THE COMMAND AND CONTROL

REFERENCE MODEL

THE COMMAND AND CONTROL REFERENCE MODEL

FOR

MODELING, SIMULATIONS, AND TECHNOLOGY APPLICATIONS

NOTE: This draft is not an official product and is subject to modifications.

It is provided for information and coordination only.

DISTRIBUTION STATEMENT
Approved for public release; distribution is unlimited.

Prepared for
Joint Directors of Laboratories
Technical Panel for C3
C3 Research and Technology Program
Basic Research Group
and for
Defense Modeling and Simulation Office

Prepared by
U.S.Army Communications-Electronics Command
Research, Development, and Engineering Center
Command, Control, and Systems Integration Directorate
C2 Technology Division
Fort Monmouth, NJ 07703-5603

NOTICES

Disclaimers

The information in this document is not to be construed as an official Government position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this document is not to be construed as an official Government endorsement or approval of commercial products or services referenced herein.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	20 January 1994	Technical Report: Coordination Draft

4. TITLE AND SUBTITLE

THE COMMAND AND CONTROL REFERENCE MODEL
FOR MODELING, SIMULATIONS,
AND TECHNOLOGY APPLICATIONS

5. FUNDING NUMBERS

6. AUTHOR(S)

Israel Mayk, Eng.Sc.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S.Army Communications-Electronics Command (CECOM)
Research, Development and Engineering Center (RDEC)
Command, Control and Systems Integration Directorate (C2SID)
ATTN: AMSEL-RD-C2-TC, Fort Monmouth, NJ 07703-5603

8. PERFORMING ORGANIZATION REPORT NUMBER

CECOM-TR-94-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Basic Research Group (BRG), Technical Panel for C^3 (TPC3)
Joint Directors of Laboratories (JDL)
NRaD, San Diego, CA 92152; RADC, Rome, NY 13441;
CECOM, Fort Monmouth, NJ 07703

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

JDL TPC3 BRG
TR-94-1

11. SUPPLEMENTARY NOTES

This report marks the seventh release of the Command and Control Reference Model (C2RM).

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The C2RM provides a framework for the evolution of a coordinated and detailed definition of a command and control (C2) discipline. The C2RM embodies an integrated multidisciplinary approach. It is intended to be complete and self-consistent for the main levels of abstractions encountered in models, simulations, operational applications, functional descriptions, paradigms and metaphors of C2. The scope of the C2RM embraces C2 using all key physical and logical interactions associated with C2 systems. It is concerned with interactions, involving not only communications (e.g., radios), but transportations (e.g., vehicles), identifications (e.g., sensors), and inflictions (e.g., weapons), which take place between resources of the same, friendly, hostile or neutral C2 units. High levels of abstractions of user requirements for C2 across the broad spectrum of military and civil domains have led to the development of the C2RM. It applies to all phases of system acquisition from the laboratory to the field and from conceptualization to realization. The C2RM is based upon generic and analog extensions to the International Standards Organization (ISO) open system interconnection (OSI) reference model (RM) which go far beyond the scope of the ISO OSI RM. The major theme, however, of layering services is preserved to facilitate understanding, reuse of design, implementation, and interoperability to the maximum degree possible with available C2 technology.

14. SUBJECT TERMS

Command and Control, Open Systems, Object-Oriented, Simulations, Decision-Support, Military Science, Interoperability, Reference Model

15. NUMBER OF PAGES

160

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

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

At the end of all our exploring will be to arrive where we started
and know the place for the first time.

T. S. Eliot, a poet

To arrive at an abstraction it is always necessary to begin with a concrete reality.

Pablo Picasso, an artist

Imagination is more important than knowledge.

Albert Einstein, a scientist

C^2 systems are the only systems that matter. Defense is command and control; everything else is a detail.

Paul Strassmann, a manager

CONTENTS

Section **Page**

Section **Page**

Section **Page**

FIGURES

Figure **Page**

Figure Page

TABLES

ABSTRACT

The Command and Control (C^2) Reference Model (RM) embraces, in an integrated fashion, analogous architectures for all key identification-, infliction-, communication- or transportation-oriented interactions subject to C^2. The C^2RM thus provides a multidimensional infrastructure for generic C^2 application and implementation entities. Applications provide the semantic shell and implementations provide the syntactic shell for C^2. Applications are layered to span services which utilize conflict-, presentation-, operation-, procedure-, network-, link-, or asset-oriented facilities. Implementations are layered to span services which utilize experience-, knowledge-, information-, object-, tool-, equipment- and supply-oriented facilities. As such the C^2RM provides a common structure for C^2 systems and underlying services which follows the International Standards Organization (ISO) Open Systems Interconnection (OSI)/Open Distributed Processing (ODP) RMs to the maximum degree practicable. The C^2RM may be used as a framework to define one's own or adversary force structures and viewpoints at multiple echelons. C^2RM entities apply to human observation-decision-action-making capabilities as well as to augmentations thereof with machine observation-decision-action-aiding capabilities.

A Note to the Reader/Reviewer
As you browse, read, or analyze this document, you will see many Keywords which look familiar. Hopefully your familiarity with a Keywords should be helpful in understanding its associated notion. If not, please refer to the Glossary at Annex B for all Keywords. In theory all Keywords should be capitalized where their formal meaning is intended. **Keywords** which appear at the beginning of a sentence should be capitalized in a bold style as indicated by the first word of this sentence. In many places, however, this formality is sacrificed in favor of readability. Also note that references enclosed in double brackets, [[n]], are an integral part of the C2RM and therefore appear in Annex A as Applicable Documents. References enclosed in single brackets [n], serve as background and relevant information.

FOREWORD

Main Idea
The C^2RM describes a generic framework for an object-oriented, open system architecture of resources which are networked and integrated to comprise a C^2 system. The idea of establishing a reference model for such a framework to coordinate standards or technical specifications is not new. The purpose of a reference model for any area of pursuit is to bring consensus and a sense of common purpose to a fragmented or distributed community of researchers, developers and users. Key to the success of any such model is its underlying principles and philosophy which must be well founded. Only models which evolve through an arduous, open and well-documented process can obtain credibility and become accepted for guiding research, development and operations of related applications. This lesson alone is highly controversial since the development of a reference model can be a long, tedious and costly process. Many organizations who can benefit do not want to become involved since they are faced with overriding near-term objectives. Nevertheless, the need for a reference model as an integral part of long-term planning cannot be ignored when much is at stake if investments in short-term projects cannot be leveraged, integrated, reused or protected as large scale systems evolve.

Accreditation
In the United States of America, the American National Standards Institute (ANSI) is chartered to accredit standards development groups. It is the responsibility of a particular community of interest such as professional organizations to organize and charter committees for the purpose of establishing new standards in the area of their expertise. ANSI, for example, has accredited the Technical Committee (TC) of the International Standards Organization (ISO) ISO/TC 97, Information Processing Systems to develop the ISO Open System Interconnection (OSI) Reference Model (RM) [[1]] to facilitate the evolutionary development of compatible telecommunications interfaces and protocols. ANSI also accredited the Institute of Electrical and Electronic Engineers (IEEE) Computer Society Technical Committee on Operating Systems (TCOS) to develop a set of interface standards known as POSIX to enhance portability of Unix-based applications across competing vendors which support Unix on their hardware.

