NAVSWC TR 91-795

COMBAT SYSTEMS VISION 2030
COMBAT SYSTEM ARCHITECTURE:
DESIGN PRINCIPLES AND METHODOLOGY

BY BERNARD G. DUREN JAMES R. POLLARD
COMBAT SYSTEMS DEPARTMENT

DECEMBER 1991

NAVAL SURFACE WARFARE CENTER
Dahlgren, Virginia 22448-5000 • Silver Spring, Maryland 20903-5000

NAVSWC TR 91-795

COMBAT SYSTEMS VISION 2030
COMBAT SYSTEM ARCHITECTURE: DESIGN PRINCIPLES AND METHODOLOGY

BY BERNARD G. DUREN   JAMES R. POLLARD
COMBAT SYSTEMS DEPARTMENT

DECEMBER 1991

Approved for public release; distribution is unlimited.

NAVAL SURFACE WARFARE CENTER
Dahlgren, Virginia 22448-5000 • Silver Spring, Maryland 20903-5000

FOREWORD

As the decade of the 1980's came to a close, the Naval surface warfare community found itself in a changing world. Changes in the Soviet Union and Eastern Europe were creating uncertainty in the future missions and roles of the Navy. Advances in technology, particularly in computer-related fields, were suggesting significant changes in shipboard combat systems. Major shipbuilding programs were beginning to consider next generation designs. Against this backdrop of uncertainty, the Naval Surface Warfare Center's management decided to develop a *vision* of the future in combat systems for surface combatants. Set in the 2030 timeframe, a Combat System Vision was considered to consist of a combat system architecture framework and a set of technology goals framed in future combat system concepts.

This report addresses some of the underlying work that went into development of a combat system architecture. It describes a loose collection of system engineering principles for application in combat system architecture and design efforts. It is the second of four reports on combat system architecture and system engineering topics. Other reports deal with the combat system architecture description, the technical and engineering problems associated with realization of the architecture in physical combat systems, and a set of analytical experiments that can be conducted to strengthen the scientific and technical underpinnings of combat system architecture and design work.

Approved by:

CARLTON W. DUKE, JR., Head
Combat Systems Department

ABSTRACT

Combat system architecture is defined, and a reference model is formulated for the operating tasks performed by combat systems. Factors driving the influence of architecture design on the military worth of surface combatants are discussed, and an associated set of design principles is provided. These are presented in the form of a building code or general design rules. The rule base is then used to trace key decisions in development of the 'H' or Vision Architecture concept stated in NAVSWC TR 91-607. Finally, a parallel derivation is given for the Vision Architecture that is based on methods for synthesis of decentralized control structures.

CONTENTS

Section	Page
1.0 COMBAT SYSTEM ARCHITECTURE DEVELOPMENT	1
1.1 COMBAT SYSTEM ENGINEERING AND ARCHITECTURE	1
1.2 WORKING DEFINITIONS	2
1.3 NEED FOR ARCHITECTURE	3
1.4 UTILITY OF ARCHITECTURES	4
1.5 ORGANIZATION OF REPORT	5
2.0 WARFIGHTING PROCESSES	6
2.1 DESCRIPTIVE FRAMEWORK	6
2.2 ENGAGEMENT INTERACTION PROCESSES	8
2.3 COOPERATIVE INTERACTION PROCESSES	11
3.0 ARCHITECTURE PRINCIPLES	15
3.1 FUNCTIONALITY ENHANCING	16
3.2 AFFORDABILITY ENHANCING	18
3.3 SYSTEM INTEGRITY ENHANCING	21
4.0 APPLICATION TO ARCHITECTURE DESIGN	24
4.1 WEAPON SYSTEM FUNCTIONS	24
4.2 MULTIPLE WEAPON SYSTEMS	25
4.3 SHARED INFORMATION	26
4.4 RESOURCE SHARING	27
4.5 CLUSTERED WEAPON SYSTEMS	28
4.6 WARFARE AREA DECOMPOSITION	28
4.7 UNIT LEVEL DECOMPOSITION	29
4.8 COMBAT SYSTEM SENSING RESOURCES	30
4.9 COMBAT SYSTEM ENGAGING RESOURCES	33
4.10 FORCE ARCHITECTURE	33
5.0 BACKBONE CONTROL STRUCTURE	35
5.1 NETWORK OF MODULES	35
5.2 SYSTEM MODELING	37
5.3 UNIT LEVEL DESIGN	39
5.4 WARFARE AREA DESIGN LEVEL	40
5.5 CLUSTERED WEAPON SYSTEMS LEVEL	42
5.6 LAYERED ARCHITECTURE FOR INTERCONNECTIONS	42
6.0 REFERENCES	45
7.0 BIBLIOGRAPHY	46

CONTENTS (Continued)

APPENDIXES
A – ARCHITECTURE DEFINITIONS A-1
B – SUBSYSTEM ARCHITECTURE PRINCIPLES B-1

DISTRIBUTION .... (1)

ILLUSTRATIONS

Figure — Page

1 BASIC CONTROL HIERARCHY 7
2 FUNDAMENTAL WEAPONS SYSTEM FUNCTIONS 25
3 COORDINATED WARFIGHTING 25
4 INFORMATION COORDINATION 26
5 RESOURCE COORDINATION 27
6 WARFARE AREA FORMATION 28
7 MULTIPLE WARFARE AREAS (HORIZONTAL DECOMPOSITION) 29
8 VERTICAL DECOMPOSITION 31
9 COMBAT SYSTEM FUNCTIONAL ARCHITECTURE 32
10 MATCHING COMBAT SYSTEM/FORCE ARCHITECTURES 34
11 MODULE ARCHITECTURE TEMPLATE 36
12 NETWORK OF MODULES–UNIT LEVEL 40
13 NETWORK OF MODULES–WARFARE AREA VIEW 41
14 NETWORK OF MODULES–AAW WEAPON CLUSTERS 43
15 COMBAT SYSTEM TACTICAL INFORMATION 43

TABLE

Table — Page

1 MULTIPLE VIEWPOINTS 3

1.0 COMBAT SYSTEM ARCHITECTURE DEVELOPMENT

This report builds on the earlier work of Lindemann,[1] Cullen,[2,3] Pollard,[4] and others. Shared interest in a strong conceptual framework for design of future combat systems is the tie that binds past and present efforts together.

1.1 COMBAT SYSTEM ENGINEERING AND ARCHITECTURE

Any combat system will go through a birth-to-death process referred to as its lifecycle. Six major events in a combat system's lifecycle often cited are: (1) requirements definition, (2) system design, (3) design implementation or construction, (4) system integration and test, (5) system operations and inservice support, and (6) system retirement. The system engineering process spans the entire lifecycle. However, the focus here is on the system design phase. Blanchard defines system engineering as the process of translating operational requirements into engineering functional requirements and subsequently expanding these functional requirements into detailed equipment and service end item design specifications.[5] Several levels of design, or stages in translating requirements into design specifications, are identified: feasibility design, preliminary design, detailed system and equipment design.

The design stages correspond to steps in the acquisition cycle. In the past, the feasibility design stage has corresponded to the Development Options Plan (DOP) process, which develops feasible design options in response to a Tentative Operational Requirement (TOR).

The preliminary design stage in turn corresponds to the Naval Sea Systems Command (NAVSEA) process for developing a Preliminary Design Report (PDR). The PDR is in response to an Operational Requirement (OR) that is a specific requirement based on the preliminary design (option) selected by the Office of the Chief of Naval Operations (OPNAV). It contains an "A level specification" for the system, much more detailed than that of the DOP.

Finally, the detailed system and equipment design levels correspond to the NAVSEA process for developing a Contractor Bid Package (CBP). The CBP again is in response to a more detailed set of requirements spelled out by OPNAV in a Top Level Requirements (TLR) document.

The architecture of a combat system is first developed in the feasibility design stage. Blanchard identifies four steps for developing a feasibility design; namely, (1) functional analysis, (2) requirements allocation, (3) tradeoff and optimization, and (4) system synthesis and definition.[5]

The purpose of the functional analysis step is to develop functional architecture for the combat system. Operational requirements are decomposed into operational and system functions and their relationships are identified. These are often displayed in functional flow diagrams. With the functions identified, they are next grouped and arranged to form a preliminary packaging concept or functional architecture. This is done by segmenting or decomposing the functional combat system into near independent subsystems with distinct functions and well defined boundaries or interfaces with other subsystems. Operational and engineering principles or rules are also used to guide the segmentation process. Thus at this stage in the combat system engineering process, the combat system functional architecture is developed.

The second step, that of requirements allocation, takes the operational requirements in the form of performance and effectiveness requirements, physical requirements, lifecycle costs, etc., and distributes or allocates them to the subsystems of the functional architecture. At this point, the functional architecture not only specifies the functions or tasks to be carried out by the combat system to meet the requirements, but also specifies how well the task is to be performed, the manner in which it will be performed, etc. Next, physical elements (equipments, people, or computer programs) are identified to perform the combat system's functions and allocated requirements. At this point in the combat system engineering process, the (initial) combat system physical architecture is developed. The requirements allocation process now continues, allocating the remaining requirements, physical constraints, cost, etc., to the combat system elements.

The third step in developing a feasibility design is that of tradeoff and optimization. The best design is determined through an iterative approach of performance and cost tradeoffs among the candidate combat systems elements. By changing and exchanging elements in this process, the final physical architecture and design emerges.

The fourth step is system synthesis and definition. Here, there is a combining and structuring of system elements to ensure they form a proper functional entity. This is usually accomplished by analytical means through modeling of the total system. It is performed to ensure the total system meets requirements after the lower level element tradeoffs and optimizations. The resultant performance, configuration, and arrangement of the chosen system are thus portrayed.

A feasibility design specification is the final result of the feasibility design stage. As mentioned above, this leads to a further refinement and definition of the operational requirements and on to more detailed combat system designs in the preliminary design, detailed system, and equipment design stages.

1.2 WORKING DEFINITIONS

A working definition for the term *combat system architecture* can be given as follows:

> "A combat system architecture is a logical construct for defining and controlling the physical realization of a combat system and associated processes for target engagement. It is formed by partitioning the system into subsystems and interconnections so that over the entire lifecycle of the system, applicable functional, organizational, and physical requirements of combat operations can be met."

In development of large-scale, complex systems, a set of architectural representations is usually produced, each depicting the perspective of a key participant such as the owner, the designer, and the builder. The representations used in building construction and combat system engineering are contrasted in Table 1 below. Different information must be provided in each representation, corresponding to the needs and tasks of the actor. Since this report considers the feasibility design phase of system engineering, a functional approach reflecting end use factors is most appropriate. Architecture design thus involves a high-level statement of system operating tasks, plus a top-level topology partitioning task elements into subsystems and interconnections. Separating architecture design from implementation design supports the basic system engineering principle that requirements be separated from design.

The above definition is rooted in the notion of engagement processes. In general, a process can be defined as a set of interrelated work activities, characterized by a set of specific inputs and value-added tasks or functions that produce a set of specific outputs. Process descriptions involve three elements. The first is a statement of what the process does, the basic goals of its execution, and what constraints are involved. The second is a model or representation for process inputs and outputs, together with an algorithm for process execution. The third element is a physical realization (implementation) for the process.

TABLE 1. MULTIPLE VIEWPOINTS

VIEWPOINT	BUILDINGS	COMBAT SYSTEMS
Ballpark	Bubble Charts	User Needs
Owner	Architect's Drawings	System-Level Feasibility Design
Designer	Architect's Plans	System-Level Preliminary Design
Builder	Contractor's Plans	System Detailed Design
Producer	Shop Plans	Equipment-Level Detailed Design
Operator	–	Integration, Test, and Deployment

An architecture may be considered successful if the system is so configured that, over its entire life span: (a) all subsystem interfaces are clearly defined; (b) qualified suppliers exist for all subsystems and components; (c) operators can make it work in the real world; and (d) system acquisition and support are affordable.

1.3 NEED FOR ARCHITECTURE

The need for a combat system architecture is not a theoretical issue. Combat systems are not available off the shelf. We cannot buy them ready for use; we must design and build them. Since this cannot happen in a haphazard, unplanned manner, appropriate guidelines must be used, that is: "adopt a system architecture and provide for system integration."

A system architecture constitutes a framework for implementation, and thus is determined with reference to the envisioned process, process model, and execution algorithm as well as technology. The term refers to a set of relationships, interactions, and principles for design of a unified operating entity capable of supporting a general concept of combat operations. Design begins with formulation of a reference model for the anticipated operating environment. This model is intended to capture the operating concept rather than to represent specific engagement details. It provides for definition of key concepts (such as entities, systems, and interactions) and a structure denoting relationships between the terms defined. For completeness, these relationships must represent all actions that may be expected in any given operating environment. Beginning with the most basic operational aims and tasks, a layered hierarchical description is then created for the combat system and its operations. Required functions are decomposed into a set of required interactions, and the interactions are subsequently broken down into interaction trees and process trees.

1.4 UTILITY OF ARCHITECTURES

A system architecture is necessary to ensure the lifecycle effectiveness of a combat system design. The term lifecycle effectiveness is used to mean the operational effectiveness of a combat system design and its extensibility or ability to accommodate change. Hornstein and Willoughby consider ways to adapt, enhance, or modify the traditional practice of system engineering management to accommodate development of systems with lifecycle effectiveness despite requirements that may be incomplete, unquantifiable, or ambiguous. In the following sections, ways are suggested to achieve lifecycle effectiveness in combat systems.

1.4.1 Operational Effectiveness

Often in the past, the operational effectiveness of a new combat system could not be determined until the system implementation was complete. Only then did enough operational experience become available to determine the modifications necessary. Redesign and reimplementation then achieved a second generation system with improved operational effectiveness. This improvement resulted from operational experience being input to the next cycle of requirements specification. This suggests that better ways of coupling measures of operational effectiveness into the traditional development process could yield similar advantages. The following are suggested:

- Start with general functional requirements applicable to all combat systems as a class; use them as a baseline in defining requirements.

- Establish and maintain competing alternative operational concepts.

- Add operational effectiveness criteria to the evaluation process used in requirements and design reviews.

- Review designs for interpretations of requirements that unnecessarily limit performance.

1.4.2 Extensible Systems

Combat system designs in the near future are expected to continue to be evolutionary redesigns that utilize many existing elements, weapon systems and components, with some new elements added to accommodate changes in requirements and technology. Totally new designs for combat systems usually turn out to be too costly and unwarranted in the face of the large investment in existing elements that often have adequate performance. Thus, multiple cycles of requirements definition, analysis, design, and implementation are expected to remain the norm, and extensibility continues to be a critical design factor.

The term extensibility refers to system ability to accommodate change or be stretched without breaking. Extensible systems are therefore those that have been designed to accommodate change. They have a robust architecture to minimize the impact of redesign and/or reimplementation and can be scaled without extensive change. They are built for general cases, with unique requirements handled as special cases. Extensible systems are built with tools that allow them to be data- and rule-driven.

Overall, an evolutionary acquisition strategy that provides for multiple cycles of design and implementation is recommended. Not all systems can be built as extensible systems. Certain conditions must exist, including the following:

- Availability of a descriptive vocabulary that is independent of any specific operational domain.
- Existence of a functional architecture that can be used across a broad range of combat system alternatives.
- Recognition of a common architecture for solving a very broad range of specific applications.
- Existence of data structures within that virtually all applications can be described.
- A set of functional building blocks or components from which customized applications can be developed by an assembly process.
- A set of tools that makes possible the development of systems that are data-driven and/or rule-driven.

Thus, a key element in the proposed methodology is to develop a general conceptual framework that is applicable to a broad class of possible designs for the system of interest. Results form a baseline for setting requirements in specific projects. Since the general framework is constructed at a higher level, these requirements can be more complete, less ambiguous, more measurable, and more stable. This approach also supports adoption of a robust architecture, compatible with a broad range of specific applications and suitable for development of extensible systems. Individual combat systems may not implement the full organizational and functional content of a generic architecture. Every system, however, will implement proper subsets of that architecture. This report is the result of efforts to establish the kind of general framework for combat system engineering implicit in the above principles.

1.5 ORGANIZATION OF REPORT

The body of the report is organized into four parts. Section 2.0 presents a conceptual and generic description of the warfighting processes supported by a combat system. Section 3.0 presents architecture design principles for achieving maximum military worth in the combat system. Section 4.0 illustrates the application of process knowledge and architecture principles in tracing key decisions in the assembly of the Vision Architecture concept described by Reference 4. The content of this chapter is of interest in its own right, but here serves to illustrate use of architecture principles as a knowledge base to support combat system engineering and/or design work. Section 5.0 gives a parallel derivation for the Vision Architecture based on methods for synthesis of decentralized control structures. Appendix A identifies terms and definitions that underpin the working definition for combat system architecture given in Section 1.4 above. Finally, Appendix B presents a larger set of architecture principles for use in subsystem-level work. These principles apply to the people, procedures, and physical plant that make up a combat system.

2.0 WARFIGHTING PROCESSES

A combat system can be viewed as a system for processing targets. In essence, each warfighting path constitutes a sequential process for end-to-end engagement of a target. Section 2.1 provides a conceptual and generic description of what a combat system is and does, in terms of basic warfighting processes. Within this descriptive framework, a reference model is then given in Sections 2.2 and 2.3 for the types of processes supported by combat system operations. This model provides an essential context and point of departure for combat system architecture design.

2.1 DESCRIPTIVE FRAMEWORK

2.1.1 Terms

A statement of such generic process requirements involves three basic terms: interactions, entities, and systems. A system-user interaction is defined as a sequence of events involving entities and the system. Entities are defined as people, machines, or events. The combat system is defined as the combination of human, computer, and network elements that constitute the warfighting capabilities of a ship.

For many systems (military as well as industrial), functionality can be described by a simple relation between initial state, inputs, and the outputs produced at some terminal state. These are called relational or transformational systems. Still other systems, sometimes more complex, are designed to maintain some interaction with their environment, terminating only if the system should fail. Examples include most kinds of real-time embedded computing systems, control plants, communication systems, interactive software and even very large scale integrated (VLSI) circuits. Since there is no natural terminal state, these reactive systems cannot be described by a simple relation specifying outputs as a function of inputs. Instead, they must be described in terms of their ongoing behavior. Adequate descriptions typically involve complex sequences of events, actions, conditions, and information flows, often with explicit timing constraints that combine to form the system's overall behavior.

