECTOR, OFFICE OF PROGRAM APPRAISAL	DEP ASAF, ACQUISITION LOGISTICS POLICY	DCS, RD&S
PRINCIPAL MIL DEPT SERVICE HQ STAFF LEVEL AND MAJOR COMMAND STAFF LEVEL FOCAL POINTS	DEPT OF ARMY SYSTEM COORDINATOR (DASC) - (DCS/RDA) DEPT OF ARMY FORCE INTEGRATION STAFF OFFICER (FISO) - (DCS/OPS) TRADOC SYSTEM MANAGER (TSM)	PROGRAM COORDINATOR (PC) - OPNAV DCNOs FOR WARFARE SPECIALITIES DEVELOPMENT COORDINATOR (DC) - OPNAV DIRECTOR RDT&E IF R&D FUNDS ARE INVOLVED	PROGRAM ELEMENT MONITOR (PEM) - AIR STAFF (SOME ARE LOCATED AT HQ AFSC) SYSTEMS OFFICER (SYSTO) - AFSC	ACQUISITION SPONSOR PROJECT OFFICER (ASPO) (LOCATED AT HQMC)
PRINCIPAL OPERATIONAL REQUIREMENTS DOCUMENTS	LOA - LETTER OF AGREEMENT ROC - REQUIRED OPERATIONAL CAPABILITY LR - LETTER REQUIREMENT	OR - OPERATIONAL REQUIREMENT (LESS THAN MAJOR PROGRAMS) JMSNS - JUSTIFICATION FOR MAJOR SYSTEM NEW START	SON - STATEMENT OF NEED (PRIMARILY FROM OPERATIONAL COMMANDS OR USER)	ROC - REQUIRED OPERATIONAL CAPABILITY (MCDEC PREPARATION)

TRADOC	OPNAV	DCS/OPS, PLANS & READINESS	ACQUISITION PROGRAM SPONSOR (APS)
RECAP - REVIEW AND COMMAND ASSESSMENT OF PROGRAMS (DARCOM LEVEL) LOGCAP - LOGISTICS AND COMMAND ASSESSMENT OF PROGRAMS (DARCOM LEVEL) IPR - IN-PROCESS REVIEW (DCSRDA, DARCOM LEVEL) PIB - PERIODIC INFORMATION BRIEFING (HQ DA LEVEL) ASARC - (HQ DA LEVEL) DSARC	ARB - ACQUISITION REVIEW BOARD (NAVMAT LEVEL) LRG - LOGISTIC REVIEW GROUP (NAVMAT LEVEL) ARC - ACQUISITION REVIEW COMMITTEE (OPNAV LEVEL) CEB - CNO EXECUTIVE BOARD (OPNAV LEVEL) SAIP - SHIP ACQUISITION IMPROVEMENT PANEL (OPNAV LEVEL) DNSARC - (NAVY DEPT LEVEL) DSARC	MAR - MGT ASSESSMENT REVIEW (PRODUCT DIV LEVEL) CAR - COMMAND ASSESSMENT REVIEW (AFSC LEVEL) PAR - PROGRAM ASSESSMENT REVIEW (HQ USAF LEVEL) SPR - SECRETARIAL PROGRAM REVIEW (SAF LEVEL) AF BOARD STRUCTURE (AIR STAFF LEVEL) AFSARC - (HQ USAF LEVEL) DSARC	IPR - IN PROGRESS REVIEW COMMITTEE (HQMC LEVEL) SPECIAL MSARC - (HQMC LEVEL) MSARC (HQMC LEVEL)
DIRECTOR, PA&E COMPTROLLER (DIR, ARMY BUDGET)	DIRECTOR, PROGRAM PLANNING (OP-090) DIRECTOR, PROGRAM PLANNING (OP-090)	DCS, PROGRAMS & RESOURCES COMPTROLLER (DIR AIR FORCE BUDGET)	DCS, REQUIREMENTS AND PROGRAMS FISCAL DIRECTOR
OTEA TRADOC DARCOM DARCOM	OPTEVFOR OPNAV NAVMAT NAVMAT	AFTEC OPERATIONAL COMMANDS AFSC AFLC	MCOTEA OPERATIONAL COMMANDS MCDEC DCS, I&L