Endorsement
Not all standards have evolved through the formal process of ISO and IEEE. Many industry de facto standards as well as government standards have evolved around successful single vendor products which captured a large share of the market place. Unfortunately, without an overall framework or a reference model, the demand to protect consumer or user investment in such products has led to entrenchment, diversification and proliferation of many competing incompatible standards. Recently however, many such vendors have realized that they have more to lose in the absence of standards than in subscribing to a common set of standards. This has led to the formation of highly powerful consortia such as the OSI/Network Management (NM) Forum and the Object Management Group (OMG) chartered by their member organizations to develop a set of standards and specifications in their respective areas. The OSI/NM Forum is an international consortium of nearly 100 members that includes most of the world's major communications systems providers. It was established in 1988, to accelerate the development of an object-oriented architecture and protocol specifications for the management of communications networks [3]. OSI/NM Forum has derived much benefit from the ISO OSI RM for its underpinnings. Similarly, the Object Management Group (OMG) boasts of over

300 members including many of the world's leading computer systems vendors. It was established in 1989 to promote interoperability between applications on different machines in heterogeneous distributed environments through seamless interconnection of multiple object systems. The first product of the OMG was a reference model for object-oriented technology [2]. The OMG RM is expected to play a role in the evolution of object-oriented technology similar to the role established for the ISO/OSI RM.

Meeting the Challenge

Most efforts in C^2 proceed directly from specific "end-user" requirements to specific technical requirements. Much of the general insight gained from these efforts is simply not captured for future reuse. Similarly, most research efforts proceed narrowly from abstract object models to concrete object models and are generally confined to only one given discipline. As in related technology areas, efforts in C^2 may be greatly enhanced and accelerated through the utilization of a common framework for the C^2 discipline, i.e., via a reference model. A reference model which is generic and amenable to benefit from existing and ongoing standardization efforts in general and specifically from those which are mentioned above will be most beneficial. It should contribute greatly to the success of many C^2 related efforts which require a high level of collaboration across multiple disciplines. The C^2RM, however, must go beyond pulling together related and support technologies into a common framework. In addition and as well, it must pull together many of its inherent and unique perspectives in a consistent, complete and clear presentation and in support of diverse, multidimensional, multidisciplinary C^2 applications. Facing this challenge, members of the Basic Research Group (BRG) of the Joint Directors of Laboratories (JDL), Technical Panel for C^3 (TPC^3) through the C^3 Research and Technology (C^3R&T) Program [5] are chartered to sponsor the development of the C^2RM described herein to foster greater collaboration among C^2 scientists, engineers and operational users and to encourage the establishment of C^2 as a discipline, endowed with a coherent set of definitions, principles and theories validated by scientific experiments and laboratory demonstrations. This requires the C^2 scientists and C^2 engineers to focus C^2 efforts and collaborate not only within their own respective C^2 community but with the C^2 operational user, i.e., the commanders, their staff and the forces which they command and control. Many aspects of C^2 are often hidden in history and doctrine, evolving organizational structures and associated technology products. In effect, C^2 is ubiquitous and may be found individually in any resource involved in a conflict and collectively in all resources teamed as a coherent force. It is the diversity of perspectives on C^2 which beckons a common reference model and associated terminology to facilitate the evolution toward a consolidated framework for understanding C^2.

Status

This preliminary draft is intended to provide an up-to-date version of what will hopefully become a universal model or framework for the evolution of a coordinated and detailed definition of a command and control (C^2) discipline. Thus far, the C^2RM is a result of an ad hoc collective effort to arrive at a common language for pursuing research and development of C^2 systems. High levels of abstractions of user requirements for C^2 across the broad spectrum of military and civil domains have led to the development of the C^2RM. The C^2RM is in the process of being coordinated by DoD and developed for C^2 applications and interoperability. Such a framework is expected to promote, facilitate and expedite coherent progress and greater collaboration in research and development, modeling and simulation, test and evaluation, experimentation and demonstrations, analysis and design, as well as education and training of C^2 systems. The primitive notions, terms and definitions and the entire structure of the C^2RM adopted in this draft are subject to change, pending future updates as required to achieve a greater common understanding. Changes and updates are being coordinated by the C^2RM Subgroup of the TPC^3 BRG with the support of the C^2 community-at-large. It is a working document intended to give interested parties in government, industry and academia an early insight into the status and direction of the BRG. The BRG recognizes that the current document in its current state may be inconsistent in presentation, incomplete from various perspectives, and may very well lack clarity in places. The BRG, therefore, solicits and encourages comments and feedback for future iterations.

Plan

In the near term, the C^2RM will continue to evolve in an ad hoc manner. Proposals are underway to charter a more formal working group to be funded by member organizations. Members of the C^2RM Subgroup will then meet on a regular basis.

EXECUTIVE SUMMARY

A Common Open Framework

The C^2RM embodies an integrated multidisciplinary approach to characterize intelligent C^2 systems and the way in which they cope with uncertainty. It is intended to be complete and self-consistent for the highest levels of abstractions which are encountered often in various other models, simulations, functional descriptions, paradigms and metaphors of C^2. C^2 systems come comple [illegible] h resources capable of initiating and maintaining physical interactions in an environment wi [illegible] er capable, friendly, neutral, or adversary resources, each performing its share towards attaining perceived C^2 system effectiveness requirements. It provides a broad but non-exhaustive general description of a C^2 Reference Model (C^2RM) for a common understanding of C^2 systems and an insight into their design and behavior. Services of a given layer are relative to the requirements of the layer above. Fully compliant resources are said to be "open" in the sense that interoperability may be enhanced by adding more resources to the system or in the sense that entities of existing resources may be enhanced or upgraded to satisfy certain shortfalls without a penalty of incompatibility or the need to re-engineer an entire subsystem.

Extension of ISO OSI RM

The C^2RM includes the ISO OSI RM by adopting it for the communications types of interactions. In parallel, layers of three other complementary types of interactions also provide services to the application layer. The C^2RM embraces analogous architectures for all the key types of physical interactions and utilizes the application layer to provide command and control over all types of interactions in an integrated fashion. The notions defined herein are established as part of the C^2RM and technically should not be confused with the notions conveyed by the same words which may be found in other documents of other disciplines or as a part of lay language. The C^2RM as described herein is by no means complete. The ISO OSI RM and related disciplines, however, serve as guidance to further development of some of the key concepts inherent thus far in the C^2RM. Much like the ISO OSI RM, it is the goal of C^2RM to provide the framework of choice to guide the development of a consistent set of standards and specifications for interoperability and to offer substantial protection of extensive investments in acquisitions by promoting modular reusable technologies. The advantage of this model is that it has the flexibility to incorporate many features of existing paradigms and to accommodate a wide variety of perspectives while promoting a greater common understanding of the levels of interoperability required among C^2 system components. With the exception of the ISO OSI communications services, the C^2RM is being developed independently of the ISO OSI RM to achieve similar advantages for C^2.