2.1.2 Combat System Processes

Combat systems support both transformational and reactive operating processes, giving rise to the basic control hierarchy shown in Figure 1 below. Posturing forces, planning, and coordination are transformational tasks, while interactions with enemy units are reactive. Each represents a distinct type of process, calling for a different approach to system design. Movement, navigation, and training, for example, are relational process. On the other hand, all forms of weapon delivery involve reactive processes. Long-range strike makes a useful example because a centralized planning approach with decentralized execution is readily implemented. A relational model is appropriate for a typical destroyer's role in such operations, which is to generate one component of the strike in accordance with orders. But the overall strike is reactive in character, a fact that becomes evident only when total detection-to-destruction cycle time is considered. The required interaction is accurate delivery of ordnance against a set of *targets that count.* Changes in target location or background, weather conditions, and availability of target intelligence give this interaction a dynamic character. Since total cycle time must eventually be reduced to deal effectively with relocatable targets, a relational model could produce systems with limited growth potential. Maneuvering in formation and the conduct of air operations, which involve safety factors with a dynamic character, also involve reactive processes.

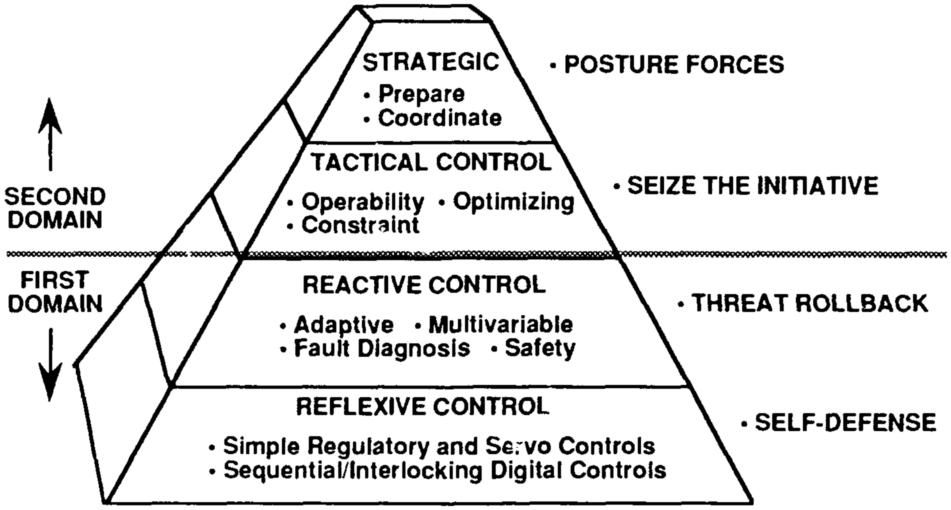

FIGURE 1. BASIC CONTROL HIERARCHY

A reactive process is said to be reflexive when human control is limited to supervisory functions, direct control having been automated. In practical terms. combat systems must be able to handle situations where missiles are in flight inside 100 mi and still make preparations for a second strike that may be as much as 1,500 mi away. This involves two domains (time scales), the first beginning perhaps 15 min prior to combat and involving real-time engagement tasks. Combat direction systems are designed chiefly for operations on this time scale.

The domain (time scale) is not yet fully defined, either conceptually or architecturally. In general, unless the unit is within 15 min of a combat engagement or actually under fire, it is operating completely in this regime. This encompasses the whole area of training, planning, coordination, and assessment.

2.1.3 Interactions

A combat system can also be viewed as a set of sequential processes together with the means to plan and control their employment. In particular, warfare operations involve the interaction of a combat system and an external entity through cooperative or engagement processes. Interaction occurs in one or more of three sequential stages–initialization, action, and termination. Basic interactions involve one entity and the system and cannot be decomposed further. Composite interactions involve system interaction with multiple entities and can be decomposed into a set of basic or composite interactions, combined sequentially, simultaneously, independently, or recursively.

2.2 ENGAGEMENT INTERACTION PROCESSES

Two forms of engagement interaction can occur. The first is reactive engagement, in which some threat is treated as a disturbance input, and requires an event-oriented or transaction processing response. The second involves a transformational process that, like decentralized execution of strike warfare, may involve a batch processing mode.

2.2.1 Preparations

The combat system supports the following:

- Training of unit personnel for engagement interactions.
- Efforts to define and bound its engagement interaction missions.
- Efforts to develop plans and doctrine for its engagement interaction missions.
- Efforts to establish and maintain material readiness of the unit for engagement interactions.

2.2.2 Initialization

2.2.2.1 Combat System.

- The system establishes and maintains a tactical picture.
- Based on available knowledge of enemy dispositions, the current tactical picture, and applicable tactical objectives, the combat system may conduct cover and deception operations.
- The combat system detects approaching threats and/or selects target entities for processing.
- The combat system identifies the target entity and assesses its tactical significance.

- The combat system allocates resources needed for the interaction. Resource allocation includes efforts to detect and track other potential targets and to assess own force/unit posture.

- The combat system assesses kinematics and electromagnetic factors governing the interaction and determines threat priority.

- The combat system generates action options and evaluates them in terms of battle doctrine, rules of engagement, feasibility, and kill probability. This includes the decision to disrupt or engage enemy platforms, weapon systems, or both.

- The combat system completes the threat evaluation and weapon assignment phase and either proceeds with the selected action option or drops the target.

2.2.2.2 External Entity (Target).

- The target entity prepares for engagement and establishing and maintaining a tactical picture.

- The target entity selects the combat system of interest for engagement and/or enters its battle space.

- The target entity generates a signature observable by the combat system.

- The target entity adopts a course of action for the engagement. This may include the use of cover and deception techniques and/or deployment of penetration aids.

- The target entity recognizes when the initialization stage ends and either proceeds with or cancels the planned course of action.

2.2.3 Actions

2.2.3.1 Combat System.

The combat system activates the resources necessary for an engagement and manages them. Resource management involves the following subtasks:

- Allow for any profile of resource usage within any activity.

- Allow for use of both pooled and individual resource types.

- Allow activities to obligate, consume, and/or generate resources.

- Allow for storage and retrieval of multiple working schedules.

- Allow editing of activities and resources, temporal and resource relationships, partial schedules, and availability profiles.

- Accommodate earliest start/latest finish windows on all activities.
- Accommodate flexible intervals for resource usage within an activity.
- Accommodate variable duration activities.
- Accommodate interruption and restart of activities.
- Accommodate any priority scheme among activities.
- Enforce all temporal and resource relationships.
- Facilitate comparison of schedules by user personnel.

In addition to these resource management functions, the combat system carries out the following actions:

- The combat system establishes and maintains a tactical picture to support combat operations.
- The combat system reallocates resources if the threat situation or its own readiness changes significantly during the process of engagement, or if system resources can be used more efficiently. Possibly, the system switches to an alternate course of action if the situation becomes unclear or essential resources are unavailable.
- The system tracks and projects the course of the engagement and assesses the likelihood of a successful outcome.
- The system makes provisions for essential feedback on the progress of the engagement.
- The system monitors the engagement and records data needed for subsequent actions, for extraction of lessons learned, and for design of improved tactics and systems.
- The combat system conducts a battle damage assessment or kill assessment (as appropriate), determines the need for reengagement, and either signals termination or initiates followup action.

2.2.3.2 External Entity (Target).

- The target entity continues the action.
- The target entity employs sensors to establish and maintain a tactical picture suitable for use in the engagement.
- The target entity may modify the conditions of engagement by coordinating its actions with those of other external entities, by releasing weapons, or by using penetration aids.

- The target entity can change its current course of action on its own initiative or in response to observable combat system actions.

2.2.4 Termination

2.2.4.1 Combat System.

- The system restores engagement resources to a ready posture, updates status displays and databases, and takes needed action to maintain logistic readiness of its component systems.
- The system determines and records engagement data.
- The system determines and records engagement results.
- The system extracts lessons learned from the engagement. Alternatives for better performance (more economical, quicker, or more effective response) are identified.
- The system communicates ways of using its capabilities better to the user. This can include options that enable the user to gain improved results in future engagements. In addition, the system can receive feedback from force sensors about engagement performance.
- The system remembers deferred communications or actions.

2.2.4.2 External Entity (Target).

- The target entity breaks off the action, restoring its equipment to an idle state.
- The target entity collects information about actions taken by the combat system.

2.3 COOPERATIVE INTERACTION PROCESSES

In this case, the external entities of interest are friendly units. The operating task may be associated with maneuvering in formation, underway replenishment, cooperative engagement, or communications.

2.3.1 Preparations

The combat system supports the following:

- Training of unit personnel for cooperative interactions.
- Efforts to define and bound its cooperative interaction mission.
- Efforts to develop plans and doctrine for its cooperative interaction mission.

- Efforts to establish and maintain material readiness of the unit for cooperative interactions.

2.3.2 Initialization

This stage begins when either an entity initiates an interaction with the system or the system initiates an interaction with an entity. It ends when the entity and the system complete plans for conduct of the interaction.

2.3.2.1 Combat System.

- The combat system requests an interaction or responds to a request for an interaction with an external entity.
- The combat system identifies itself to the entity of interest. The combat system then proceeds to determine the identity of the external entity and to what extent interaction is permitted.
- The combat system must evaluate the proposed interaction in the context of its assigned missions and roles, current loading, and battle doctrine. In addition, the system must evaluate its own capability to perform the tasks required for the proposed interaction and may generate alternatives to permit more effective or more efficient use of available resources.
- Possibly, the combat system negotiates service with the external entity if resources are scarce or in contention. Negotiation involves presenting the entity with alternatives, coordinating plans, and determining that the entity has resources necessary to complete the interaction.
- The combat system allocates resources needed for interaction. Resource allocation includes attempts to locate other entities for dialog or assistance.
- The combat system recognizes the end of initialization and either proceeds with the cooperative action or cancels the request.

2.3.2.2 External Entity.

- The external entity requests an interaction or responds to a request for cooperative action.
- The external entity identifies itself to the combat system.
- The external entity negotiates for cooperative action. In this phase, the entity selects a set of cooperative tasks to be performed by the combat system. This set of tasks may have to be modified if the cooperative action requested is unclear or unavailable, or if the combat system offers alternatives.

- The external entity recognizes when the initialization stage ends and either proceeds with the operation or cancels the request if cooperative action is denied.

2.3.3 Actions

The action stage begins when the system executes the course of action formulated during initialization. However, the intended course of action may be modified even as it is carried out.

2.3.3.1 Combat System. The combat system activates necessary resources for the cooperative operations and manages them. Resource management involves the following subtasks:

- Allow for any profile of resource usage within any activity.
- Allow for use of both pooled and individual resource types.
- Allow activities to obligate, consume, and/or generate resources.
- Allow for storage and retrieval of multiple working schedules.
- Allow editing of activities and resources, temporal and resource relationships, partial schedules, and availability profiles.
- Accommodate earliest start/latest finish windows on all activities.
- Accommodate flexible intervals for resource usage within an activity.
- Accommodate variable duration activities.
- Accommodate interruption and restart of activities.
- Accommodate any priority scheme among activities.
- Enforce all temporal and resource relationships.
- Facilitate comparison of schedules by user personnel.

In addition to these resource management functions, the combat system carries out the following actions:

- The combat system reallocates resources if requested cooperative tasks change during the interaction, or if system resources can be used more efficiently. Possibly, the system renegotiates service if changing conditions make needs unclear or resources unavailable.
- The combat system tracks its use of resources for cooperation actions and computes the impact of resource tie-up or consumption on own-ship readiness posture.
- The combat system signals the external entity of changes in its operating posture.

- The combat system monitors progress of the cooperative action, but may not pass this information to the external entity immediately. Monitoring functions include informing the entity of more effective or more efficient methods, and evaluating the interaction for lessons learned.

- The combat system determines completion of cooperative operation and signals termination or responds to such a signal.

2.3.3.2 External Entity.

- The entity makes use of the cooperative actions taken by the combat system.

- The entity may inquire about various aspects of the cooperative action, such as timing and coordination requirements.

- The entity may negotiate a modified plan of action. For example, another party may be added to the operation.

- The entity can respond to signals from the combat system, or signal the combat system on its own. Either the combat system or the external entity may signal for termination.

2.3.4 Termination

The termination phase begins when the system detects that the interaction should be deferred or ended and accordingly releases all resources associated with the interaction. Especially in the case of an interaction error, it may also involve action to return the system and/or external entity to a defined state.

2.3.4.1 Combat System.

- The system restores resources and updates readiness data.

- The system determines and records interaction data.

- The system determines and records interaction results.

- The system extracts lessons learned from the interaction. Alternatives for better performance (more economical, quicker, or more effective response) are identified.

- The system communicates ways of using its capabilities better to the external entity. This can include options that enable the entity to gain improved support. In addition, the system can receive feedback from the external entity about the quality of support provided.

- The combat system remembers deferred communications or actions.

2.3.4.2 External Entity.

- The entity restores equipment to an idle state.
- The entity requests information about the support provided.

3.0 ARCHITECTURE PRINCIPLES

This chapter presents a set of principles to guide architecture design. The postulated goal of combat system development is to maximize the military worth of the surface combatant produced. Factors important for achieving this objective include:

- Mobility: Seaworthiness, endurance, speed.
- Sustained Readiness: Capacity to perform when needed and sustain that capacity over extended periods of time.
- Correct and Rapid Action: Probability of correct evaluation and rapid response in ambiguous and high risk situations.
- Area Control: Ability to fight effectively against all threats within the battle space.
- Force Projection: Ability to put ordnance on target effectively over great distances.
- Firepower: Expected target handling capacity per unit time, for standard conditions and a reference target set.
- Coverage: Ability to engage standard target types within the battle space, regardless of geographic location.
- Environmental Hardness: Resistance to both manmade and environmental sources of interference with combat operations.
- Survivability: Ability to avoid hits and fight hurt.
- Affordability: Overall warfighting capabilities in balance with costs of ownership.

The military worth of a combat system also depends on the scope of operations supported. For antiair warfare (AAW), antisubmarine warfare (ASW), antisurface warfare (ASuW), and strike warfare (STW) the intended scope can be described with reference to threat rollback operations. The term rollback signifies that operations must be conducted in the face of a significant threat. In regional conflict situations, this is most likely to apply at the onset of warfare or in a subsequent buildup phase. A residual threat, from isolated units or fragmented forces remaining to the enemy, would likely persist after completion of rollback operations.

For threat rollback in these warfare areas, three levels of capability can be identified. The most capable units represent a core element of battleforce defenses and represents the first level. Operating in sufficient strength, they could make a major contribution to threat rollback. The term *independent operations* is used to signify that the subject capability can be produced without direct assistance from more capable units.

Units with capabilities of lesser scope can still contribute to threat rollback, usually by operating together with more capable units in a battle group or task group. This represents a second level of capability. If several of these units were formed into a task group, the group still might lack some capabilities needed for independent operations.

A third level of capability may suffice to counter a residual level of threat, where hostile forces are no longer able to attack in strength. A ship with only rolling airframe missile/close-in weapons system (RAM/CIWS) for antiair self-defense might operate effectively in a DESERT STORM environment, for example, but could not be expected to deal with a heavy air threat.

Military worth of the combat system also depends on projected threat capabilities or the task difficulty levels they impose. A ship can have a lot of capability and still be unable (unless assisted) to accomplish a particularly difficult operating task or handle a specific advanced threat. For example, crossing air targets generally demand an extra measure of performance from the combat system as compared to radially inbound targets. The same is true for targets conducting radical maneuvers during the terminal phase of flight. Thus, it is important to recognize that two ships can provide the same (functional) capabilities at very different levels of performance quality. Maximum capability in every area is neither necessary nor affordable.

Within the context established by these factors, maximizing the military worth of a combat system depends on three fundamental sets of properties: functionality, affordability, and system integrity. Architecture design principles that enhance these properties are considered in this chapter.

3.1 FUNCTIONALITY ENHANCING

The following principles influence the functionality properties of a combat system:

3.1.1 Usability

- Any weapon system can be broken down into sensing, controlling, and engaging components as follows:

  1. At least one sensor associated with the system, if for no other reason than to provide targeting information.

  2. A command and control element with supporting communications to link sensor information to system ordnance and to control the weapon's actual operation.

  3. System ordnance (rounds and launcher).

This principle is of fundamental importance because warfighting paths are the primitive entities around which combat systems are formed; and the sense, control, and engage functions are the elementary constituents of all warfighting paths. This applies at all levels of military organization; a rifle squad, for example, is as much a weapon system as a surface to air missile (SAM) system.

- Individual sense, control, and engage assets should be complete functional modules, able to operate independent of other elements, yet ready for interconnection with other weapon systems in any useful arrangement.
- Each warfare area should have a separate control system so that simultaneous or independent actions can be conducted in the individual warfare areas.
- Interactions between air, surface, and underwater operations are expected to require minimal coordination between warfare areas.
- Short action paths should be provided to permit the quick responses needed in AAW and sometimes in ASuW and ASW. The total number of steps and/or stages needed to complete an engagement cycle should be held to a minimum.
- The combat system should provide for an arbitrary number of system and user processes, and ideally, should provide for flexible response to changes in battle organization as well.

For example, the design should allow multiwarfare use of sensors and weapons when such use would not interfere with primary service needs.

3.1.2 User-System Correspondence

- The combat system control structure should correspond to the battle organization, supporting and enhancing operational effectiveness.
- Since the battle organization is capable of functioning in both centralized and decentralized modes, the control structure should support operation in both coordinated and/or autonomous modes.
- The combat system control structure should provide for clear lines of authority, with a minimum number of decision points, and permit decisions to be made at the lowest appropriate level.
- Both the force and unit organizational structures are based on the principle of delegation and distribution of warfighting authority by warfare areas. Accordingly, the total surface ship combat system should allow for a decentralized control structure, with a unit command authority supported by delegated warfare mission area coordinators.
- There ought to be at least one person for each component of the combat system who is responsible for its operation, who uses it, and who needs it.

3.1.3 Automation

- The allocation of functions between men and machines should be appropriate, balanced, and consistent across the warfare areas.

- Among machines, control responsibilities should be hierarchically distributed according to the principle of increasing precision (or decreasing scope) with decreasing intelligence.

Many functions can be automated by exploiting the continued progress of technology. Increased computer capabilities make it possible to solve problems using algorithms that were not practical a few years ago. Better use of personnel may be realizable, if man/machine function allocations are reassessed across traditional subsystem boundaries, a tack that has not been taken for some time. The skill levels needed for increasingly sophisticated and complex systems are of particular concern.

Threat technical growth is increasing the need for reflexive operating modes, making automation more and more important for operational effectiveness.

3.2 AFFORDABILITY ENHANCING

3.2.1 Evergreen Design

Combat systems should be designed to permit growth and change over a long service life, including ability to replace or redesign subsystems with minimum impact on the total system. The affordability of a combat system is dependent on the ease with which this can be accomplished.

Design for change requires anticipation of the ways in which the system might be required to change; including additions, deletions, and modifications. Past approaches to design for change, applied at the element level in many cases, emphasized provision of reserves (e.g., consoles and computer equipment–peripherals, memory, and input/output (I/O) channels) and growth margins (e.g., space, weight, power, and cooling). Once these reserves are used, however, change can involve drastic system revisions. Change is further complicated by the numerous different types of elements and independently developed components for incorporation into combat systems.