Two-Dimensional Layered Entities

The C^2RM provides a seven-layer structure for generic C^2 problem-solving application entities spanning across humans and machines. Orthogonally, seven layers of generic technology implementations span across persons and equipments which correspond to each application entity. Thus, much like human behavior is describable generically, i.e., independently of the person implementing (realizing) the human, machine behavior is also describable generically, i.e., independently of the technology underlying its implementation. At such a level of abstraction, but from another perspective, applications are also said to provide the semantics, where-as implementations are said to provide the syntax for C^2. Just as a language requires both semantics and syntax, C^2 requires applications and implementations. An open system architecture for implementation is equivalent to providing a context-free grammar for a language. As such, the C²RM provides a universal structure for C² and underlying services which follow ISO OSI to the maximum degree practicable. Clearly, services which embed human decision-aiding or decision-making will require extensive R&D and a high level maturity of understanding before any agreement may be reached for the purpose of standardization.

Application Sublayers. Applications are layered according to the level of conflict, presentation, operation, procedure, network, link, and asset services. Layered application services provide generically productized hierarchical missions, plans, tasks, jobs, assignments, transactions and packages. The C²RM identifies generic officials who are responsible hierarchically for the execution of a class of methods unique to each layer. Thus, generic commanders, planners, controllers, agents, administrators, coordinators, and operators are responsible for the analysis and synthesis of policies, strategies, tactics, schemata, disciplines, techniques, and instructions, respectively. Note that application sublayers define the problem/solution domain (psd).

Conflict Sublayers. The generic conflict sublayer of the application layer is also layered sevenfold in a nested fashion to address missions and leadership. Generic missions are generated by the layers of conflict for generic peace, war, campaign, battle, combat, engagement, and armament. The C²RM identifies generic commanders as official leaders responsible for each class of missions. Missions for each sublayer of conflict are generated through the initiative, motivation, and will exhibited by presidents, generals, directors, managers, captains, partners, and experts, respectively. Since many C² systems are nested hierarchically, leadership characteristics will also be distributed in a recursive fashion. Thus, the C²RM may be used as a framework to define one's own force structure and viewpoint at multiple echelons as well as for defining the perception of an adversarial force structure and viewpoints

Implementation Layers. Implementations are layered to provide technological representations for problem/solution domain entities. Implementation/technology domain (itd) products consist of units of judgements (e.g., decisions, approvals), recommendations (e.g., actions, conditions), conclusions (e.g., observations, assessments), bundles (e.g., records, slots, blobs), parcels (e.g., code, symbols), impulses (e.g., signals, sparks) and energy (e.g., electricity, fuel) which result from layered services of experience, knowledge, information, objects, tools, equipment, and supply, respectively.

Relation to Problem Solving

The C²RM is also based upon a general seven layer metamodel for problem solving. Problems are identified and framed at the highest level of abstraction for a given application. Solutions are provided by three subordinate time dependent layers involving future, present and past time frame considerations and three subordinate space dependent layers involving multilateral, bilateral and unilateral capabilities.

Relation to C² Organizations

From a structural perspective, the C²RM is also consistent with a metamodel which calls out the need to define C² organizations in terms of architectural frameworks, system configurations, functional elements, resource capabilities (e.g., layered applications), entity collections (layered services), utilities and facilities which coordinate and process interaction outcomes, and action events. Such models are pervasive in many organizations of military or civilian constituency.

OBJECTIVE

The objective of the C^2RM is to provide a standard, object-oriented, open system architecture, and open system interconnection framework to be used in modeling and simulating C^2ed objects, and in developing technology and applications of managed objects for command and control (i.e., C^2 objects) with standard C^2 protocols and interfaces for interactions among heterogeneous and distributed resources of C^2 systems. Thus it is hoped that the C^2RM will:

a) Provide focus and guidance to any enterprise C^2 efforts
b) Foster the evolution of common
   taxonomy (classification scheme) for C^2
   partonomy (composition scheme) for C^2
   symbology (presentation scheme) for C^2
   protocols (interaction scheme) for C^2
c) Improve interoperability of C^2
d) Improve SW reuse for C^2
e) Accelerate the evolution of C^2 systems
f) Facilitate C^2 technology insertion
g) Leverage industry's independent R&D in C^2
h) Promote competition in C^2 and yet reduce duplication of efforts

Implementation flexibility is retained by allowing for independent implementations as long as the services are provided and use a coordinated set of interaction protocols. The R&D and standardization of application program interfaces (APIs) is critical to the success of the above objectives. Acquisition of quality and cost-effective design and implementation is highly dependent upon the establishment of formal interfaces between modules developed by different people on a project whether they come from a single organization or are trying to collaborate across organizational boundaries. A system is said to be open to the degree to which it: a) embraces well-defined, non-proprietary interfaces and protocols, b) makes intelligent use of well-established commercial standards, c) supports export and import of capabilities in multiple formats and multiple media, d) accommodates scalability for improved capacity and performance, e) permits portability across heterogenous platforms, and f) promotes reusability and interchangeability of modules encapsulating competing and proprietary technologies.

SCOPE

Applicability

The C^2RM is applicable to any C^2 system of military or civilian entities. It pertains to any effort of C^2 whether real-time, time-critical, near real-time, or non-real-time This role of the C^2RM is depicted in Figure 1. It applies to all phases of C^2 system acquisition from the laboratory to the field and from conceptualization to realization. It relates to all perspectives and levels of abstraction of C^2 which may be derived from user requirements and applied to technical specifications and products. The C^2RM is concerned with all abstract or concrete C^2ed objects which span both the end-user's C^2 process problem or solution domains and the developer's technology requirements, design or implementation domains.

It does not specify actual services and protocols. Moreover, it is neither an implementation specification for C^2 systems, nor the basis for appraising implementation conformance. It is intended only to provide uniformity of guidance with respect to standards and specifications for services and protocols needed for C^2 systems.

The architectural principles of the C^2RM are applicable in the broadest sense possible to embrace all key physical and logical interactions associated with C^2 systems and their resources. It is relevant to a wide range of physical interactions involving not only communications (e.g., radios), but transportations (e.g., vehicles), identifications (e.g., sensors), and inflictions (e.g., weapons), as defined herein. These interactions may take place between resources of the same, friendly, adversarial or neutral C^2 systems. The C^2RM is concerned equally with the logical (peer-to-peer/client-server) interactions of entities as well as with associated physical interactions.

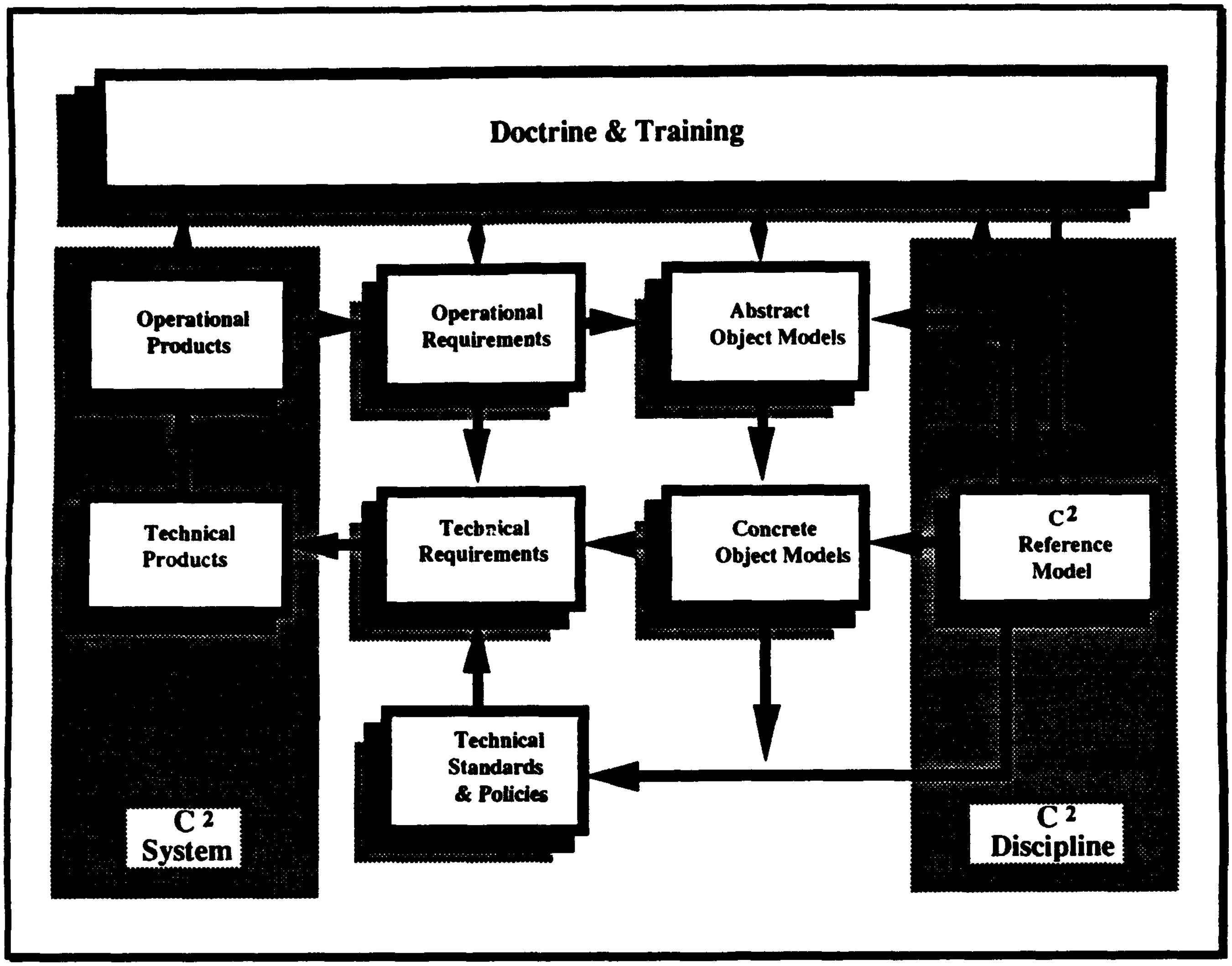

Figure 1. The role of the C^2RM[1]

[1] Adapted and extended from [2]

BACKGROUND

C² Jargon

The C² community has been and continues to generate in an ad hoc fashion many paradigms for understanding C² systems. As a result, observations made by the Defense Science Board study [4] in 1978 are still true in 1992, i.e.,

> ...there is almost no commonly understood vocabulary or conceptual framework for analyzing, designing, or evaluating command and control systems...

The lack of common definitions or a set of definitions which is complete and consistent is well recognized to be a serious obstacle to the evolution of a C² discipline. For example, command is typically associated with exercising "authority" whereas control is associated with exercising "direction" over assigned forces in the accomplishment of missions. The definitions for "Command and Control (C²)" usually imply the combination of command and control in an additive fashion. Definitions of "Command and Control System," however, typically include the facilities, equipment, communications, procedures and personnel essential to the commander for planning and controlling operations of assigned forces pursuant to the assigned missions. Note that in many paradigms of C² [11] the force and the commander are generally excluded from the "C² system," yet both the commander and the C² system are included in the Table of Organization and Equipment (TO&E) of force units. The above definitions are insufficient to resolve the following related issues: Is it true that, as asserted in Reference [6], "commanders are part of C³ systems, not just users of them"? If not, what is the name of the system that includes the commander? What is the name of the integrated system that includes not only the commander but his forces as well? What is the difference between "C²" and "C³" and between "C² System" and "C³ System"? An agreement on these and numerous other terminology issues is essential for progress in C² research and developments.

Informally, the fundamental notions of C² have existed since man began to understand himself and resolved or attempted to resolve potential conflicts which lurked in his path. Formally, however, fragments of these notions evolved in narrow contexts, scattered and imbedded in a variety of distinct but broad disciplines such as management science, behavioral science, operations research, physical sciences, cybernetics, automatic control, communications and computer science, and more specialized disciplines such as artificial intelligence, robotics, distributed processing, and signal processing. A coherent merger of the theories supported by each of these disciplines as they apply to C² systems through common definitions will constitute the theory of C².

Operationally, C² is used to identify which units will be subordinate to a given commander. Different levels of responsibilities will be invoked depending upon the attribute ascribed to a subordinate unit. Thus a Resource may be Organic_ | Assigned_ | Attached_ | Mission_Controlled.

C² Theory

C² theory is evolving as that coherent body of knowledge which provides for an understanding of the effective potential for a C² system to accomplish its mission. C² theory is based upon a formal discipline for describing C² architectures. C² theory facilitates the development of models of C² systems which would allow for the evaluation of the effectiveness of C² capabilities in the context of a wide range of conflicts and scenarios. A C² theory, like a theory in any other well-founded discipline, is based upon a set of key primitive notions and a set of corresponding terms and definitions which serve as the foundation for research and as building blocks for development. Since C^2 systems include human decision-making, however, the theory of C^2 cannot be expected to provide a high degree of confidence with respect to any prediction of performance and effectiveness. Nevertheless, a coherent theory of C^2 may provide a significantly higher level of insight and motivation for understanding of complex but rational man-machine-based systems than otherwise possible. The C^2RM identifies the key primitive notions of C^2, provides a consistent set of associated terms and definitions and incorporates them in an integrated fashion as a basis for describing C^2 architectures and systems. The set of key primitive notions and associated terms established as part of the C^2RM are defined in Annex B and should not be confused with notions conveyed by the same terms as used in other models, disciplines, or in the lay language.

Operationally, a theory of C^2 provides the framework to gain insight into the organizational structure of a C^2 system and its effectiveness.

Two-Body Problems

The use of a specific paradigm in various applications of C^2 often depends on the background and affiliation of the people using it. Successful paradigms must be useful in relating to real architectures. The description of resources of C^2 systems provides a vehicle to consolidate perspectives from different fields into a common architecture. In a very general way, each specializing discipline attempts to describe the dynamic behavior of what is essentially a two-body problem from its own point of view. Two-body interactions are subsequently described in the presence of a network of interacting bodies. Depending upon the discipline, the two bodies assume different names and definitions. Since each discipline is well entrenched in its own jargon, much confusion may be generated when the same or similar applications are addressed by more than one discipline, as may be ascertained from Table 1. In the C^2RM these bodies are meta-objects of resources, sub-resources or entities required to support specific services.