However, the system should be structured to facilitate change at all levels. This is implemented by classification of the types of elements that constitute a combat system and the physical and functional relationships between them. A basic set of elements can be defined that is smaller and simpler than those found in systems today. Functions can be allocated in the warfighting, detect-control-engage paths to permit additional sensors and/or weapons to be added, and the associated new control functions, with minimum impact on existing paths. In the tactical information and command paths, functions are allocated by warfare area that establishes distribution protocols for adding new warfare areas or modifying existing areas.

What follows are sets of desired combat system properties that support an "Evergreen Design" that can accommodate change and promotes affordability. They are grouped into categories of Modular Design, Open Architecture, and Self-Revealing Designs.

3.2.1.1 Modular Design. The importance of modularity lies in the fact that without it, a small change in one place can require many compensatory changes elsewhere; changes ripple through the system design. These interactions also make it difficult to devise a practical division of labor for system development and use, since even component-level design and operations will demand a grasp of the entire system.

Required functionality of the overall system should be allocated to an array of coordinated but weakly interacting subsystem-level and element-level modules, with predefined interfaces (man/machine or machine/machine) to reduce system changes necessary when new functions or resources are added, changed, or deleted.

The following are principles for enhancing modularity characteristics in combat system architecture design:

- Domain Clarity: Form subsystems (or elements) on those functions whose individual and aggregate performance is closely coupled. Subsystem (or element) domains should be clearly defined.
- Domain Distinctiveness: The domain of each subsystem (or element) should be distinctive (with respect to other subsystem or element domains).
- Domain Boundaries: Establish a subsystem boundary between two sets of functions when the interface between those sets is stable. Subsystems (or elements) should have clear boundaries.
- Domain Stability: Subsystem (or element) domains should be relatively stable over the expected service life of the combat system (or subsystem).
- Minimal Crossover: Each subsystem (or element) should avoid interference and/or dependence on operations of the other subsystems (or elements).

3.2.1.2 Open Architecture. A system with an open architecture is one where it is easy to introduce new interfaces to the system and new modes of interaction. This implies a number of desirable properties for a combat system dealing with connectivity and simplicity of the design. These properties are as follows:

- The system architecture should be modular, with elements designed to operate in a loosely coupled manner; in particular, dependence on interelement data transfer should be minimized.
- The system should provide reliable, capable, and secure means for essential internal and external communications connectivity.
- The data transfer mechanisms linking system elements should be standardized to reduce the need for specialized integration measures.

For example, communication may be achieved by message passing on a shared databus (excluding shared memory). Standard computers, consoles, interfaces, computer programming languages, and operating systems should be employed wherever it is advantageous to do so.

3.2.1.3 Self-Revealing Design. The following are desirable properties of a combat system that deal with simplicity and clarity of design:

- Battle operations should be kept as simple and direct as possible.
- Reduce design complexity by factoring the overall problem into layers; the design of each layer can then be carried out somewhat independently of the design of the other layers.

Combat systems are large and complex, and the highest standards of engineering must be achieved if they are not to be complex and awkward to operate as well. A comprehensive system engineering process is essential to reduce design complexity to manageable levels.

3.2.2 Economy

The issues of poor quality, high cost, and long development lead time of new weapon systems have received considerable attention from senior management within the Department of Defense (DoD). The Defense Science Board, in a 1983 summer study, found that the most important and problematical factor was lack of a thorough design process–giving proper consideration to related processes such as manufacture and support as well as the product system. It is helpful to observe that the entity that develops a complex system (i.e., the development project organization) is a complex system. In particular, project management involves an information-intensive system with a control mission–a system that can be partially described and analyzed as an information system, though it contains other important elements as well. Acquisition management thus involves a dual problem relating to system design. As indicated by the emphasis on manufacturing and support factors, the Defense Science Board task force was concerned with this dual problem as much as the primary system design problem. Indeed, the more successful programs appear to draw disproportionate shares of critical attention. The root concerns are product quality and industrial efficiency, brought into question by U.S. industrial rather than military competitiveness. What is at stake, therefore, is no less than the capacity of the United States to build sustainable warfighting advantages from its basic industrial strength.

Early design decisions, such as system partitioning, can and often do have a disproportionately large impact on eventual success of large scale systems. Rogan and Cralley report that the Boeing Aerospace Co., in a study of ballistic missile system acquisition, found that while only 1 percent of system lifecycle cost (LCC) was expended in concept development, design decisions made in this phase implicitly determined 70 percent of LCC.[6] (Another 25 percent of the LCC was determined in the full-scale development phase.) Followup studies indicate that for many systems, more than half of all design flaws and errors discovered originate in the requirements stage and escape early detection. Since nonrecoverable costs accrue as they propagate into system design, implementation, and test stages, these errors lead to poor performance and skyrocketing cost. Changes to equipment and facilities are generally costly, so design changes grow steadily more expensive as development progresses. Once production begins, changes in design can mean factory retooling at tremendous expense. For computer programs, studies by three major suppliers show that the cost to detect and repair an error is 5 to 10 times greater in the coding stage, and 200 times greater in the maintenance stage than in the requirements stage.

Thus acquisition cost can be reduced by early discovery and correction of design errors. Given the large sums spent by DoD for embedded computer programs and equipment, the potential is enormous for improving performance and affordability through better management of system requirements definition and design procedures. Achieving economies of scale, eliminating errors, and preventing duplication of effort can make the overall system more affordable in terms of people, plant, and procedures.

3.2.2.1 Producibility. The following principle enhances producibility characteristics of a combat system architecture design:

> Individually and collectively, modules should be designed for producibility and interoperability. The aim should be to permit construction of subsystems (elements) by a parallel assembly process from lower level modules. This approach avoids the high cost of serial construction practices and associated bottlenecks.

3.2.2.2 Supportability. The following principles enhance supportability characteristics of a combat system architecture design:

- The subsystem (element) task loading factors should be at least nominally in balance.
- Each subsystem (element) module should be packaged to avoid any need for specialized operator skills. This principle pays dividends in terms of reduced training, maintenance, and life support costs.
- The system should include provisions that contribute to its readiness and reliability, through embedded self-test, monitoring, online training, and logistics support.

3.3 SYSTEM INTEGRITY ENHANCING

3.3.1 Distributed Systems

The principles of modularity and open system design are combined in the concept of distributed systems. Although the term is most often applied to computing systems, it will be used here in the broader sense of a system that contains embedded processing elements such as those found in sensors and weapons. The following are viewed as general rules for design of architectures with the characteristics of distributed systems:

- Provide for alternative warfighting paths to assure high levels of reliability, flexibility, survivability, and extensibility in the combat system.

- Systemwide control should be performed, so as to support seamless integration of modules into a uniformly operating combat system (subsystem).

- Control functions should be distributed to individual modules to make them ready for interconnection with other modules in any useful arrangement.

- Provide for combat system information and control flow paths such that the connections between each controller and his correspondents form direct and nonredundant paths without internal cycles or loops.

- The databases used to support the different combat system control functions should be formed into a single, comprehensive database architecture.

Though integrated, this common database does not have to be monolithic. Distributed database techniques can be used to break it down into more manageable pieces.

The trend in combat systems today is for the adoption of distributed computing techniques. Thus, some kind of structure (architecture) becomes imperative, for decentralization without structure is chaos. The magnitude and pace of change requires that a baseline be established for analyzing its impact on the structure of combat systems. This baseline also gives a needed framework for implementation of future combat systems. Structuring interactions between subsystems and components makes it possible to decompose and coordinate the associated information flows. For example, use of architectures can make it possible for multiperson teams to implement complex systems without losing control over their integration. Thus, integrated systems will remain possible despite complexity levels that make it impossible for any one person to know and remember all the design details. Establishing a reference architecture, with a descriptive vocabulary that is portable across *viewpoints* will also support adaptations to achieve system modularity and extensibility.

3.3.2 Continuity and Consistency

The following are viewed as general design rules to ensure continuity and consistency of information and data across the combat system and thus are desired combat system properties:

- Each unit of the battle organization should have ready access to all information essential to performance of its allocated functions.

- Important control decisions should be made at the point in the system architecture where all the relevant information has been brought together–a form of distributed decision making.

- Provide for measures to ensure use of a consistent information set across the entire span of action (for each controller).

This principle extends to both tactical picture data and procedural information (such as decision rules) used across battle force elements. The information structure should consistently support and be consistent with the battle organization and system structure.

3.3.3 Embedded Decision Support

The following are viewed as general design rules for enhancing the potential for embedded decision support in architecture design:

- The combat system control structure, consisting of the battle organization, consoles and workstations, computers, and interfaces, should support fully integrated operation of the combat systems, exploiting strengths and overcoming weaknesses of each component to achieve effective performance in high stress environments.

  The control structure is the glue that binds plant physical and information resources into an effective warfighting machine.

- Since integration at one level can interfere with integration at another, the degree to which higher and lower echelon subsystems will interact for coordination and control purposes should be carefully considered.

- Computer agents should be designed for service in the unique high-stress environment of combat systems and tailored for the associated decision-making organization and process, which is attuned to human abilities in combat conditions.

- Overall control should be retained by the unit commander. Each position in the battle organization, however, should report upward to only one supervisor at any point in time.

- Accountability constraints should be considered in the partitioning process; so long as humans but not computers are held accountable for military actions, human control of our military systems must be maintained.

- Allow for delegation of command and control responsibilities in accord with the principle that the scope of decision making should be matched to the scope and competence of the decision maker.

3.3.4 Reliable Operation

The following are viewed as general design rules for reliable combat system operations and thus enhance architecture design characteristics:

- Emphasis should be given to design features that result in ability of the crew to achieve sustained operation of the plant at adequate performance levels in realistic environments.

- The necessity for reliable operation in a harsh physical environment, with imperfect logistics support and a fallible crew, should be reflected in the design.

- The system should include embedded support functions that contribute to its survival and readiness through embedded self-test, monitoring, online training, and logistics support.

3.3.5 Survivability

The combat system should include provisions that contribute to its survival by the use of embedded systems for physical protection, damage control, and recovery or reconstitution. The following are two categories of general design rules for survivable combat systems and thus enhance architecture design characteristics:

3.3.5.1 Avoid Damage.

- Provide for comprehensive management of own-ship's signature to achieve and sustain capability for first delivery of firepower in effective batches.
- Provide for active self-defense operations against threat weapons (at sea and in port); including evasion and active countermeasures as well as antiweapon warfare.
- Provide for passive self-defense to avoid hits from threat weapons when possible and to minimize damage from primary weapon effects otherwise.

3.3.5.2 Recover and Fight Hurt.

- Survey and assess damage effects, plan and deploy assets for damage control and system reconfiguration within time T.
- Control damage effects and reconfigure systems to continue operations without loss of a primary warfare mission area.

4.0 APPLICATION TO ARCHITECTURE DESIGN

This chapter traces the derivation of the functional architecture given by Pollard,[4] identifying the decision made at each step and citing the major architecture principles from which it is derived. The content of this chapter is of interest in its own right, but here serves here to illustrate how this report may be used as a knowledge base to support combat system engineering and/or design work.

The point of origin for the trace is the concept that warfighting paths, found in weapon systems, are the basic working units of a combat system. This reflects the view that the function of a combat system is to process targets by one or more warfighting paths. Each path can produce a complete engagement process by a string of discrete actions or functions designed and sequenced to achieve essential combat tasks.

4.1 WEAPON SYSTEM FUNCTIONS

DECISION I: Warfighting paths for primary weapon systems are decomposed into functional modules–sense (S), control (C), and engage (E)–as shown in Figure 2.

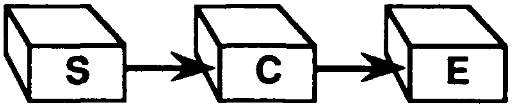

FIGURE 2. FUNDAMENTAL WEAPON SYSTEM FUNCTIONS

This decision was based on the following three architecture principles:

- Every weapon system contains three elements: sensors, control, and weapons (to include rounds plus launcher).
- Subsystems are formed on those functions whose individual and aggregate performance is closely coupled. Subsystem domains should be clearly defined.
- Short action paths should be provided to permit the quick responses needed in AAW and sometimes in ASuW and ASW. The total number of steps and/or stages needed to complete an engagement cycle should be held to a minimum.

Interconnected sense, control, and engage elements are necessary and ideally sufficient to constitute a warfighting path.

4.2 MULTIPLE WEAPON SYSTEMS

DECISION II: Coordination is necessary to achieve best performance from multiple weapon systems (see Figure 3). In Figure 3 and subsequent figures, Cd represents the Warfare Mission Area Coordinator.

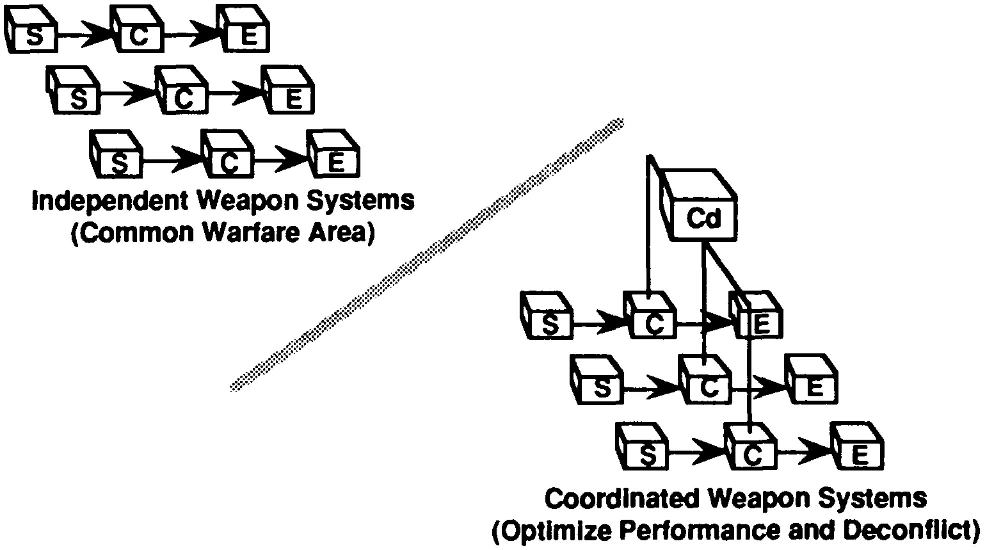

FIGURE 3. COORDINATED WARFIGHTING

This decision was based on the following two architecture principles:

- The control structure should support seamless integration of subsystems into a uniformly operating combat system.
- The combat system should be structured to support both coordinated and/or autonomous operating modes (flexibility).

Simultaneous action by multiple weapon systems against the same target can give rise to undesirable interference effects and wasteful use of resources. The concept of layered defense-in-depth indicates the benefits achievable from even simple forms of coordination.

4.3 SHARED INFORMATION

DECISION III: Information will be shared between sense, control, and engage modules of different weapon systems to improve performance and create new warfighting paths (see Figure 4).

This decision was based on the following two architecture principles:

- Provide for multiple data paths through the system to achieve greater survivability, flexibility, and growth potential than single path designs.
- Subsystems should have minimal crossover effects (interference and/or dependence) on operations of the other subsystems (e.g., electromagnetic interference between SLQ-32 and CIWS).

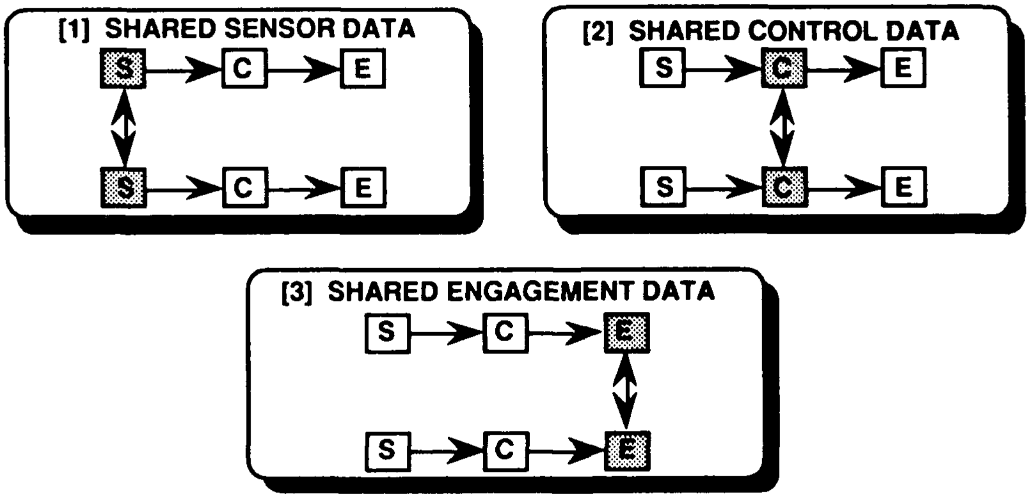

FIGURE 4. INFORMATION COORDINATION

This decision recognizes that coordination opportunities are not limited to the weapon system level. New data paths aid coordination at the weapon system level and create new warfighting paths for use as casualty modes. Sharing of sensor data also creates opportunities for improved performance through data fusion. These opportunities, no doubt, were first exploited by human controllers receiving reports from multiple sensors. In particular, the identification function (IFF) tends to draw together information from diverse sources. The decision to exploit this potential drives creation of an information coordination function.

4.4 RESOURCE SHARING

DECISION IV: Shared use of sense, control, and engage modules of different weapon systems will be supported by adding control paths between them to create new warfighting paths (see Figure 5).

This decision was based on the following three architecture principles:

- Provide for multiple control paths through the combat system to achieve greater survivability, flexibility, and growth potential than single path designs.
- Subsystems should have minimal crossover effects (interference or dependence) on operations of the other subsystems.
- Individual sense, control, and engage assets should be complete functional modules, able to operate independent of other elements, yet ready for interconnection with other weapon systems in any useful arrangement.

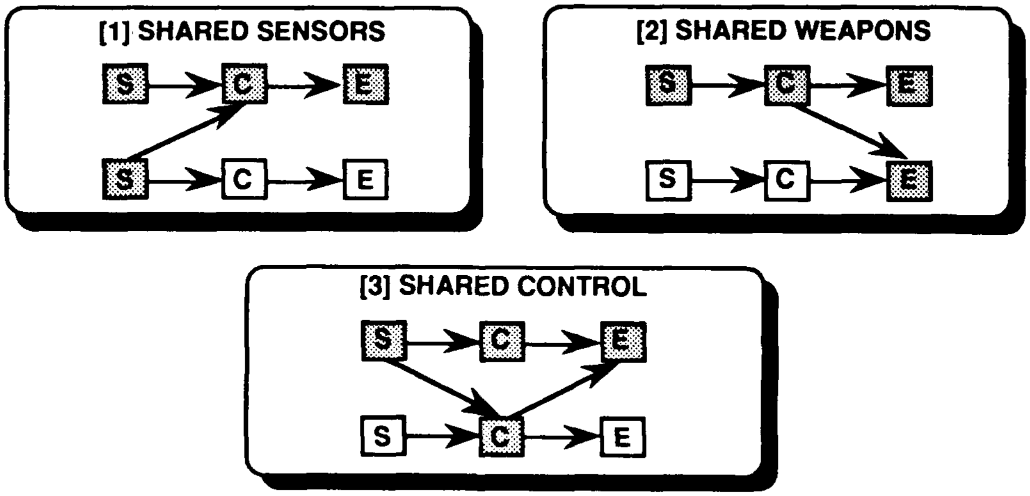

FIGURE 5. RESOURCE COORDINATION

Weapon systems are designed for autonomous operation, but failure of a critical subsystem can make the entire system inoperable. This need not occur if substitute equipment with acceptable functionality exists in other weapon systems. As shown by Figure 5, a resource coordination function is created to exploit such opportunities, whether for primary or secondary (casualty) modes of operation. This decision also motivates interest in overlapping hierarchial control systems.

4.5 CLUSTERED WEAPON SYSTEMS

DECISION V: Information and readiness coordinators are subordinated to the warfighting coordination level to form a fully coordinated weapon system cluster.

This decision was based on the following architecture principle:

- The total surface ship combat system should allow for . . . warfare mission areas capable of independent action.

At this point the interconnected sense, control, and engage modules of primary weapon systems plus coordinating modules are assembled into a cluster. As Figure 6 indicates, this forms the simplest possible warfare area module. However, such clusters could occur below the warfare area level. In the AAW area, for example, electronic warfare subsystems and missile subsystems of existing combat systems operate largely as coordinated but distinct clusters.

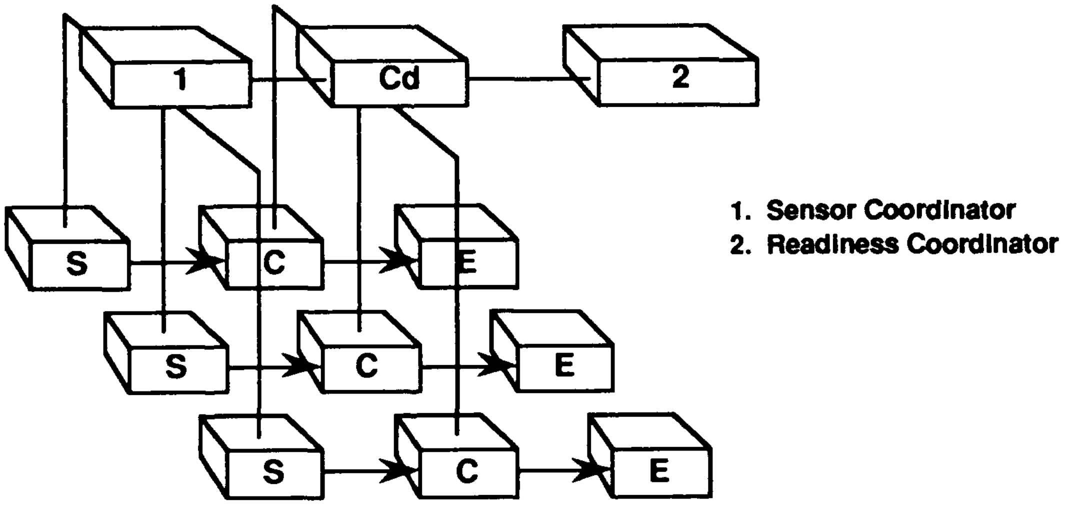

FIGURE 6. WARFARE AREA FORMATION

4.6 WARFARE AREA DECOMPOSITION

DECISION VI: Here the decision is made to divide the battle space into air, surface, undersea, and land domains, with a different cluster of fully coordinated weapon systems in each area. Figure 7 reflects the division into separate warfare mission areas.

This decision was based on the following architecture principle:

- Each warfare area should have a separate control system to permit simultaneous multiwarfare operations.

This is where the idea of decentralized command, in accordance with the composite warfare commander concept, comes into play. The warfare area coordinators conduct threat evaluation, weapon assignment, and engagement control functions. Interactions between air, surface, and underwater operations are expected to require minimal coordination between warfare areas.

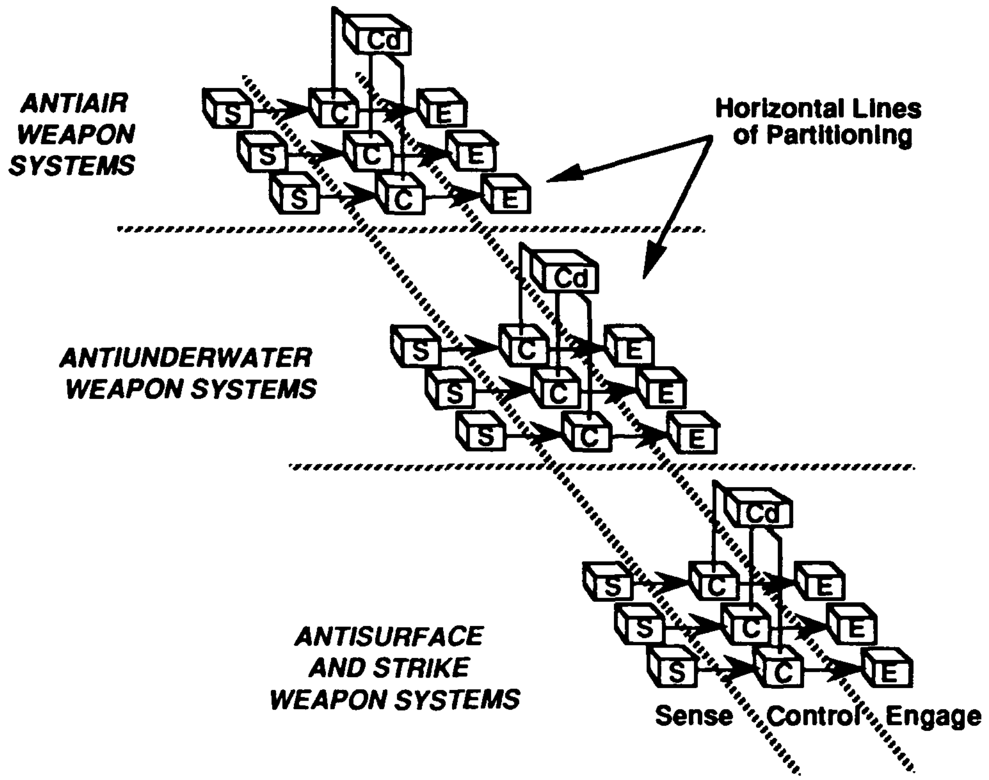

FIGURE 7. MULTIPLE WARFARE AREAS (HORIZONTAL DECOMPOSITION)

4.7 UNIT LEVEL DECOMPOSITION

DECISION VII: Unit level coordination functions (for warfighting, information, and resources) for the Warfare Area clusters are adopted (see Figure 8).

This decision was based on the following three architecture principles:

- The control structure should support seamless integration of subsystems into a uniformly operating combat system.

Once multiple warfare areas have been established, coordination across the warfare areas becomes a necessity.

- Every weapon system contains three elements: sensors, control, and weapons (to include rounds plus launcher).

  This principle applies as well to the unit level as to warfare area and weapon system levels. This principle is invoked to support creation of unit level information and resource coordination positions as well. The use of a nested and recursive organizational approach reflects basic concepts of military organization.

- The combat system should be structured to support both coordinated and/or autonomous operating modes (flexibility).

  The unit commander determines the degree of centralization to be used in control of warfare area operations. Any conflicts that arise about the allocation of shared resources (e.g., launchers or external communications) are also resolved.

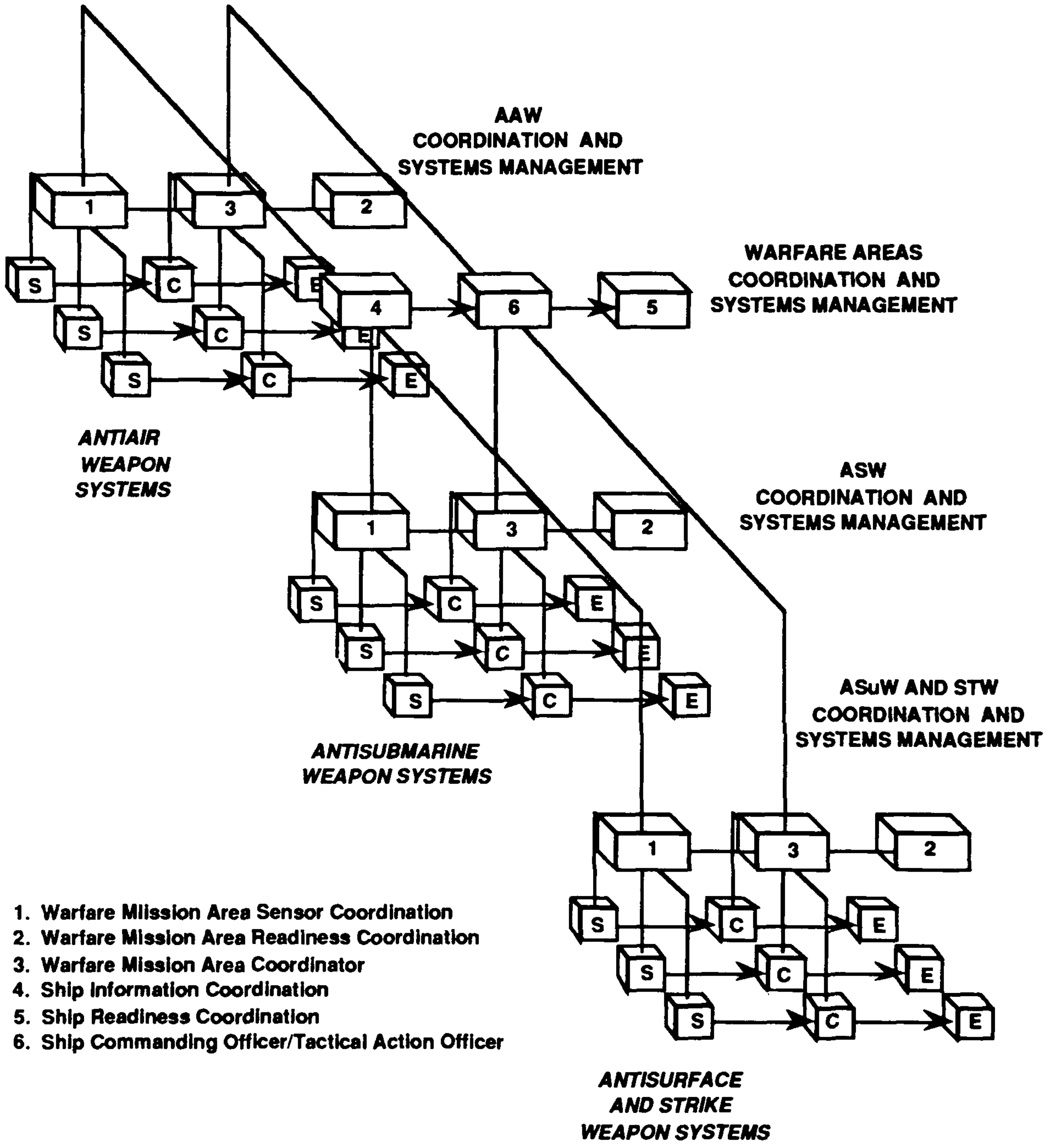

FIGURE 8. VERTICAL DECOMPOSITION

4.8 COMBAT SYSTEM SENSING RESOURCES

DECISION VIII: Individual warfare area sensing assets tend to lose their identity in the combat system as a sensing functional subsystem is formed (see Figure 9).

This decision was based on the following three architecture principles:

- Each element of the battle organization should have access to adequate information to perform its allocated functions. The information (database) used should be consistent and appropriate to the user system needs (not always common).
- To avoid ambiguous or conflicting reports, sensor information should be processed and transmitted to each user by well-defined and nonredundant paths.
- Important control decisions should be made at the point in the system architecture where all the relevant information has been brought together.

Once the flexibility provided by computer data transmission is taken into account, it is evident that tactical data flows cannot be handled by ad hoc procedures. The problems and opportunities of fusion alone suggest that tactical data flows will eventually be handled as a multipurpose utility. This must include situation data obtained through tactical networks, since own-ship communications equipments are the local sensing element for offboard information sources. However, each level of coordination may use some unique elements of information. For unit command, this may include receipt of orders or indications and warning reports needed for planning. For the individual weapon system operators, unique elements of target information or environmental data may similarly be needed. Thus, it appears that distributed fusion assets may be necessary.

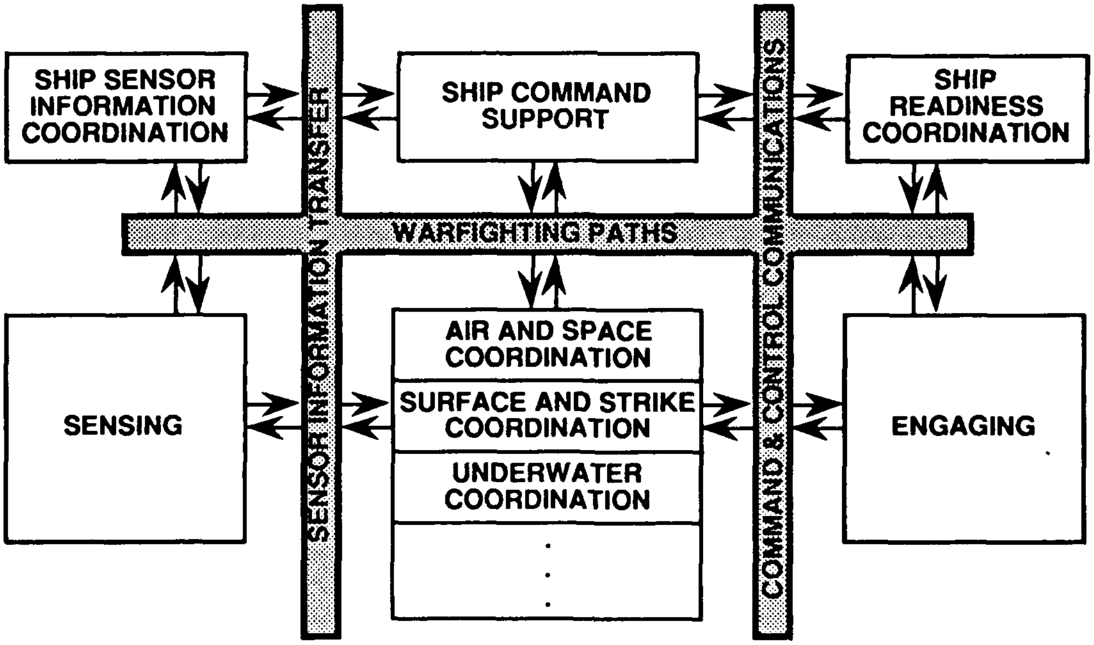

FIGURE 9. COMBAT SYSTEM FUNCTIONAL ARCHITECTURE

4.9 COMBAT SYSTEM ENGAGING RESOURCES

DECISION IX: As indicated in Figure 9 above, engaging assets for the warfare areas in the combat system are also collapsed into a functional subsystem.

This decision was based on the following three architecture principles:

- Provide for multiple control paths through the system to achieve greater survivability, flexibility, and growth potential than single path designs.
- Subsystems should have minimal crossover effects (interference or dependence) on operations of the other subsystems.
- Individual sense, control, and engage assets should be complete functional modules, able to operate independent of other elements, yet ready for interconnection with other weapon systems in any useful arrangement.

The rationale for consolidated control of engaging assets parallels that used for information flows (data paths) in Section 4.8 above. Vertical launch technology already in use makes launcher functions a shared utility. Current trends could lead to common launch control or weapon control equipment as well. Aircraft can aim, fire, and control a variety of weapons in very little space. Future ships can do the same to achieve simplified logistics, space and weight reductions, better use of manpower, and increased operational flexibility. Both sensing and engaging assets are instances of the potential for consolidation by function across the many warfighting paths that may exist in modern combat systems.

4.10 FORCE ARCHITECTURE

DECISION X: The functional architecture derived for surface ship combat systems can be extended to the force level as well (see Figure 10).

This decision was based on the following architecture principle:

- Carefully consider....the degree to which higher and lower-echelon systems will interact for control purposes. Integration at one level can interfere with integration at another.

The battle organization used for surface combatants is recursive in character, reflecting the composite warfare commander concept adopted at battle force level. This step simply recognizes that combat system and force architectures should be consistent and mutually supporting.