Table 1. Discipline-oriented resources

Discipline	Meta-object A	Meta-object B
Control	Controller	Plant
Communications	Transmitter	Receiver
AI	Planner	Agent
Economics	Consumer	Supplier
Queueing	Customer	Server
Distributed Process	Client	Server
C^2	Commander	Controller

All of the disciplines identified in Table 1, and many others, provide important insight into C^2. Results from related domains of knowledge need to be properly reframed before associated entities may be leveraged and integrated into layered applications of C^2 systems.

Layering Principles

Each one of the above meta-objects may be an intelligent resource and interact with its dual on multiple levels. Entities tightly coupled at a given peer level of interaction are grouped to form a layer. A group of such entities is shown in Figure 2. The layer of the architecture in which a process is embedded, therefore, becomes important since it provides the context within which to formalize definitions in a coherent and complete fashion. As an extreme point of view, any process with inputs and outputs may be said to be controlled by its inputs and provide control to others through its outputs. The importance of layered structures is accentuated for resources of systems subject to competitive multi-vendor acquisition environments. The notion of layering is not new and has been known and used effectively for many years in many disciplines. Each layer or sublayer must provide services to the layer or sublayer directly above. Conversely, the services defined at each layer or sublayer rely upon the services provided by the layer or sublayer which may be directly or indirectly underneath. In particular, the ISO OSI RM was developed using the generic principles similar to those listed in Table 2:

Table 2. General principles of layering resources

a) the number of layers should be manageable
b) the boundaries between layers should be clear with minimum interactions
c) each layer should have unique functions
d) related functions should be grouped into a single layer
e) past experience should be considered
f) functions within a layer should be tightly coupled
g) only layer boundaries should be standardized
h) layers should be distinguished by hierarchical abstractions
i) layers should be encapsulated and highly transparent to other layers
j) a layer boundary should be required only for adjacent layers
k) layers may be layered internally into sublayers
l) layers should have a common interface with adjacent layers
m) sublayers may be degenerate

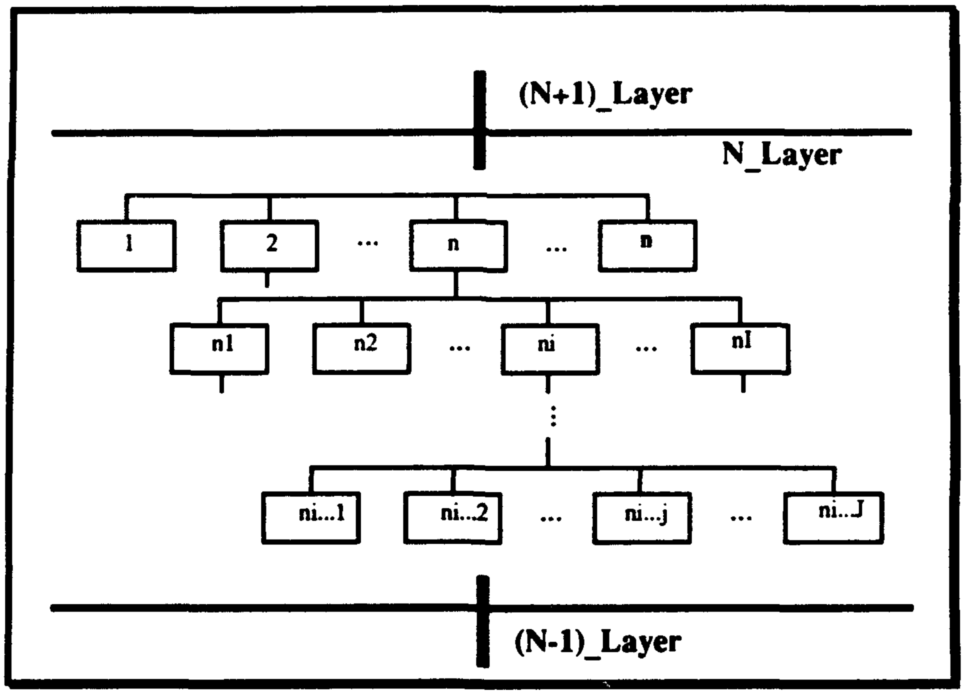

Figure 2. An abstract layer of entities

APPROACH

Object Oriented

The C^2RM is a high-level descriptive model phenomenologically derived through knowledge acquisition and object-oriented analysis of operational environments. Regardless of the stage of technological development, it is presumed herein that a real system, called a C^2 object, which comprises the interacting entities of personnel and materiel, may be represented by embedded object-oriented fundamental building blocks layered to achieve effective command and control. These embedded objects are called C^2 layer objects. C^2 layer objects may be functional or physical. Functional descriptions may be incorporated into physical realizations, and, conversely, physical descriptions may be reducible to functional modules. The features and characteristics of C^2 layer objects, the possible local and global associations within and among the C^2 layer objects, and their means of interactions constitute the C^2 architecture and imply the potentials of all derivative systems. The initial stage of developing the C^2RM is directed towards conceptual explications and elucidation of generic and technology-independent principles supported by object-oriented methodologies. The C^2RM is independent of OOT yet it provides an essential unifying framework for bridging the gaps between OOA and OOD and between OOD and OOP. No matter which OOT methodology is used, persistence of objects and associated taxonomy and partonomy between the domain model and the reference architecture and between the reference architecture and component implementations is essential to maintain traceability and to protect investments in OOT. Conceptual developments will be aided by and stimulate the further development of common formal descriptive techniques. The approach is comprehensive in the sense that it accommodates a wide spectrum of theories which may be based upon the phenomenology of macroscopic behavior as well as basic principles of meta-, meso- and microscopic nature of C^2, i.e., within each C^2 layer object are found C^2ed objects which present and represent the perceived view of the universe within the framework of that layer of the enterprise. Decision-aid objects are responsible for presenting such views, whereas conflict region objects are responsible for representing the perceived view.

System Acquisition

It will be incumbent upon each C^2 organization to help determine its enterprise-wide requirements for C^2 layered objects and embedded C^2ed objects. Abstract as well as concrete C^2ed objects are envisioned to become part of a reusable object library. C^2 objects and embedded C^2ed objects may be defined and refined in an iterative fashion through modeling, demonstration, prototyping and integration efforts as shown in Figure 3. C^2 objects are also validated and become mature as they transition from research to operational systems as shown in Figure 4. New or improved C^2 objects are always subject to constraints of system integration as dictated by the more advanced phases of acquisition. Initially, as shown in Figure 5, C^2 objects will evolve from a C^2 discipline well-founded in history, operational art, and research. For given C^2 problem domains, C^2 objects will be selected for implementation of exploratory nature. Successfully demonstrated C^2 objects will be applied to specific user operational environments and developed for potential technology insertion. Following a successful operational and technical evaluation, C^2 objects may be engineered into operational systems with minimum risk. "knowledge acquisition" (**ka**) applies to all bases of technology required for implementation of all applications in the Problem/Solution Domain (psd). Often **ka** is covered through extensive knowledge acquisition. Note that Knowledge acquisition should not be confused with knowledge acquisition. The former applies only to knowledge which pertains to the Knowledge layer of the Implementation/Technology domain (itd). The latter, however, applies to knowledge about any of the itd base layers, i.e., Supply, Equipment, Tool, Object, Information, Knowledge, and Experience layers as defined herein.

C^2 or C^2ed?

A C^2 object is a real C^2 resource or a virtual C^2 resource (not to be confused with virtual reality of a C^2 object). These two classes (varieties) of C^2 objects should not be confused. For example, human A is a real C^2 object and robot A is a virtual C^2 object which may simulate or emulate human A. Both real or virtual C^2 objects may be layered conceptually, by design, or implementation. (According to this C^2RM a C^2 object has seven classes of C^2 layers and seven classes of technology bases whether or not the C^2 object is layered as an open-system). A C^2 layer object is a C^2 application embedded in a C^2 object. A C^2 layer object is composed of two classes of C^2ed objects: a) a decision-aid(Da) object or b) a Conflict region (Cr) object. A Da object has services, utilities, and facility objects which provide access to and manipulate Cr objects. A Cr object of a given C^2 object represents a) the real or virtual environment objects, b) the real or virtual coordination objects, or c) the real or virtual C^2 objects at a given level of abstraction and from the point of view of the given C^2 object. As such, a Cr object may have any number or a subset of Cr layers. Note that in this document a layered object is an object which has layers and not an object which is identified with a layer of an object.

As an example of the distinction between a C^2 object and a C^2ed object consider two real humans, A and B, who know about each other. The perception of expected behavior from human B is presumed to be encapsulated within human A as a C^2ed Cr object representing human B and of course vise versa is also true. Similarly, the manipulation of the perception of expected behavior of human B in the mind of human A is encapsulated within human A as a C^2ed Da object. These notions are extended and formalized as part of the Application layer entities (see Figure 22 for detail). Note also that a C^2ed object is not a virtual C^2 object. A C^2ed object must be internal to either a real or a virtual C^2 object . Thus if robot A is a virtual of human A, it must have both capabilities, i.e., a) to perceive behavior associated with objects in its external environment (i.e., Cr objects), as well as b) to manipulate these perceptions (i.e., using Da objects). Virtual reality (VR) on the other hand encapsulates numerous representations of the external environment, external coordination objects, and external C^2 objects as they would be required to appear, sound, or feel to the senses of a given C^2 object. Virtual reality may be distributed across a set of simulation resources each of which includes a set of C^2ed objects necessary to trigger the various desired effects expected of them. SemiAutomated FORces (SAFOR) within VR resources are examples of Cr unit objects [20]

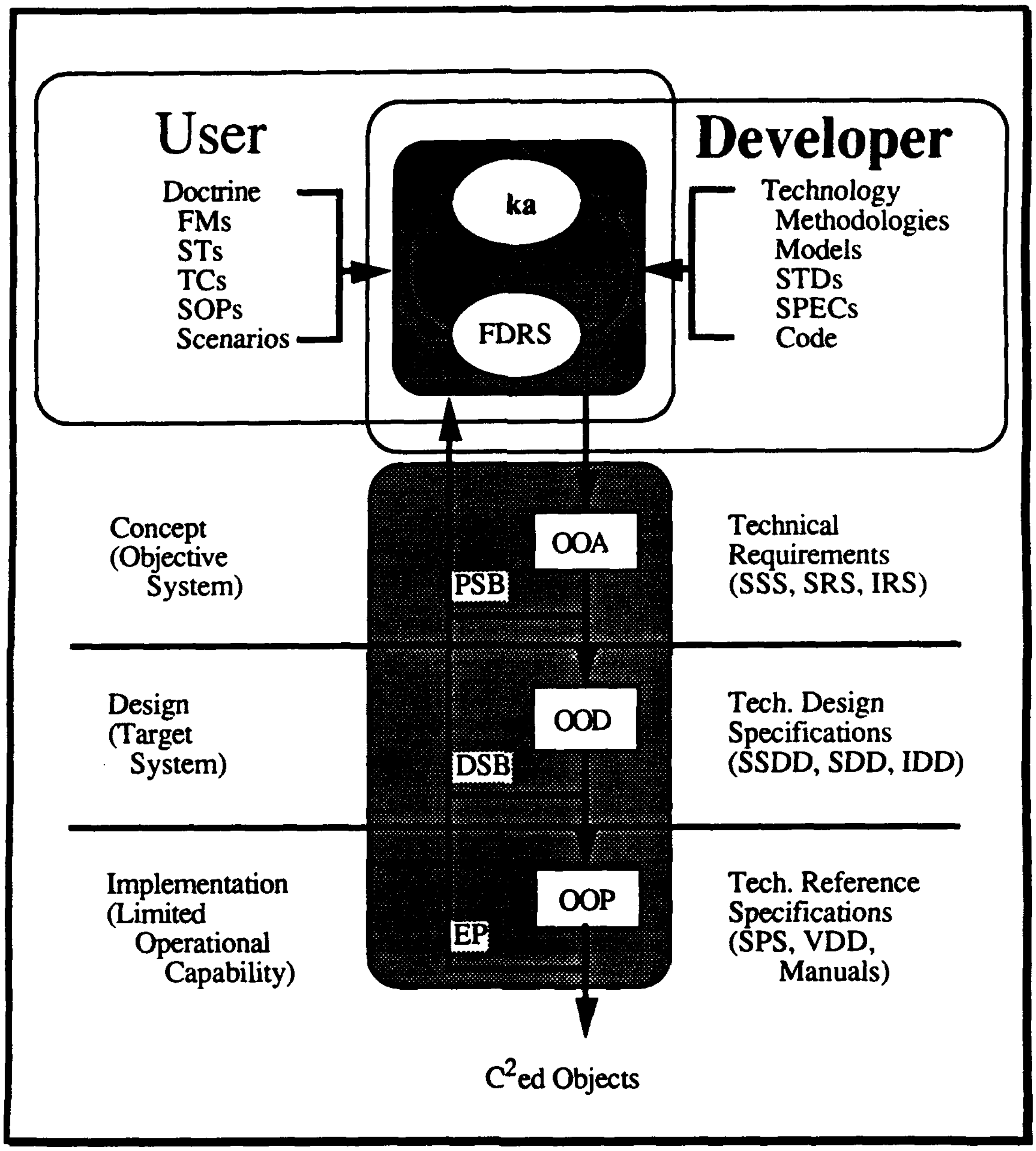

Figure 3. An acquisition process for defining and refining C^2 and embedded C^2ed objects