1. **Overall Tactical Commander (OTC) or Composite Warfare Commander (CWC)**
2. **Force Information Coordination**
3. **Force Readiness Coordination**
4. **Force Warfare Areas Commander**
5. **Force Warfare Area Information Coordination**
6. **Force Warfare Area Readiness Coordination**
7. **Ship CO or TAO**
8. **Ship Information Coordination**
9. **Ship Readiness Coordination**
10. **Warfare Area Coordinator**
11. **Warfare Area Information**
12. **Warfare Area Readiness Coordination**

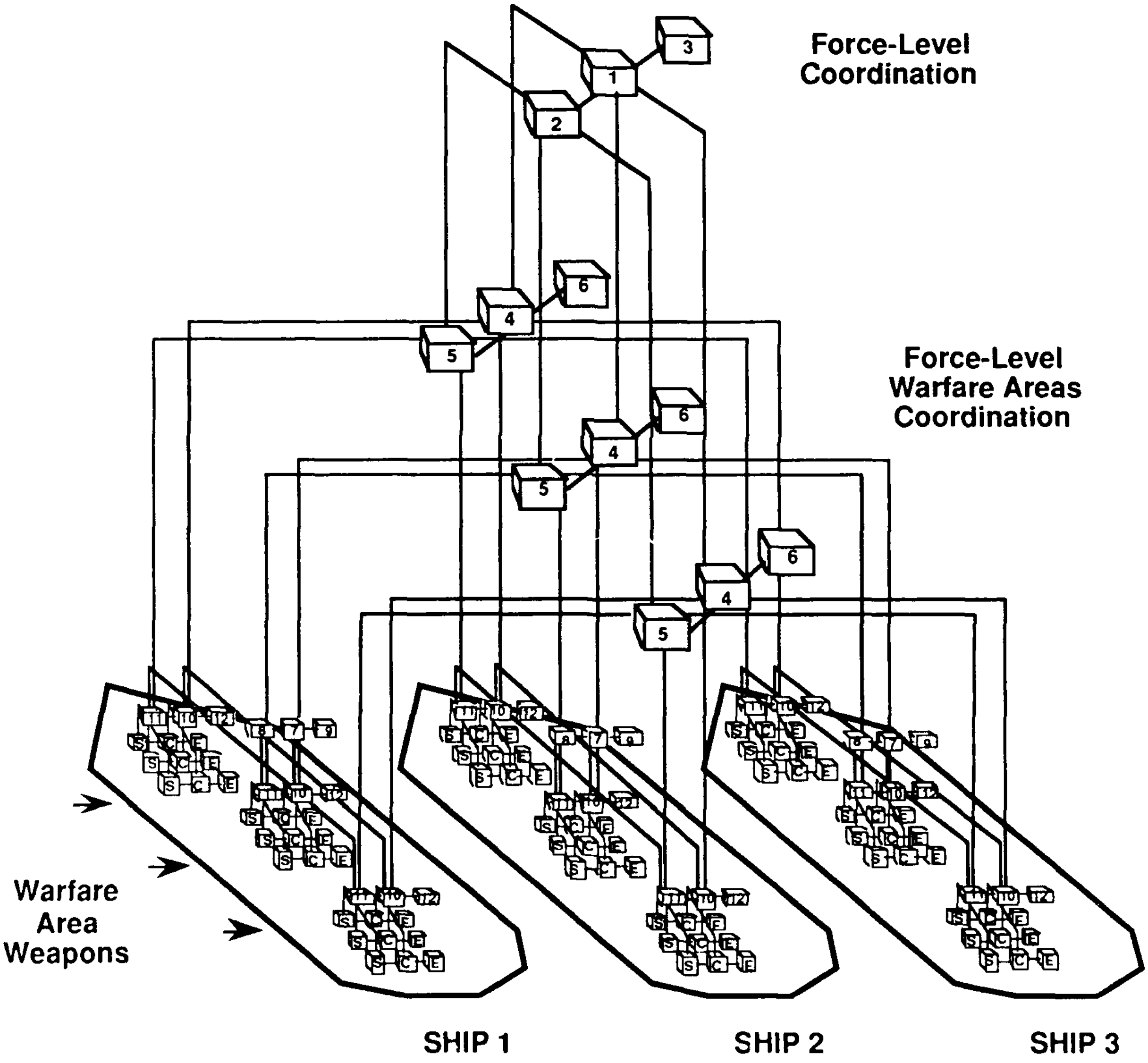

FIGURE 10. MATCHING COMBAT SYSTEM/FORCE ARCHITECTURES

5.0 BACKBONE CONTROL STRUCTURE

Methods of control theory are used in this chapter to derive the structure of the Vision Architecture from a general model of decentralized control. Section 5.1 establishes a modular approach to architecture design. Section 5.2 presents a model for decentralized control of complex systems, which is applied to combat systems at the unit level in Section 5.3. The model is extended to warfare area level in Section 5.4 and to clusters of individual weapon systems in Section 5.5. Together, Sections 5.3 to 5.5 generate a structure comparable to that of the Vision Architecture. Finally, Section 5.6 considers a widely used architecture for interconnection systems.

5.1 NETWORK OF MODULES

Generally, complex systems are said to be *modular* if they are composed of building blocks that can be added, removed, or interchanged to convert from one organization to another with different but usually similar functional properties. These building blocks, often called *modules*, represent physical, logical, or functional units with known properties and considerable internal complexity. Using installations may choose the modules that best meet present needs, including or omitting any optional modules, and so tailor the system configuration to its own operational needs. Finally, a malfunctioning unit may replaced with an identical, operable unit, improving ease of repair.

Fiorio and Villa observe that if each component is controlled by a dedicated decision agent, the pair can be regarded as a module of the overall system.[7] The combat system then constitutes a layered hierarchial network formed by nested composition of modules. Each module must be able to solve a decoupled control problem for its components, to include coordinating the behavior of any lower-layer modules nested within it. This means that information and control signals must be exchanged between each decision agent and all entities in the corresponding module. In particular, the system must provide as follows for each module:

- Application: Establish current operating objectives, configuration, and set points (control templates) for the module.
- Presentation: Maintain interfaces with decision agents providing situation assessment and control of module functions.
- Network: Exchange of information with related control elements to facilitate coordination between modules. Information and control signals must be exchanged between the decision agent and all component modules without excessive control efficiency loss in transport and disaggregation processes.
- Protocols: Provide for authentication, activation, and management of links to decision and action nodes.

- Virtual Connectivity: Provide physical and logical connectivity and route control for linkage to decision and action nodes.
- Algorithm: Exercise information processing resources to obtain needed judgments, forecasts, and/or perceptions. Suitably aggregated information must flow between the decision agent and any associated lower layers without excessive information losses.
- New Information: Task sensors and links (by invention and testing of hypotheses) to acquire the explicit knowledge needed for control tasks.
- Prior Knowledge: Provide for access to, and maintenance of, database elements containing prior tactical knowledge. In particular, the decision agent must be provided with the knowledge assets necessary for control: plant models, control efficiency measures, and a way of selecting appropriate control actions.

An architecture template for modules is given in Figure 11. The dynamical models and coordination efficiency measures for the modules are related by the following module inclusion principle: For a module at layer j, the corresponding dynamical model is the union of component models from lower layers of the network; and the coordination efficiency measure is the sum of the corresponding component efficiency measures and a measure unique to the current module. Given that links are defined by module inclusion according to this principle, the network will form a graph that contains no internal cycles or loops. The layers of the network are arranged in a generalized hierarchical structure, with each layer itself forming a network of modules.

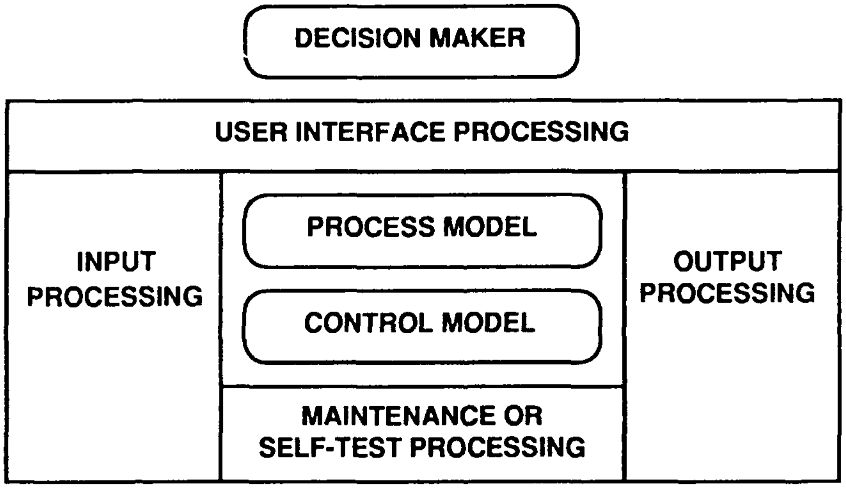

FIGURE 11. MODULE ARCHITECTURE TEMPLATE

In general, complex systems have a layered hierarchy of at least three layers. Entities residing on each layer have distinct responsibilities and functions, and are coupled only through message passing. Messages are exchanged only by well-defined interfaces between entities located on the same layer or adjacent layers. These messages imply functionality performed by a supplier entity on behalf of a using entity, and thus support identification of the functionality required for implementation. System physical assets, chiefly equipment, reside in the bottom layer.

Fundamental operating tasks appear in the top layer, which is the layer seen by the user. Intermediate layers map the task objectives into physical reality, representing the successive resource manipulations needed to complete an operation. This creates an audit trail for design to ensure that system implementation supports the user's operating concept.

Architectures that meet these conditions are said to contain strict separation of concerns, both horizontally and vertically. Vertical separation of concerns implies separation of task objectives into vertical layers, from the abstract to the concrete. Horizontal separation of concerns implies separation of functional objectives into distinct, independent entities. Cameron et al suggest experience has shown that architectures with strict separation of concerns are easier to develop and maintain.[8] In fact, virtually all architectures contain some separation, though not to the degree envisioned here. Further, the increasing complexity of systems is expected to make such architectures necessary in future developments.

The quality of system coordination achieved is limited by errors and inconsistencies among modules in terms of plant modeling, efficiency measures, communications, and control efficiency losses. The complexity of the plant model necessary for design depends on both the complexity of the physical system and on how demanding the design specifications are. An important tradeoff exists between complexity of a model and the feasibility of exercising it to aid design.

5.2 SYSTEM MODELING

Calvet and Titli consider a dynamic state variable model for a system S composed of N interactive subsystems.[9] The main results given in the paper are briefly reviewed here. In this report, a plain font is used with lower case letters to denote scalars, while upper case letters are used to denote sets. For example, x and t denote scalar variables, while m and n are positive integers. Greek letters are used at times for key parameters. Such letters as S and X are used to denote sets and transformations as indicated in the text. However, letters M and N are used to denote the limits of an index set. A bold font is used with lower case letters to denote vectors; and with upper case letters to denote matrixes. Thus $\mathbf{x}$ denotes an nx1 state vector and $\mathbf{x}(t)$ is a vector valued function of time. Similarly, $\mathbf{u}$ denotes an mx1 input vector. The symbols $\mathbf{A}$, $\mathbf{B}$ denote matrixes of dimensions nxn and mxn, respectively.

For each subsystem S_i, let $\mathbf{x}_i$ denote the local set of state variables, $\mathbf{u}_i$ the local set of controls, and $\mathbf{v}_i(\mathbf{z}_i)$ the corresponding input (output) coupling vectors. If $\mathbf{x}_i$ is an element of the vector space $\mathbf{X}_i$, and $\mathbf{u}_i$ belongs to the vector space $\mathbf{U}_i$, then the state $\mathbf{x}$ of the composite system S belongs to the product space $\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times \ldots. \times \mathbf{X}_N$. Similarly, the control $\mathbf{u}$ for S is an element of $\mathbf{U} = \mathbf{U}_1 \times \mathbf{U}_2 \times \ldots. \times \mathbf{U}_N$, and the composite input-output vector$(\mathbf{v},\mathbf{z})$ belongs to the interaction space $\mathbf{V}$. The component model for the subsystem S_i is then given by the following equations:

$$\partial \mathbf{x}_i / \partial t = f_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{v}_i) \qquad (1)$$

$$\mathbf{z}_i = \mathbf{e}_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{v}_i) \qquad (2)$$

$$\mathbf{v}_i = \mathbf{h}_i(\mathbf{z}_j, \mathbf{u}_j) \text{ for } j = 1,\ldots.,N \qquad (3)$$

where functions f_i, $\mathbf{e}_i$, $\mathbf{h}_i$ together with their first and second order derivatives are continuous in all arguments. The model is stated in nonlinear form for maximum generality.

Takahara[10] solves a linear version of this model using a quadratic index of performance

$$\mathbf{P}_D = \int_D \{\Sigma_I\ (\mathbf{x}_i\mathbf{Q}_i\mathbf{x}_i + \mathbf{u}_i\mathbf{R}_i\mathbf{u}_i)\ \}$$

where the domain of integration **D** is the interval $[t_0 \le t \le t_f]$ and the index set for summation is $I=\{1,....,N+1\}$. Each matrix $\mathbf{Q}_i$ is assumed to be symmetric and positive semidefinite, while the block-diagonal composite matrix $\mathbf{R}=[\mathbf{R}_1,....,\mathbf{R}_{N+1}]$ is symmetric and positive definite. To simplify the notation, $\mathbf{x}_{N+1}$ replaces **z** while $\mathbf{u}_{N+1}$ replaces **u'**. Performance indexes of this type are widely used in control system engineering to express consumed resources (energy, fuel, and time) or some other physical parameter associated with the trajectory of the system from its initial state to its final state. Thus performance is expressed in terms of expected costs for some period of operation. With an appropriate performance index, linear quadratic design methods can be shown to have desirable robustness and sensitivity properties.

The linear-quadratic formulation leads to an efficient two-level solution algorithm. The task of the higher level is to choose approximate values for the coupling inputs $\mathbf{v}_i$ and the LaGrange parameters $\mathbf{b}_i$ associated with the coupling constraints, based on stationarity conditions for the problem. For given values of the $\mathbf{v}_i$ and $\mathbf{b}_i$ the LaGrangian function for the overall problem is separable into N independent minimization problems. The lower level thus functions to optimize the subsystems independently (in parallel). This algorithm can be solved iteratively and has given satisfactory results in a variety of examples. The equations appear in nonlinear form as the most general statement of the problem. For computation, a series of linear approximations, each correct over a small part of the problem space, would most likely be used. Linear or quadratic performance indexes for the overall system and each subsystem would also be used.

The results given by Calvet and Titli are of special interest due to the next step.[9] By substituting Equation 2 into Equation 3, it is possible to obtain a composite equation

$$z = \mathbf{g}(\mathbf{x},\ \mathbf{u},\ \mathbf{v}) \tag{4}$$

which gives a static interconnection system for S. Since the research was motivated by an application with dynamical interconnections, the model was revised to treat the interconnection system as a separate subsystem, making N+1 in all. The equations for the revised system model S' then become:

$$\partial \mathbf{x}_i/\partial t = f_i(\mathbf{x}_i,\ \mathbf{u}_i,\ \mathbf{v}_i) \tag{5}$$

$$\mathbf{v}_i = \mathbf{h}_i(\mathbf{z},\ \mathbf{u}') \tag{6}$$

$$\partial \mathbf{z}/\partial t = \mathbf{g}(\mathbf{x},\ \mathbf{u},\ \mathbf{u}',\ \mathbf{v},\ \mathbf{w}) \tag{7}$$

$$\mathbf{w} = \mathbf{h}(\mathbf{z},\ \mathbf{u}) \tag{8}$$

where **w** is the input coupling vector and **u'** the control for the new subsystem. The algorithm outlined above is easily adapted to the revised system model S' resulting from this change. The higher level of the algorithm then sets values for coordination parameters **w**, **b'** and finds an optimal solution for the interconnection subsystem. The lower level sets values for coordination parameters $\mathbf{b}_i$, $\mathbf{v}_i$ and finds optimal solutions for the first N subsystems (in parallel).

A third model is given for cases in which the original subsystems are governed by different dynamics than the interconnection subsystem. This will hold for many practical systems, in which the modes of the interconnection system are slow compared to those of the original N subsystems. A temporal decomposition of the overall system S' can then be achieved. The *fast* part consists of N independent subsystems, with separable performance index; and the *slow* part consists of the interconnection subsystem.

5.3 UNIT LEVEL DESIGN

The starting point for control design is the unit level, at which the combat system is regarded as a composite of sensing, control, and engaging subsystems plus interconnections (see Figure 12). The corresponding dynamical equations are as follows.

$$\partial \mathbf{x}_1/\partial t = f_1(\mathbf{x}_1, \mathbf{u}_1, \mathbf{v}_1) \qquad \text{[Sensing Subsystem]} \tag{9a}$$

$$\mathbf{v}_1 = \mathbf{h}_1(\mathbf{z}, \mathbf{u}') \qquad \text{[Sensor input coupling]} \tag{9b}$$

$$\partial \mathbf{x}_2/\partial t = f_2(\mathbf{x}_2, \mathbf{u}_2, \mathbf{v}_2) \qquad \text{[Control Subsystem]} \tag{10a}$$

$$\mathbf{v}_2 = \mathbf{h}_2(\mathbf{z}, \mathbf{u}') \qquad \text{[Control input coupling]} \tag{10b}$$

$$\partial \mathbf{x}_3/\partial t = f_3(\mathbf{x}_3, \mathbf{u}_3, \mathbf{v}_3) \qquad \text{[Engaging Subsystem]} \tag{11a}$$

$$\mathbf{v}_3 = \mathbf{h}_3(\mathbf{z}, \mathbf{u}') \qquad \text{[Engage input coupling]} \tag{11b}$$

$$\partial \mathbf{z}/\partial t = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{u}', \mathbf{v}, \mathbf{w}) \qquad \text{[Interconnection Subsystem]} \tag{12a}$$

$$\mathbf{w} = \mathbf{h}(\mathbf{z}, \mathbf{u}) \qquad \text{[Interconnect input coupling]} \tag{12b}$$

where $\mathbf{w}$ is the input coupling vector and $\mathbf{u}'$ the control for the interconnection subsystem. The equations are given in nonlinear form, according to Equations 5 through 8 of Section 5.2, as the most general statement of the problem. For computation, a series of linear approximations, each correct over a small region of the problem space, would most likely be used. Linear or quadratic efficiency measures, corresponding to the overall system and each subsystem model, would also be employed. The indexes have the form

$$\mathbf{P[D,I]} = \int_D \{\Sigma_I \, (\mathbf{x}_i\mathbf{Q}_i\mathbf{x}_i + \mathbf{u}_i\mathbf{R}_i\mathbf{u}_i)\}$$

where I={1}, for example, gives the performance index for sensing. This begins a top-down partitioning of the combat system on functional lines, as opposed to a bottom-up procedure beginning with warfighting paths. Since combat systems are warfare systems, just as infantry battalion or tank companies are warfare systems, they can be broken down into sense, control, and engage elements. The functions assigned to each subsystem are indicated in Figure 12. This gives a starting point for control design by a top-down process. Comparing the control structure derived to the goalpost architecture identified previously may indicate if the solution is sensitive to the chosen point of entry, or to the particular sequence of decisions considered.

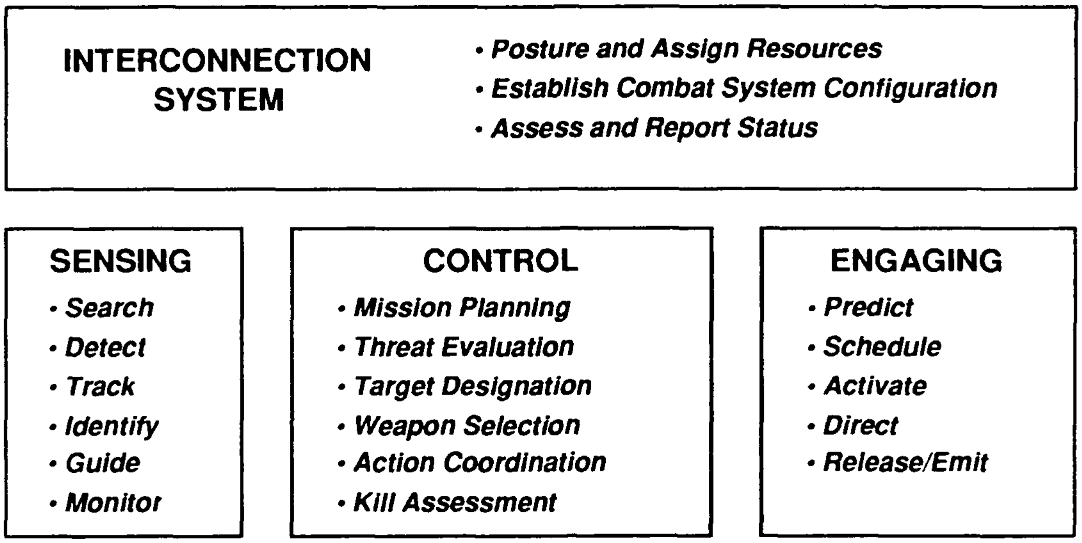

FIGURE 12. NETWORK OF MODULES–UNIT LEVEL

As stated in Section 5.1, each module contains a decision agent and must make provisions for exchange of information and coordination signal flow to and from component entities. This suggests a decomposition of interconnections at each level into information, readiness, and decision categories. Measures for consolidation of such assets at unit level involve a wide range of design issues and trades.