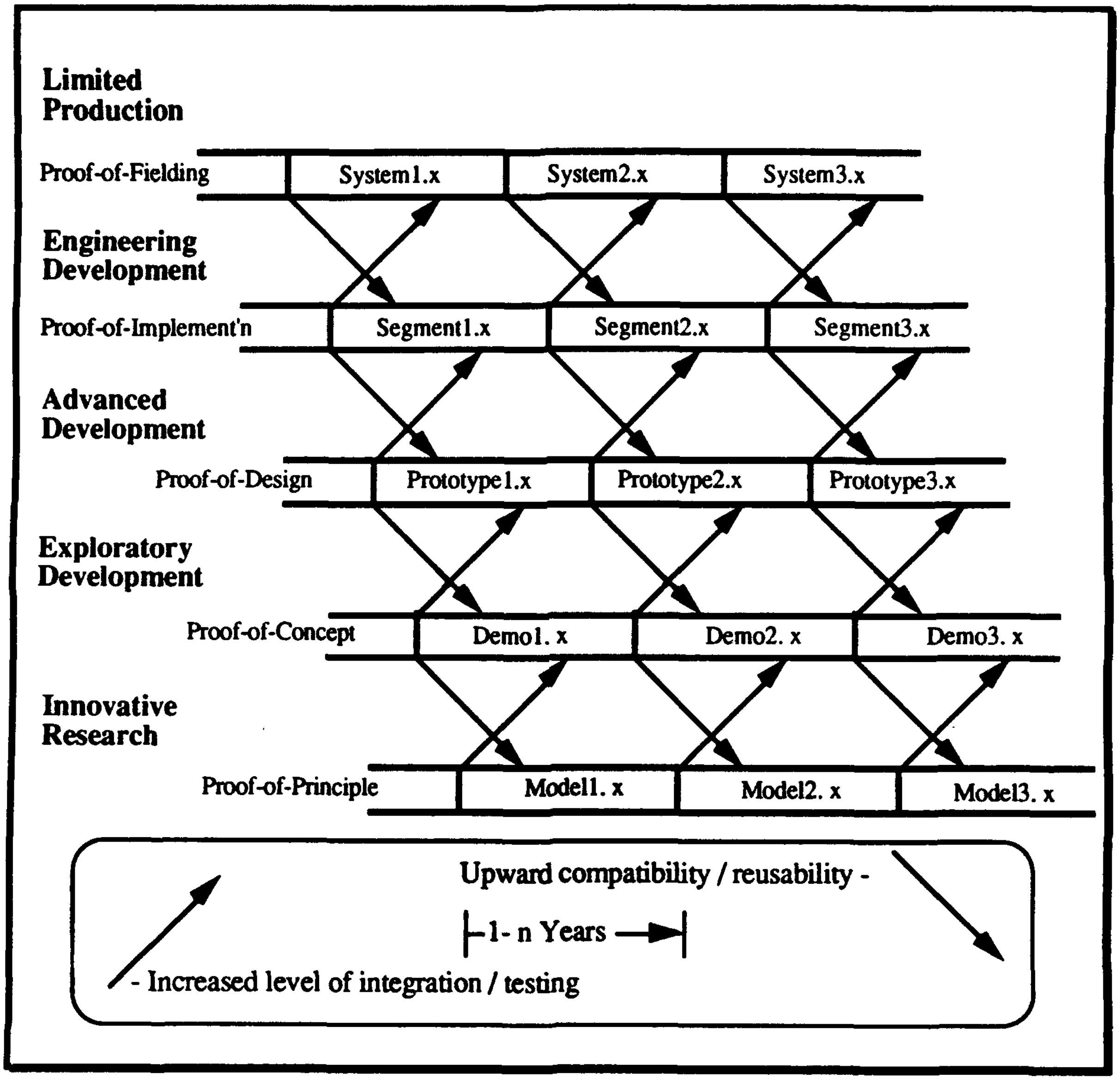

Figure 4. Acquisition phases for evolving persistent C^2 and embedded C^2ed objects

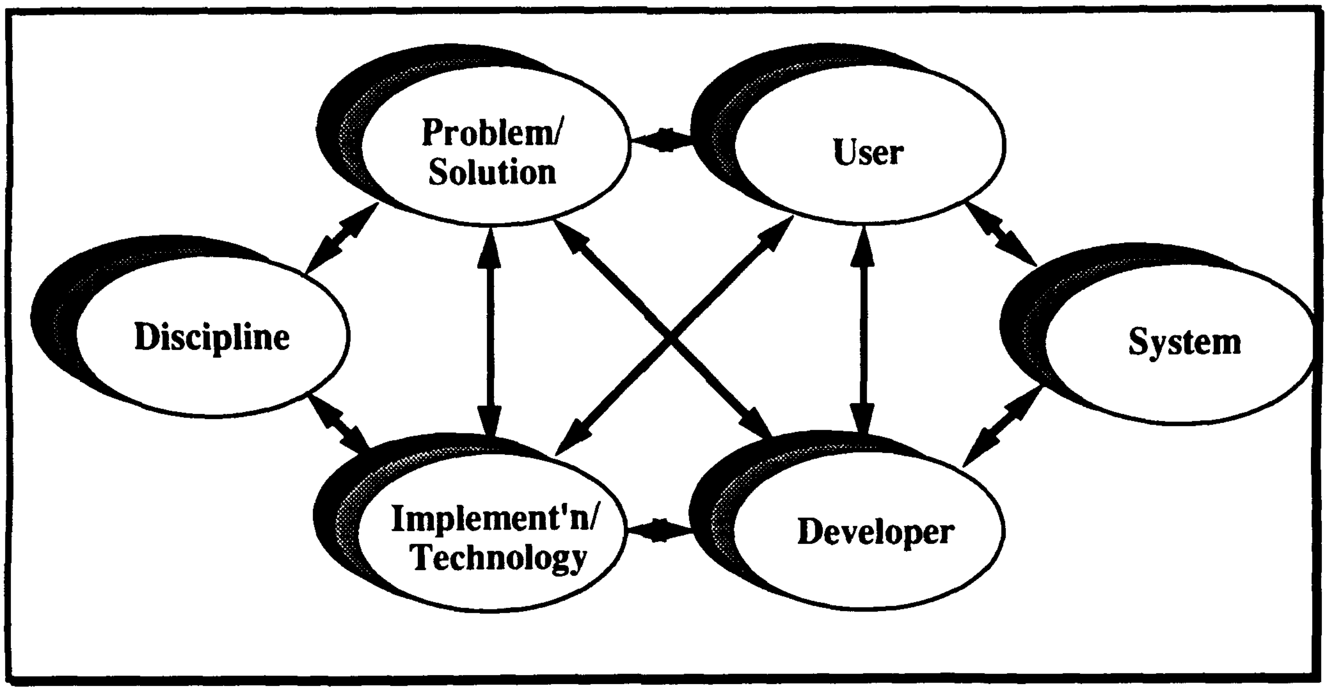

Figure 5. Domains for defining and refining C^2 and embedded C^2ed objects

Focal Points

In the long term, any organization may choose to manage the evolution of its unique C^2 objects. In the near term, however, use of the C^2RM will be subject to reviews and and comments by the C^2RM Subgroup of the BRG. *Any changes, additions, deletions or proposals* for consideration thereto should be clearly noted by its authors. Draft position papers and standards for specific types and combinations of interactions conforming to the C^2RM will also be referenced herein as they become available. The C^2RM defines herein a basic set of abstract C^2ed objects as they relate to the global C^2 process which spans more than one resource, layer or service. In addition, C^2ed objects may be applicable to a local C^2 process constrained to one resource, layer, or service. The C^2RM, therefore, provides the generic features and functionality of C^2ed objects and associated resources, layers and services to the extent that they are essential to the overall C^2 process.