In addition, control of the ship's service infrastructure (electrical power, communications, piping, and mobility) must be considered an important function at this level. Since they tend to cut across weapon system and warfare area product lines, the importance of service infrastructure is easy to overlook. A model formulated at the unit level encourages proper attention to these factors.

5.4 WARFARE AREA DESIGN LEVEL

Naval forces operate in three domains (air, surface, and undersea) with radically different physical characteristics. This has long been a major factor in naval battle organization and weapon systems usually are designed to work in a particular domain. Most surface combatants are equipped for operation against threats in each domain; that is, with subsystems specialized for AAW, ASW, and ASuW. Since the characteristics of action against land targets resemble those of ASuW, it is customary to treat strike/antisurface warfare (STK/ASuW) as a single (multipurpose) subsystem.

The battle organization for surface combatants is based on the principle of decentralized command, with warfare area coordinators delegated to handle each warfare area subsystem. The subsystems interact weakly and require minimal coordination, while the separate control systems permit simultaneous multiwarfare operations. The decentralized command concept reflects and supports the composite warfare commander concept typically employed for naval battle forces (see Figure 13).

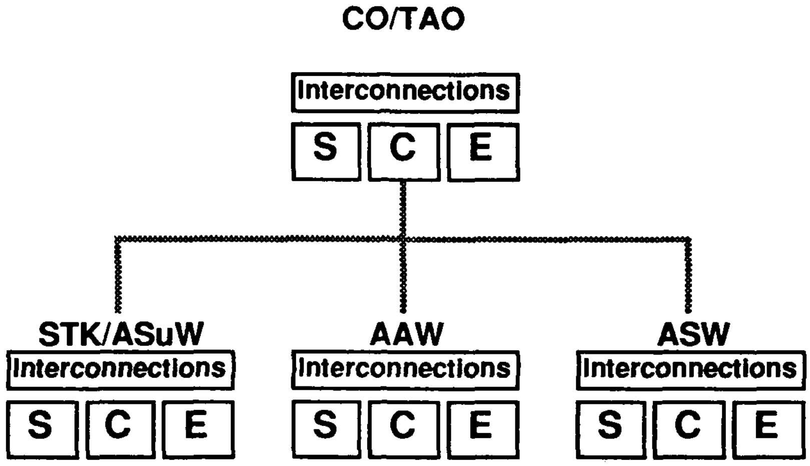

FIGURE 13. NETWORK OF MODULES–WARFARE AREA VIEW

Maintaining the view of combat systems as a layered network of modules, the breakdown into warfare areas is illustrated in Figure 13 above. Since each warfare area coordinator will have a local interconnection system to manage, as will the unit command authority, the backbone control system must be designed specifically to implement and support the desired battle organization. Both the unit commander and warfare area coordinators may need to retain capability for autonomous action. The dynamical equations for the warfare area level are as follows:

$$\partial x_{1j}/\partial t = f_{1j}(x_{1j},\ u_{1j},\ v_{1j}) \qquad \text{[Sensing subsystems]} \qquad (13a)$$

$$v_{1j} = \mathbf{h}_{1j}(\mathbf{z},\ \mathbf{u}') \qquad \text{[Sensor input couplings]} \qquad (13b)$$

$$\partial x_{2j}/\partial t = f_{2j}(x_{2j},\ u_{2j},\ v_{2j}) \qquad \text{[Control subsystems]} \qquad (14a)$$

$$v_{2j} = \mathbf{h}_{2j}(\mathbf{z},\ \mathbf{u}') \qquad \text{[Control input couplings]} \qquad (14b)$$

$$\partial x_{3j}/\partial t = f_{3j}(x_{3j},\ u_{3j},\ v_{3j}) \qquad \text{[Engaging subsystems]} \qquad (15a)$$

$$v_{3j} = \mathbf{h}_{3j}(\mathbf{z},\ \mathbf{u}') \qquad \text{[Engage input couplings]} \qquad (15b)$$

$$\partial z_j/\partial t = \mathbf{g}_j(\mathbf{x},\ \mathbf{u},\ \mathbf{u}',\ \mathbf{v},\ \mathbf{w}) \qquad \text{[Interconnection subsystems]} \qquad (16a)$$

$$\mathbf{w}_j = \mathbf{h}_j(\mathbf{z},\ \mathbf{u}) \qquad \text{[Interconnect input couplings]} \qquad (16b)$$

where the subscript j denotes the particular module described by the set of equations given, with j=0 for the unit, j=1 for the STK/ASuW warfare mission area, j=2 for AAW, and j=3 for ASW. Thus the components shown for the unit level model are essentially broken down into warfare area components. Since the unit level module contains each of the warfare area modules, the equations indexed by j=0 represent components that do not fit into the warfare mission areas; e.g., reserve or multipurpose assets. If it were desired to explicitly represent other areas, such as mobility, a corresponding module would be added. Performance indexes take the form

$$\mathbf{P[D,I\times J]} = \int_{\mathbf{D}} \{\Sigma_{\mathbf{I\times J}} \, (\mathbf{x}_{ij}\mathbf{Q}_{ij}\mathbf{x}_{ij} + \mathbf{u}_{ij}\mathbf{R}_{ij}\mathbf{u}_{ij})\,\} \tag{17}$$

making the same use of subscripts i and j as in Equations 13 to 16.

5.5 CLUSTERED WEAPON SYSTEMS LEVEL

Warfare area modules break down further into clusters of weapon systems, as shown for AAW Figure 14 below. The figure refers to surface launched missiles, manned aircraft and anti-leaker defenses. The last includes electronic support measures (ESM) equipment, electronic countermeasures and a CIWS. The CIWS, however, could use SeaSparrow or RAM instead of a Gatling gun system. The dynamical equations given by Equations 13 through 17 above apply to this level as well, once a subscript (say, k) is added to index the weapon clusters.

Antileaker defenses could mean a new level of system integration, a design issue of possible importance. The integration opportunity occurs below the level of the antiair warfare controller (AAWC) but above the level of individual weapon systems. This may signify a change of the level at which the mission organization should give way to a functional approach.

Beyond this third layer lies a layer containing the individual components of the combat system. The modules involved in a discrete action sequence link together to form a warfighting path. This represents a virtual path rather than a physical path and may be significantly longer than the direct action path.

5.6 LAYERED ARCHITECTURE FOR INTERCONNECTIONS

Treating interconnection structure as a distinct subsystem brings the role of combat system integration into focus. The array of computing, communications, stored program, and knowledge base assets that link command decision-makers to sensors and actuators are an essential concern of combat system design. A layered architecture for systems of this type, as shown in Figure 15 below, is widely used for both industry and military applications. This architecture has five separate yet interrelated layers, as shown by a pyramid with the command element at the top and the delivery system at the bottom. The first four levels are logically connected in top-down fashion. The fifth, the delivery system architecture, is the foundation architecture. Created to satisfy requirements of the other levels, its success is dependent on definition of relevant operating goals and objectives. Each level may contain multiple components as well as a set of discretionary and nondiscretionary standards for the enterprise.

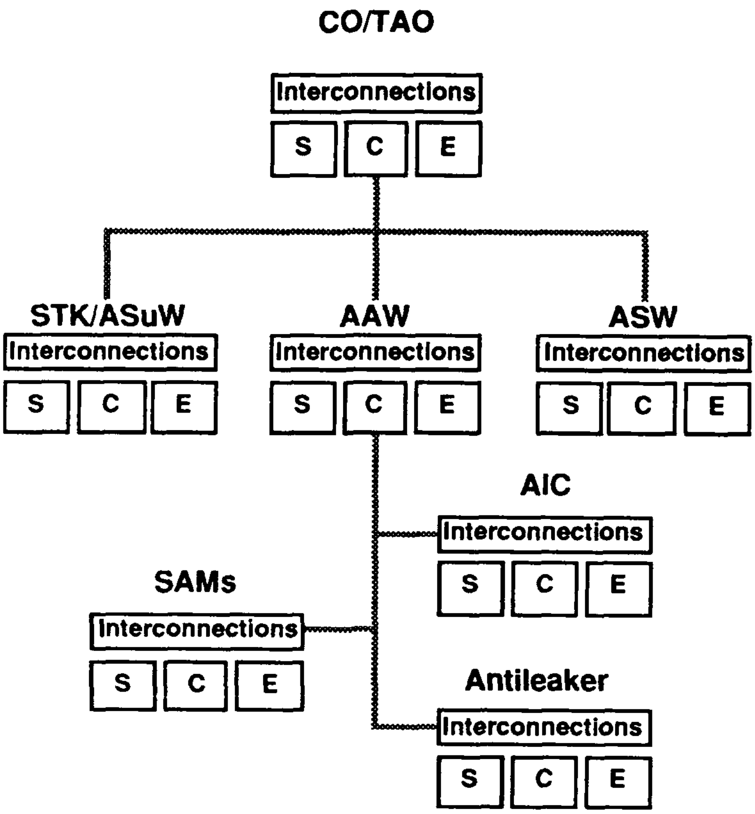

FIGURE 14. NETWORK OF MODULES–AAW WEAPON CLUSTERS

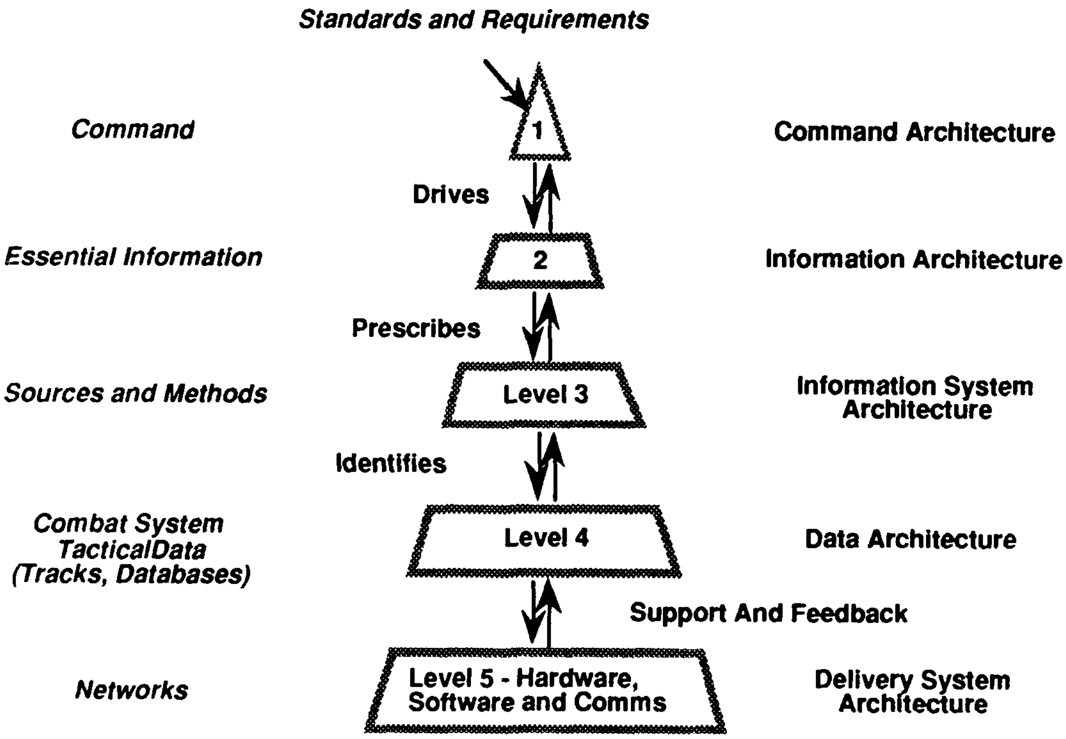

FIGURE 15. COMBAT SYSTEM TACTICAL INFORMATION

Architecture at the *Command* level establishes a framework for satisfying both the internal information needs and those imposed by external activities. The latter includes friendly forces, adversaries, and higher level command elements.

Architecture at the *Information* level establishes a framework to meet the information needs of the *Command* level. It specifies content, presentation form, and format of the information, thus establishing requirements for the *Information System* architecture.

Architecture at the *Information System* level establishes a framework for meeting the specific requirements of the *Information* level. Its components include automated and procedure-oriented information systems supporting internal and external information flows. They are used first to acquire and process data, then to produce and distribute information in accordance with requirements and standards. Logical database designs occur at this level.

Architecture at the *Data* level establishes a framework for maintenance, access, and use of the data of the enterprise. The data should meet the standards of all higher levels of the architecture, especially *Command.* Key components include data models that support physical database design; database and file structures; and data definitions, dictionaries, and data elements that underly the information systems of the enterprise. The creation of a data dictionary and associated naming conventions is an important aspect of the Data Architecture, because these conventions establish the vocabulary necessary for human communication.

The *Delivery System* architecture is a technical implementation to meet the requirements of all higher levels. Key components include computers, communications, and computer programs needed to support the *Data* and *Information System* levels of the enterprise architecture. Infrastructure and facility support assets needed to properly accommodate and connect the components in an integrated manner are also included. Since a collection of processing elements embedded in an interconnection network comprises a distributed computing system, distributed control computing is a critical technology associated with interconnection systems.

6.0 REFERENCES

1. Lindemann, M.J.; Grubbs, J.L., Jr.; Eason, W.J.; Lehmann, L.L.; Shields, E.F.; and Evers, T.H., MODFCS Architecture Baseline: A Combat System Architecture. NAVSWC TR 79-357, NAVSWC, Dahlgren, VA, January 1980.

2. Cullen, R. P., *A Concept for Future Combat Systems: An Enduring Structure*, NAVSWC TR 90-121, NAVSWC, Dahlgren, VA, 1990.

3. Cullen, R.P.; Hill, L.; et al, *Combat System Architecture: Description and Use in Surface Combatants*, NAVSWC, Dahlgren, VA, March 1981 (unpublished report).

4. Pollard, James R., *Combat Systems Vision 2030: A Combat System Architecture for Future Surface Combatants*, NAVSWC TR 91-607, NAVSWC, Dahlgren, VA, September, 1991.

5. Blanchard, B.S. and Fabrycky, W.J., *Systems Engineering and Analysis*, New Jersey, 1981.

6. Rogan, J.E. and Cralley, W.E., Meta-Design–An Approach to the Development of Design Methodologies; IDA Paper P-2152, January 1990.

7. Fiorio, G. and Villa, A., "A Comparative Analysis and Evaluation of Decentralized Control Structures," *Large Scale Systems: Theory and Applications*, Pergamon, Proc. IFAC Symposium 1986, pp. 335-339.

8. Cameron, E.J., Petschenik, N.H., Ruston, L., Shah, S., and Srinidhi, H., "Description to Simulation to Architecture: An Approach to Service-Driven Development," *Proceedings of the First Int. Conference on Systems Integration*, April 1990, pp. 264-274.

9. Calvet, J.L., and Titli, A., "Hierarchical Optimisation and Control of Large Scale Systems With Dynamical Interconnection System," *Large Scale Systems Theory and Applications*, Pergamon, Proc. IFAC Symposium 1981, pp. 117-126.

10. Takahara, Y., *Multilevel Structure for a Class of Dynamical Optimization Problem*, Master's Thesis from Case Western Reserve University, Cleveland, Ohio, 1965.

7.0 BIBLIOGRAPHY

Birdwell, J.D.; Costanon, D.A.; and Athans, M., *On Reliable Control System Designs*. SMC-16(5): 703-711, 1986.

Callender, E.D., Hartsough, C., and Morris, R.V., *Relations Between Information System Engineering and Software Engineering*, IEEE 1981 Int. Conference on System Engineering, 1981.

Chorafas, D.N., *Systems Architecture & Systems Design*, McGraw, 1989.

Crecine, J.P. and Salomone, M.D., *Organization Theory and C3; in Science of Command and Control: Coping with Uncertainty*, Johnson and Levis (eds.), 1989.

Duren, B.G. , *The Orgins of System Engineering: A Survey of Practice in the Bell System*, NAVSWC TR 91-141, NAVSWC, Dahlgren, VA, May 1991.

Fong, E.N. and Goldfine, A.H. (eds.), *Information Management Directions: The Information Challenge*, PB90-219866, NIST Special Publication 500-167, September 1989.

Marr, David, Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information, Freeman Press, San Francisco, 1982.

Noble, D. and Mullen, R., *Information Presentations for Distributed Decision Making; in Science of Command and Control: Coping with Uncertainty*, Johnson and Levis (eds.), 1989.

Özgüner, U., *Decentralized and Distributed Control Approaches and Algorithms*, IEEE Proceedings of the 28th Conference on Decision and Control, December 1989.

Singh, M.G., Titli, A. and Malinowski, K., "Decentralized Control Design: An Overview," *Large Scale Systems: Theory and Applications*, Pergamon, Proc. of the IFAC/IFORS Symposium, July 1983, pp. 1-15.

Surface Ship Combat System Operational Philosophy, CNO letter serial 03/70258, 8 November 1976.

APPENDIX A

ARCHITECTURE DEFINITIONS

ARCHITECTURE DEFINITIONS

The following definitions shed light on the meaning and nature of the term, combat system architecture.

1. *Dictionary of Computing*, 3rd Edition; Oxford U. Press, 1983.

   "An architecture is the specification of a system at a somewhat general level, encompassing the nature, configuration and interconnection of its major elements."

2. Oxford Dictionary of the English Language, Oxford U. Press, NY 1983.

   "Architecture is the art or science of designing and framing complex structures of any kind for human use. Regarded in this wide application, architecture is divided into civil, ecclesiastical, naval, and military branches, dealing respectively with houses and other buildings (such as bridges) of ordinary utility, churches, ships, and fortifications. In connection with computers or computer-based systems, the term refers to the conceptual structure and overall logical organization of a system from the point of view of its use or design, or to a particular realization of this. But architecture is sometimes regarded solely as a fine art, and then refers more narrowly to the adornment of edifices raised by man."

3. J.M. Rosenberg, Dictionary of Computers, Data Processing, and Telecommunications; Wiley 1984.

   "An architecture is a specification which determines how something is constructed, defining functional modularity as well as protocols and interfaces which allow communication and cooperation among modules."

4. M.E. Melich, Electronic Warfare Integration into a Combat System; pages 1-4, Proc. 7th MIT/ONR Workshop on C3 Systems, June 1984.

   "Melich reports a debate was conducted during the 1970s on the nature of a combat system architecture. Though no uniform technical definition was found, a useful operational definition did evolve: that is, a systems architect succeeds if his system is so partitioned that, over its entire life span, (1) subsystem interfaces are clearly defined; (2) qualified suppliers exist for all components; (3) operators can make it work in the real world; and (4) system acquisition and support are affordable."

5. David C. Opferman, A Design Methodology for System Quality, AT&T Technical Journal 65(3): 60-72, May/June 1986.

"System engineers translate customer needs into detailed system requirements independent of structure. System architects allocate the requirements to components and elaborate the resulting structure as necessary to identify all system components, along with their relationships, their interfaces, and perhaps their residency in hardware or software."

6. J.A. Zachman, A Framework for Information Systems Architecture; IBM Systems Journal 26(3): 276-292, 1987.

   "A system architecture is a logical construct for defining and controlling the system's components and interfaces. Zachman argues that in a discipline where construction of complex engineering products is a primary objective, not one but a set of architectural representations must be produced, each one depicting the perspective of a key participant such as owner, designer or builder."

7. Douglas M. Considine (ed.), Van Nostrand's Scientific Encyclopedia, 7th Edition; Van Nostrand-Rheinhold, NY 1989.

   "The organization of a control system is frequently referred to as its architecture. In this concept, the control system includes not only those elements (sensors, measurement, decision making, actuation and feedback) that make control of a process or machine possible, but also numerous support functions, particularly the data display and analysis functions which assist in the making of human as well as automatic decisions."

8. D.N. Chorafas, Systems Architecture and Systems Design; McGraw-Hill, New York 1989.

   "System architecture provides a unifying, coherent, logical structure permitting integration of a number of devices not necessarily compatible among themselves. Such devices may be mainframes, minis, personal computers, self-standing databases and so on. The system architecture makes feasible not only a significant ability to integrate but also flexibility to further expension and transparency for many functions which otherwise would have burdened the user. It encompasses style and norms of design based on consistent rules defining the interface structure between entities and the way they interact to form an integral system. It employs design principles, defines relationships between components and assures conformity to norms. This facilitates the interaction of attached devices including hardware, basic software and applications programs. At higher levels it defines formats which are compatible among dissimilar enterprise-wide systems."

APPENDIX B

SUBSYSTEM ARCHITECTURE PRINCIPLES

1.0 ORGANIZING FRAMEWORK

As indicated by Figure B-1, combat systems consist basically of people, procedures, and physical plant. This breakdown can be used to organize principles of subsystem architecture design. The principles thus support elaboration and/or extension of the combat system architecture within selected categories of subsystems. The battle organization category includes control structures for both unit command support and warfare area coordination. The physical plant category contains the corresponding weapons elements–the material resources used in the combat system. Finally, the information systems category includes sensing elements that support execution of warfighting procedures.

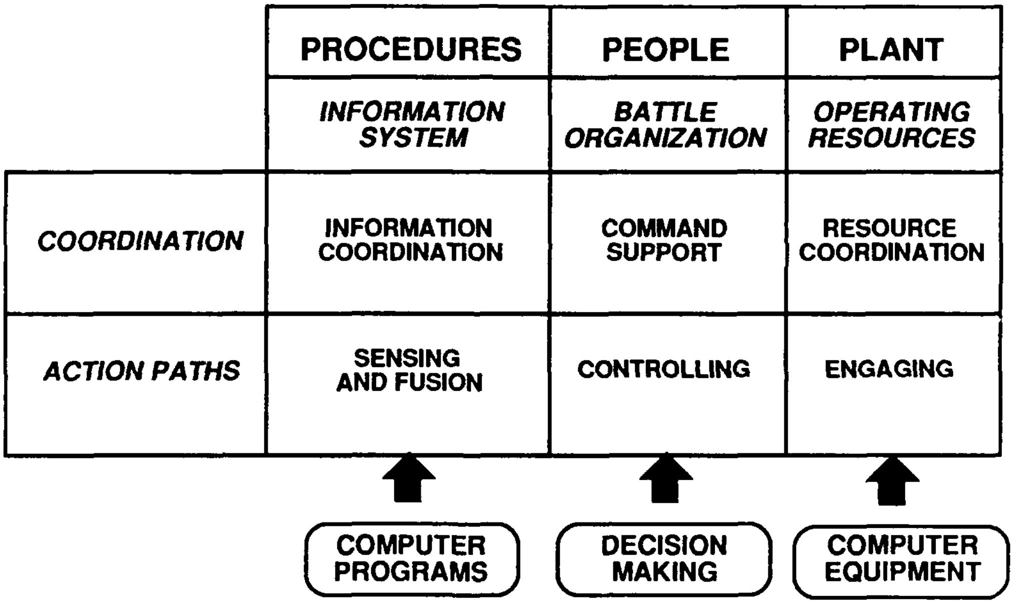

FIGURE B-1. COMBAT SYSTEM DESIGN FRAMEWORK

The unit level principles presented herein have been drawn from various combat system engineering and general system engineering sources. Section 2.0 on procedures draws from research on fusion principles (Reference B-1) and research on situation assessment (Reference B-2). Section 3.0 dealing with people is based on principles of decentralized command, derived from the composite warfare command concept and related work by Reference B-3. Section 4.0 on the physical plant draws from generic resource management requirements identified in research performed for the National Air and Space Agency (NASA); see Reference B-4.

2.0 INFORMATION COORDINATION (PROCEDURES)

2.1 INFORMATION SYSTEM ENGINEERING APPROACH

The growing importance of computers and automation reveals the central role of information in combat system performance. For any combat system, the key question for control structure design is to determine who makes the important control decisions. The answer to this question, more than any other factor, determines the overall architecture of the system. Trying to build a combat system without a concrete approach to this problem is to invite disaster.

In short, at the heart of every combat system is a control structure driven by an embedded information system. Information considerations dominate the architecture of a combat system because sensors and other information sources are often shared among systems. By contrast, sharing of weapons occurs much less extensively. Where there is sharing, there is contention, and resolving the various kinds of contention drives how the combat system should be put together.

In the field of Information System Engineering, two key problems were recognized: (1) transition from strategy to implementation and (2) a plethora of development tools and methodologies. Architectures were evidently important for (1), but the notion had not yet been clearly defined; it was *fuzzy* because people in different roles perceive architectures differently. Table B-1 contains a framework developed by Zachman (Reference B-5) that shows these different roles and provides a context for understanding and evaluating the purpose of each of the tools and methods. The idea here was first to understand the enterprise as a system in its own right and then to overlay that system with the infrastructure to support it.

Currently, industry activity in the area of information system development is high. In essence, corporations have automated their manufacturing processes and are now looking at the way information flows within the company. Information flow is viewed as the next area for modernization and cost containment.

Reference B-6 solves a longstanding riddle by noting that the basic purpose of information is to control action in an organized system. Thus, if an organized system is a set of elements that interact to produce some phenomenon of interest, then information may be defined as that which is exchanged by components of an organized system to effect their interactions. This definition permits a distinction to be drawn between information and data. In addition, it shows clearly that information and control structures are inseparable. The array of computing, communications, stored program, and knowledge base assets that link command decision-makers to sensors and actuators are thus the essential concern of combat system design and development.

Reference B-1 identifies some design principles for use in command and control system automation. The principles are stated in terms of surveillance data fusion applications, but they have meaning in a broader context as well. Each control action is based on situation assessment for a controlled plant or subsystem. In particular, control actions are based on state feedbacks, battle doctrine, orders, warfighting objectives, and constraints. The possible use of those principles for design of control structures is thus a matter of interest.

TABLE B-1. INFORMATION SYSTEM ARCHITECTURE FRAMEWORK

DESCRIPTION	SYSTEM DATA	INFORMATION SYSTEM PROCESSES	NETWORK
Objectives and Scope	List of Entities Important to Operations	Combat System's Major Processes	Correspondents in Battle Space
Combat Operations Model	Entity/Relationship Diagrams	e.g., Functional Flow Diagrams	e.g., Logistics and Tactical Networks
Information Systems Model	e.g., Data Models	e.g., Data Flow Diagrams	e.g., Distributed System Architecture
Technology Model	Data Design (e.g., Entity = Segment and Rein = Pointer)	e.g., Structure Charts	e.g., System Architecture
Detailed Description	e.g., Database Fields/Addresses	Computer Programs	e.g., Network Architecture

2.2 PRIOR KNOWLEDGE

The following are combat system architecture principles dealing with knowledge bases that exist in the combat system:

- Provide for access to essential prior knowledge through database or knowledge base elements.

- Efficient database sharing is perhaps the most crucial factor in combat system information coordination. Global databases should be made available for use in planning, coordination, threat assessment, targeting, intelligence production, and status monitoring. A variety of command displays and decision aids may be used, from specialized data association and fusion techniques to general purpose word processing, spreadsheet, text search, database management, and graphics applications.

- Provide for keeping databases of different security classification separated as much as possible, to avoid security-level creep and associated contamination of stored data.

2.3 KNOWLEDGE ACQUISITION

The following are combat system architecture principles dealing with acquiring knowledge for use by the combat system:

- Provide elements for acquiring explicit knowledge needed for use in assessment tasks by the following means:

  - report filtering and extraction
  - inventing and testing hypothesis

  - report filtering and extraction
  - inventing and testing hypothesis
  - managing sensors to gain needed information

- Form system information resources into modules that leave similar sources within a single subsystem, while dissimilar sources fall into different modules.

- Do not mix disparate reports and requirements. For instance, avoid mixing security classifications because the results inherit the highest security classification and must be inspected before dissemination. Where it occurs, such inspection tends to slow reporting. This can be a significant constraint on the data connectivity digraph, and has implications for handling of tactical signals exploitation reports.

- Detail of information sought should match the level of decision that it supports.

- Implement all useful sensor to sensor relationships while relying on the user community for determination of what is useful.

- Each sensor system should determine and report the variances of its own measurements. Since a statistic such as area of uncertainty makes this operationally convenient, the simple step of reporting covariance information across system interfaces is a must. An important corollary is that reports provided to fusion systems should be unadulterated. Smoothing filters combine the results of many measurements and therefore produce highly correlated results.

2.4 STRUCTURED COMMUNICATIONS

The following are combat system architecture principles dealing with communications between components of the combat system:

- Make information available to the users who need it. Information available to each warfare coordinator and subsystem should be at a level of detail appropriate to its intended use.

- Provide structured communications, including associated physical, protocol, and data connectivity for sharing of essential problem data across function and component lines.

- Delays in availability of data should be minimized to support time-sensitive operations.

- Provide for essential and capable communications between elements for reliable dissemination of messages carrying requirements, commands, warnings, and status information.

- Provide intership communications for reliable dissemination of messages carrying requirements, commands, warnings, and status information.

- Participate in tactical networks to exchange track file data with other tactical units, correlating it with own-ship data and providing for display of an extended-horizon tactical picture.

- Provide for communication with cooperating units and commands of other U.S. and allied services, as appropriate for support of joint and/or combined operations.

- Avoid bottlenecks in people, computers, and/or interfaces. Relationships between elements should be considered from the broadest viewpoint to exploit composite capabilities in a synergistic manner. At the same time, interfaces should be kept to a minimum and functional relationships should remain clear and well defined. Dependence on single lines of communication should be minimized both from the standpoint of single point of failure as well as the potential for overload and consequent degradation of system performance.

- Provide for voice and data networks needed to support tactical operations.

- Avoid multiple sensor data paths: Provide for control of data flow paths so that the connections between each controller and his control information sources form a direct and nonredundant path without internal cycles or loops. This is necessary to avoid error cascades arising from self-sealing fusion processes and the handling of conflicting reports or hypotheses.

- Users holding sensor reports in common should be able to exchange data fusion decisions about the reports. Only by feedback can every node in the combat system be provided with information consistent with the big picture finally derived by the overall system. (This has important benefits in the task of track number assignment.)

- Allow data of different security classifications to be kept separate as long as possible (to avoid security-level creep and consequent transmission delays). This can create a significant constraint on the data flow structure.

- Protocol connectivity should be a subset of physical connectivity, and data connectivity should be a subset of protocol connectivity.

- Avoid mixing disparate timing requirements (slow vs. fast and synchronous vs. asynchronous), event types (periodic vs. aperiodic), and information or data control disciplines.

2.5 PROCESSING

The following are combat system architecture principles dealing with the processing of information within the combat system:

- Guaranteed performance (latency rather than efficiency) is the ultimate measure of real-time information processing. Sufficient computer resources should be provided to perform complex and time-critical functions without excessive queuing and consequent delays. A distributed computing approach, if properly implemented, greatly reduces potential for overload inherent in a single centralized computer complex.

- Provide for information processing techniques as necessary to produce required judgments, forecasts, and perceptions.
- Form and maintain an adequate tactical picture, including such items as surveillance envelope, friendly force disposition, own-ship equipment status, and assigned force resource status.
- Support use of a common force coordinate system, giving accurate gridlock for all air, surface, and subsurface tracks.
- Provide for correlation and fusion of reports from dissimilar sources.
- Maintain multiuser, multilevel security of information within all information processing functions.
- Do not replace current sensor reports with prior information. The system should be able to perform its fusion functions without the use of look-up tables, for example. Excessive dependency on database or library information can make an automated system vulnerable to deception.
- Do fusion before classification, where the latter is the process of figuring out what physical object a track represents. There are two competing approaches. One is to base classification at each fusion step on results derived at the previous steps. Unless handled very carefully, this tends to result in cascaded guesses and frequent error. The preferred approach is to defer classification until the fusion process for a sensor report is complete.
- Provide for measures to ensure use of a consistent information set across the entire span of action (for each controller). This principle extends to both tactical picture data and procedural information (such as decision rules) used across battleforce elements.
- Ensure that the system information structure consistently supports and is consistent with the organization and system structure.
- Provide for integration of information from sources with common data and measurement structures first. It is commonality of measurement spaces that defines similar sources rather than target type, medium, sensor type, or any of the other reasons that might be invoked to justify a particular data fusion method. Mathematical properties make angle measurement a different process than measurement of range radio frequency (RF) or pulse repetition frequency (PRF). Specifically, angle is measured on a compact space. In multitarget tracking, angle track density increases rapidly without limit, even if track density in other dimensions remains small. Association errors then multiply. This suggests that

passive intercepts should be fused before they any attempt is made to combine them with active sensor data and/or imagery.

- All data fusion processes should consider the accuracy of the data to be combined. Association activities should proceed from the most accurate sources to the least accurate sources. Even if a system employs sophisticated algorithms for data association, the algorithms work most efficiently using the best data first.
- Determine and report the quality of all information processing tasks to users. (An observability condition for control feedback.)
- Base information integration and fusion elements on operational requirements.
- The design should reflect the fact that interprocess message transit delays are variable and some nonzero time always exists between production of an event by a process and the materialization of this production at the destination process location (different from observation of the event by the destination process).

3.0 WARFARE COORDINATION AND CONTROL (PEOPLE)

The basic mechanisms for achieving coordination in human organizations are direct supervision, mutual adjustment, standardization, and explicit planning. By whatever means, achieving coordination is costly. Organizations therefore devise strategies for limiting the degree of coordination attempted.

3.1 COORDINATION STRATEGIES

3.1.1 Decomposable Task Structures

A task structure is nearly decomposable if, in the short run, subtasks can be performed without regard for how others are being performed, and in the long run, depend only on a few aggregate characteristics of how other tasks are being performed. The amount of information that must be gathered and transmitted (in such cases) is much smaller than in highly interdependent task structures. The planning processes involved in creating a decomposable task structure are not cost-free, but changes in one component will have little effect on others, and repairs are accordingly easier to make.

Hierarchical control structures are especially useful when the task structure is decomposable (or nearly so) and they require less communication than other approaches. If communication is strictly hierarchical, each actor communicates with only one superior and a small number of subordinates. In addition, the complexity of a hierarchical organization is nearly constant across all levels as far as the individual actor is concerned. This is helpful because the ability to think about more than one complex task at once is not likely to vary significantly among individuals at different levels in the hierarchy. There is an advantage also in planning and control processes.

3.1.2 Ignoring Interdependencies

This approach reduces coordination cost, but may lead to situations in which subunits act at cross purposes. For example, the tragic APOLLO I fire that killed three astronauts was facilitated by coordination failures between design groups. One chose a 100-percent oxygen environment for the capsule while another chose to use materials that were inflammable in most environments, but highly flammable, in fact virtually explosive, in the 100-percent oxygen atmosphere of the capsule. The costs of ignoring all but exceptional cases of interdependence are similar though if the exceptions are well chosen, the costs may be considerably smaller.

3.1.3 Creating Buffer Stocks or Slack Resources

Many coordination problems arise when one subunit requests resources or assistance that are to be provided by another subunit. Short-term outages are likely to create delays in filling such requests. A crude solution is to hold large buffer stocks of slack resources to be used for meeting such demands. However, this may be costly.

3.1.4 Reliance on Subunit Resources

Another possible approach is to establish flexible, general purpose resources within the subunits. Such resources are likely to be more expensive than specialized resources and due to their complexity, may suffer from poor reliability. The difficulties of procuring multipurpose fighter aircraft are instructive in this regard.

3.1.5 Reliance on Standardization

This is a very inexpensive way of achieving coordination, aside from the inherent loss of flexibility. Use of finely tuned standard operating procedures is a remedy, but can mean substantial planning costs.

3.1.6 Shared Outlook

The simplest two-person system involves individuals with common goals, common experience, and a hardened communications link. Thus, they would have highly shared models and the opportunity to keep them consistent. Having a colleague can reduce some of the difficulties experienced by individuals. For example, information overload can be reduced by dividing information processing responsibilities, and some mistakes can be avoided by having someone to check one's work. But having someone who thinks similarly in the system may just mean having two people prone to the same judgmental difficulties. It might even make matters worse if they draw confidence from a convergence of similarly flawed judgmental processes. One complication is that frequency of interaction can create a perception of completely shared models when sharing is inevitably incomplete.

As organizational size increases, the possibility of completely shared experience decreases. Maximum homogeneity might be found in a hierarchical organization whose leaders have progressed through the ranks from the very bottom, so that they have a deep understanding of the reality of their subordinates' worlds. Thus they will be able to imagine what the subordinates will be thinking and how they might respond in particular circumstances. In such situations, less needs to be said and more can be predicted, making the organization more intimate than it seems.

The down side is that shared misunderstandings also become more likely, and such problems are more difficult to treat because they are broadly entrenched and the organizational climate is likely to be very rough for those who think differently. Indeed if there are any common biases in communications between individuals, then the cumulative bias may be well out of hand by the time communications have cascaded up or down the organizational chart.

Arguably, the heterogeneity of an organization's selection and retention policies may be a good indicator of its resilience within a complex and changing reality. The operators of surface ship combat systems form largely homogeneous battle teams whose strengths include the ability to use individuals and resources interchangeably, existence of a shared organizational culture, use of relatively simple organizational models, and relative ability to interpret one another's actions. Homogeneity of perspectives and skills also entails a degree of vulnerability to *intellectual common mode failures.*

3.2 PRINCIPLES OF BATTLE ORGANIZATION

3.2.1 Overall Command

The following are combat system architecture principles dealing with all levels of command within the combat system:

- It is essential to support generation and monitoring of orders, plans, other tactical information, and control data throughout the combat system as necessary to accomplish tactical command and control functions. In particular, this includes integrated Combat Information Center operations supporting overall system monitoring and control.
- The unit commander and each warfare area coordinator should be free to absorb information, make decisions, and coordinate actions. Information and readiness coordinators are needed to relieve these positions from details of data integration and readiness monitoring and to provide command with a comprehensive summary of the tactical situation.

3.2.2 Unit Command

The combat system must support the commanding officer (CO) and tactical action officer (TAO) in execution of their responsibilities in all warfare mission areas and in all modes of control, ranging from centralized to decentralized. This entails providing for the following capabilities:

- Compliance with force level decisions affecting ship movements and operation.
- Identifying current state and readiness of the combat system.
- Altering combat system configuration and state, to include:
  - Delegating command authority to a lower echelon
  - Assigning multipurpose sensors and weapons to warfare areas
  - Conducting onboard test and training activities
- Resolution of competing demands between individual warfare areas for use of ship's sensors, weapons, and communications.
- Execution of force warfare area commander functions in one or more of the antiair (AAW), antisubmarine (ASW), antisurface (ASuW), and strike (STW) warfare mission areas.
- Exercise broad control over the warfare areas, to include:
  - Directing and coordinating activities (transmission of orders)
  - Command override of lower level decisions
  - Monitoring the tactical situation
  - Establishing tactical plans
  - Establishing battle doctrine, including rules of engagement

3.2.3 Warfare Mission Area Coordinator

The combat system must support the Warfare Mission Area Coordinators in execution of their responsibilities and in all modes of control. This combat system engineering principle entails providing for the following capabilities for the Warfare Mission Area Coordinators:

- Control and direction of engagements within the assigned warfare area, as necessary to accomplish the ship's mission.
- Interface with command to conform to established rules of engagement.
- Exercise control of assigned resources according to command policy.
- Communicate to command the status and planned employment of assigned resources, providing summary information as needed to maintain a comprehensive view of the tactical situation.
- Request from command, as appropriate, control over sensors and weapon systems not currently assigned.

3.2.4 Element Level

The combat system will include sensing, control, engagement, and support elements necessary to conduct of warfare area operations. The following are combat system architecture principles dealing with these elements of the combat system:

- Sensor elements will monitor the environment to establish and maintain a comprehensive tactical picture.
- Control elements will assist with tactical planning, with interpretation of the tactical picture, and with engagement direction.
- Weapon or engagement elements will provide the capability of either neutralizing or destroying designated targets.
- Support elements provide essential communications, data transmission, readiness assessment, training, and electrical power control.

3.3 PRINCIPLES OF CONTROL

3.3.1 Combat System Resources and Activities

The following are combat system architecture principles dealing with the combat system's resources and activities:

- A combat system should exhibit flexibility of use.
- Allow warfighting activities to use either pooled resources, individual resources, or both.
- Allow any profile of resource use by any warfighting activity.
- Allow Warfare Area Coordinators to obligate, to consume, or to generate resources.
- Allow any warfighting activity to alter attributes of resources and/or other warfighting activities.
- Provide for any initial availability profile in either resources or in resource attributes.
- Accommodate earliest start and latest finish windows on all warfighting activities.
- Accommodate warfighting activities of variable duration.
- Accommodate alternative activities.
- Accommodate flexible intervals for resource usage within an activity.

- Accommodate interruptible activities.
- Accommodate both pooled and individual resources. Communication services, global databases and excess capacity in terminals, backup processors, and input/output devices are resources typically shared by multiple nodes.
- Provide directory services to keep track of users, software modules, and data.
- Accommodate organic training activities.

3.3.2 Combat System Relationships

The following are combat system architecture principles dealing with the relationships within the combat system:

- Allow relationships between operational sequences and warfighting activities, or between sequences and sequences.
- Recognize redundant timing relationships.
- Accommodate any number of temporal relationships.
- Accommodate any algebraic relationships between start and end times for a warfighting activity.
- Accommodate selection of any feasible resources, but enforcing that they be the same for any prescribed set of activities.
- Accommodate selection of any feasible resources, but enforcing that they be different for any prescribed set of activities.
- Provide for alternative paths through the system to achieve greater processing, control, interfacing, and storage growth potential than single path designs.
- Organizational unit functions and data formats should be standard.
- Each unit of the battle organization should supply its immediate supervisor with status information for the unit and all subordinates.
- Provide for interprocessor communications.
- Carefully consider exactly who will use the system as well as required functionality. Potential for reducing the number of operators, and for identifying the need for a remote control station, can otherwise be missed.

3.3.3 Algorithms

The following are combat system architecture principles dealing with the combat system's computer algorithms:

- Accommodate any initial conditions; e.g., preobligated resources.
- Accommodate any priority scheme among warfighting activities.
- Accommodate enforcement and/or reporting on constraint violations in any combination of manual decision making and algorithmic decision making using the same constraint checking logic.
- Enforce all temporal and resource relationships.
- Provide scheduling of tasks involving interprocessor interaction.

3.3.4 User Interface and Control

The following are combat system architecture principles dealing with the human interface and control with the combat system:

- Allow assessment of environmental conditions with respect to own unit weapon performance.
- Allow editing of activities and resources, temporal and resource relationships, partial schedules, and availability profiles.
- Allow users to join or merge schedules.
- Allow users to manage display content and layout dynamically as part of the interactive work session.
- The combat system command and control elements should be of modular type and concentrate on collection, reduction, and presentation of information. Reliability is of utmost importance.
- The organization should provide an unambiguous division of responsibility.
- The organization should be able to resolve resource allocation and ship maneuver conflicts in a timely manner at the lowest level.
- Parallel control paths should be provided, permitting simultaneous and/or independent action in each warfare mission area.
- Structure the system to minimize the number of interfaces needed between parts of the system.

- Provide guidelines for man/machine functional interaction to ensure consistency across subsystems. In particular, establish the following guidelines: (a) a positive response to console actions is required; and (b) current system state, mode, and available action options should always be made clear to operators.
- Provide for exchange of information about control actions (bearing on resources, constraints and objectives that are shared concerns) to ensure consistency of actions taken by different controllers.
- Allow conditions of operation to be established or changed for any activity, as directed.
- Allow storage and retrieval of multiple alternative working schedules.
- Facilitate comparison of schedules by user personnel.
- Facilitate comparison of schedules by control elements.
- Facilitate determination of overall state of readiness for any warfighting activity.
- Compute alternative figures of merit for one or more schedules.

3.3.5 Links to Decision and Action Nodes

The following are combat system architecture principles dealing with communication links to decision and action nodes within the combat system:

- Provide for appropriate links to decision and action nodes, including interfaces, displays, and means for transmission of orders.
- Seek *simplest* interface and/or set of interfaces.
- Interfaces should be standardized to the lowest level practical to reduce the cost of integration.
- Implement all meaningful sensor/weapon combinations, reserving judgments on utility to users.
- Avoid bottlenecks in people, computers, and/or interfaces.
- Provide for control information flows such that the active connections between each commander and the weapons under his control form direct and well-defined paths without internal cycles or loops.
- Build the system to support the battle organization.
- Provide for effective interaction with higher echelons of command.

- Minimize potential for conflict in the allocation of sensors and engagement assets.
- Provide the shortest and most efficient paths possible between sensors and weapons (least number of elements and actions).

4.0 READINESS (PHYSICAL PLANT)

The combat system includes a resource management information and control database. Entries or queries to and from this database enable command to monitor and manage the entire plant by tailoring the combat system configuration to prevailing conditions or setting resource priorities. This means monitoring the status of each subsystem and workstation in the plant: what operating tasks are in progress, what operational sequences are being executed, what step in the sequence, how long in that step, what contact is being worked, and so on. One section of this data base will identify each currently active warfighting path, its readiness and current assignment, what actions have been completed with it, quality control information, etc. A number of readiness monitors should be provided at each level of the combat system to extract and report the information necessary for readiness coordination.

4.1 RELIABILITY

The following are combat system architecture principles dealing with combat system reliability:

- Each functional component of the combat system should be protected against malfunctions in other components.
- Distribute sensor and weapon functions, including processing and control, so the action elements are capable of autonomous operation or switchover to other equipment should warfare control equipment fail.
- Employ parallel control systems, providing alternative paths through the combat system so that the loss of one processing or control path is not catastrophic. N+1 redundancy for these paths permits the institution of backup without having to reconstitute the whole inner part of the system.
- Employ a distributed database structure so that restoration of *integrated* storage can be accomplished from the data sources by at least one method in each case.
- Establish bypasses, such as direct sensor/weapon interconnection, to provide at least reduced capability if a critical element fails.
- Provide monitoring and detection of anomalous conditions for any operating element and/or warfighting activity (including ordnance).

- Provide for fault and reasonableness checks on all processes and interfaces (plus dedication of the Resource Monitor to fault detection and isolation) to quickly detect and prevent the spread of failures.

- Provide for offline, separate, preventive maintenance testing at both the combat system and subsystem levels. (The combat system level is built on the separate subsystem level capabilities.)

- Design the capability into the combat system to retrieve information for testing and diagnosis of computer programs. Whether implemented in hardware and/or software, this capability should function as part of the tactical system. As such, the retrieval system function will not change tactical operational characteristics whether in use or not.

- Provide degraded mode options to maintain continuous operating capabilities despite the loss of information system components. This should include the capability to monitor the status of the equipment and the information sources essential to warfare area operations.

- Allow a controller to perform basic coordination functions in the absence of an a priori database. (Otherwise errors or component casualties in prior information systems could lead to a control failure.)

- Provide fault handling and system recovery capabilities.

4.2 MAINTAINABILITY

The following are combat system architecture principles dealing with combat system maintainability:

- Distribute functions into physical modules that can be isolated from the rest of the system for repair while the remainder of the system continues to function.

- Provide for online checks of equipment and interfaces by separate system level test equipment, with results continuously reported at the system level to achieve quick fault detection and isolation to the separable modules.

- Creation of unnecessary differences between parts of a combat system structure or across different ship classes should be avoided.

- Modules should be designed for simplicity, which is reflected not only the number of elements employed but also in ease of learning and ease of use.

- Provide sufficient flexibility (via system reconfiguration, embedded maintenance programs, and/or new technology) to minimize combat system downtime for maintenance.

- Establish predictable system performance characteristics so that the abnormal can be quickly and easily distinguished from the normal.

- Provide for a relatively simple, orderly structure, making it easier to trace failures and implement module isolation for repair.

- The combat system should allow for control and conduct of training, testing, and fault detection without degrading operational capability. In many existing combat systems test, diagnosis, and repair functions are optimized by weapon system; combat system level functions are nonexistent. Disparities in weapon system approaches further aggravate the situation. Combat system level requirements need to be defined to increase weapon system commonality and provide the basis for orderly implementation of the system level functions.

4.3 SURVIVABILITY

The following are combat system architecture principles dealing with combat system survivability:

- Provide for graceful system degradation and the ability to fight hurt despite the loss of key subsystems and components.

- Excessive dependence on critical components tends to produce catastrophic failure modes with limited potential for backups and casualty mode operations. Distributed systems, with increased element autonomy, would permit fewer critical points of failure and more recovery options.

- The warfare area coordinator units should be able to operate in a standalone mode in the event of loss of the command unit.

- Sensor units and weapon units should be able to operate independent of other elements regardless of allocation.

- Redundant or backup capabilities should be provided for critical functions to allow fault recovery.

- Allow for ship arrangement flexibility to support combat system reconfiguration minimizing the effects of battle damage.

- Command and warfare areas should be capable of performing the critical functions of other warfare areas and/or command for backup and casualty mode operations.

- The combat system should exhibit minimum single points of failure.

- Distribute functions and the database and couple these measures with geographic separation to reduce the amount of capability lost should a given area sustain damage.

- Provide for alternative data and control paths through the system and couple these measures with geographic separation to reduce the amount of capability lost should a given area sustain damage.

- Provide for system level monitoring, backup features, and repair capabilities to enhance survivability as well as system reliability and maintainability.

- Data or information transferred should be broadcast to provide the most coordination redundancy and survivability for the bandwidth expended.

5.0 REFERENCES

B-1. DeLong, R. Peter, "Ten Principles of Command and Control System Automation," *Naval Engineers Journal*, January 1990.

B-2. Duren, B.G., *Integrating Uncertain Information in Situation Assessment Tasks*, NAVSWC TN 90-323, NAVSWC, Dahlgren, VA, August 1990.

B-3. Lindemann, M.J.; Grubbs, J.L., Jr.; Eason, W.J.; Lehmann, L.L.; Shields, E.F.; and Evers, T.H., MODFCS Architecture Baseline: A Combat System Architecture. NAVSWC TR 79-357, NAVSWC, Dahlgren, VA, January 1980.

B-4. Hornstein, R.S., and Willoughby, J.K., "A System Management Methodology for Building Successful Resource Management Systems," *AIAA Computers in Aerospace VII Conference*, October 1989.

B-5. Zachman, J.A., "A Framework for Information Systems Architecture," *IBM Systems Journal* 26(3): 276-292, 1987.

B-6. Scarrott, G.G., "The Nature of Information," *Comp. J.* 32(3): 262-266, 1989.

DISTRIBUTION

		COPIES
ATTN	OP 03	1
	OP 03C	1
	OP 05	1
	OP 05W	1
	OP 06	1
	OP 07	1
	OP 08	1
	OP 091	1
	OP 094	1
	OP 30	1
	OP 31	1
	OP 32	1
	OP 321	1
	OP 322	1
	OP 324	1
	OP 33	1
	OP 37	1
	OP 39	1
	OP 551	1
	OP 603	1
	OP 604	1
	OP 605A	1
	OP 70	1
	OP 71	1
	OP 72	1
	OP 74	1
	OP 75	1
	OP 76	1
	OP 801L	1

CHIEF OF NAVAL OPERATIONS
DEPARTMENT OF THE NAVY
WASHINGTON DC 20350-5000

ATTN	DIRECTOR SURFACE WARFARE	1
	DIRECTOR AIR WARFARE	1
	DIRECTOR SUBMARINE WARFARE	1

ASSISTANT SECRETARY OF THE NAVY (RDA)
DEPARTMENT OF THE NAVY
WASHINGTON DC 20350-5101

		COPIES
ATTN	SPAWAR 00	1
	SPAWAR 09	1
	SPAWAR 005	1
	SPAWAR 10	1
	SPAWAR 30	1
	SPAWAR 30T	1
	SPAWAR 31	1
	SPAWAR 31F	1
	SPAWAR 314	1
	SPAWAR 315	1
	SPAWAR 316	1
	SPAWAR 317	1
	SPAWAR 32	1
	PD 40	1
	PD 50	1
	PD 60	1

COMMANDER
SPACE AND NAVAL WARFARE
SYSTEMS COMMAND
WASHINGTON DC 20363-5100

ATTN	OFFICE OF THE COMMANDANT (CMC)	1

HEADQUARTERS
US MARINE CORPS
WASHINGTON DC 20380-0001

ATTN	ONR 100	1

OFFICE OF THE CHIEF OF
NAVAL RESEARCH
BALLSTON TOWERS
821 QUINCY STREET
ARLINGTON VA 22217-5000

COMMANDER CARDEROCK DIVISION NAVAL SURFACE WARFARE CENTER BETHESDA MD 20084-5000	1

DISTRIBUTION (CONTINUED)

		COPIES
ATTN	SEA 00	1
	SEA 05	1
	SEA 06	1
	SEA 06A	1
	SEA 06K	1
	SEA 06D	1
	SEA 06U	1
	SEA 50	1
	SEA 61	1
	SEA 91	1
	SEA 93	1
	SEA 300	1
	SEA 312	1
	SEA 400	1
	SEA 411	1
	SEA 441	1
	SEA 06C	1
	SEA 06	1

NAVAL SEA SYSTEMS COMMAND
WASHINGTON DC 20362-5101

COMMANDING OFFICER
COASTAL SYSTEMS STATION
DAHLGREN DIVISION
NAVAL SURFACE WARFARE CENTER
PANAMA CITY FL 32407-5000 — 1

COMMANDING OFFICER
NAVAL AIR DEVELOPMENT CENTER
WARMINSTER PA 18974 — 1

COMMANDER
NAVAL UNDERSEA WARFARE
CENTER DIVISION
NEWPORT RI 02841-5047 — 1

		COPIES
ATTN	R LOBB	1
	P DEPOY	1

CENTER FOR NAVAL ANALYSES
4401 FORD AVENUE
ALEXANDRIA VA 22314

DEFENSE TECHNICAL INFORMATION
CENTER
CAMERON STATION
ALEXANDRIA VA 22314 — 1

INTERNAL DISTRIBUTION:

		COPIES
C		1
D		1
D2		1
D25		1
E		1
E261	FINK	1
E231		3
E232		2
E32	GIDEP	1
F		1
FOX		1
F10		1
F20		1
F30		1
F40		1
G		1
G10		1
G20		1
G30		1
G40		1
G60		1
G70		1
K		1
K10		1
K50		1
N		1
N02		1
N04		20
N05		1

DISTRIBUTION (CONTINUED)

		COPIES
N06		1
N10		1
N20		1
N30		1
N35	CULLEN	1
N40		1
U		1
U04		1
U07		1
U10		1
U20		1
U30		1
U40		1

REPORT DOCUMENTATION PAGE

Form Approved
OBM No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 1991	**3. REPORT TYPE AND DATES COVERED**

4. TITLE AND SUBTITLE Combat Systems Vision 2030 Combat System Architecture: Design Principles and Methodology	**5. FUNDING NUMBERS**
6. AUTHOR(S) Bernard G. Duren and James R. Pollard	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Surface Warfare Center (N04) Dahlgren, Virginia 22448-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER** NAVSWC TR 91-795
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(Maximum 200 words)*

In this report, combat system architecture is defined, and a reference model is formulated for the operating tasks performed by combat systems. Factors driving the influence of architecture design on the military worth of surface combatants are discussed, and an associated set of design principles is provided. These are presented in the form of a building code or general design rules. The rule base is then used to trace key decisions in development of the 'H' or Vision Architecture concept. Finally, a parallel derivation is given for the Vision Architecture that is based on methods for synthesis of decentralized control structures.

14. SUBJECT TERMS Combat Systems Vision 2030, Combat System Architecture, Vision Architecture	**15. NUMBER OF PAGES** 79
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	**18. SECURITY CLASSIFICATION OF THIS PAGE** UNCLASSIFIED	**19. SECURITY CLASSIFICATION OF ABSTRACT** UNCLASSIFIED	**20. LIMITATION OF ABSTRACT** SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102