Problem Oriented

The seven layer approach as instantiated for the C^2RM may be motivated from very general principles of problem solving as described herein. According to the C^2RM, C^2 Officials must be capable of accepting a C^2 Product for execution, i.e., as a reference input upon which to base a more detailed schedule to initiate, maintain and terminate involvement. A C^2 Product is essentially a definition of a set of actions. Quite generally, a C^2 Product at a lower level may refer to a solution of a C^2 Problem at a higher level. Products in general may be factorized (decomposed) into atomic products of services to define facts. The complexity of the problem and the means required for its solution will dictate the complexity involved within any given layer in terms of the number of layers and associated services. The number of overall layers and their key purpose, however, remain invariant. C^2 Problems which provide the context for C^2 Officials fall into a class of problems which may be solved through logical, *i.e.*, peer-to-peer, interactions of resources in seven layers as shown in Figure 6 and derivable as follows:

Initially consider that for every important Problem one may try to seek a Solution. Note that herein a Problem is a statement of <what_generally> is required. A Solution is a statement of <what_specifically> needs to be done <where>, <when>, and by <whom>. A Solution need only point to execution capabilities without actual execution. The <how> is addressed by <whom> as its Problem. A Solution should not be confused with Implementation which refers to an executable process of a C^2 object. Thus at the highest level of abstraction we have the Problem layer responsible for identifying, defining and prioritizing problems and developing requests for services which may be expected to lead to and culminate in solutions as implemented by the lower layers. This layer is equivalent to the C^2 Conflict layer. As shown in Figure 6, the lower layers are called the Solution layers. The Problem layer is responsible for defining the problem in a manner suitable for the subsequent development of admissible and acceptable solutions which obey some form of controllability and observability. Solutions for dynamic systems are layered further into three time domain layers and three space domain layers. In the time domain, far-term (FT) solutions are developed by the Future layer. The FT layer corresponds to the C^2 Presentation layer. Near-term (NT) time domain solutions which are consistent with a given FT solution are developed by the Present layer. Thus, a Controller is responsible for the services being provided in the present, and NT issues are assigned to the C^2 Operations layer. NT solutions must also rely upon previous-term (PT) time domain solutions as developed by the Past layer. The past may be rich with historical solutions which, if captured properly, may provide key ingredients for use in developing NT solutions by intelligent Controllers and FT solutions by intelligent Planners. Therefore, the PT layer appropriately corresponds to the C^2 Procedure layer. Time-domain solutions require the ability to activate plant resources positioned in the space domain. Solutions for individual resource-control problems are developed by the Unilateral layer which corresponds to the C^2 Assets layer. Solutions for a higher level of coordination problems involve the cooperation of resources. Two-way relationships between spatially distributed resources are handled by the Bilateral layer which corresponds to the C^2 Link layer. Solutions for facilitating control over complex relationships among all the resources are developed by the Multilateral layer which corresponds to the C^2 Network layer.

A solution developed at the Multilateral layer depends upon the options available from the Bilateral layer. In turn, success of a Bilateral layer solution depends upon the Unilateral layer options for the resources involved. In a manner which is consistent with typical use of these words, Commanders and Planners are defined as Official entities which are responsible for providing services at the Problem and Future layers, respectively. Controllers and their Agents are defined as Officials who are responsible for providing services at the two lower time-domain layers, i.e., Present and Past. Administrators, Coordinators and Operators under control are defined as Official entities which are responsible for providing services at the lowest three layers of the space domain.

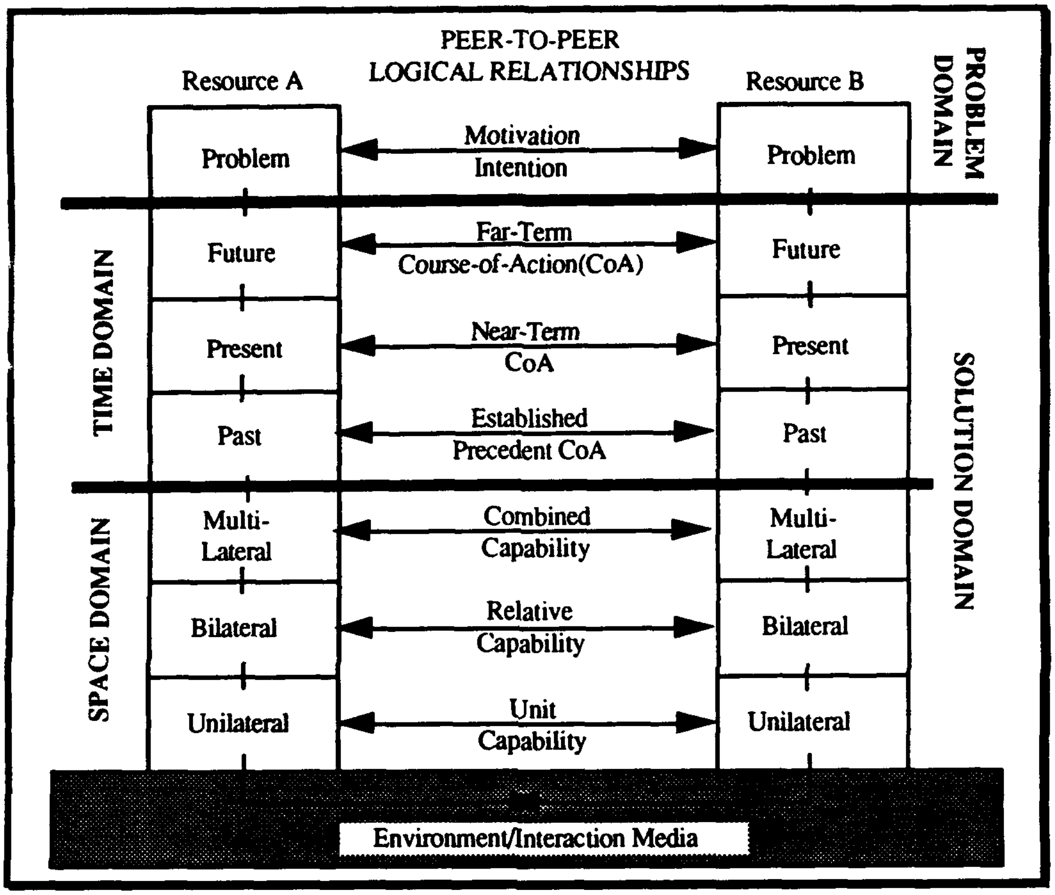

Figure 6. Layering of the Problem / Solution domain

INTRODUCTION

GENERALIZED DIMENSIONS

Four generalized dimensions of C^2 architectures form the basis for constructing C^2 systems as represented by the various C^2 paradigms. These four dimensions involve

a) Environment E_i,
b) Resource X_i,
c) Interaction Y_i, and
d) Conflict Z_i.

The universe is defined to include the totality of all physical objects, including potentially adversarial C^2 systems, F,G,H,... and the environment E. A C^2 system F/G is a collection of resources $\{X^{F/G}{}_i\}$ located in the environments $\{E^{F/G}{}_i\}$ and capable of interactions $\{Y^{F/G}{}_i\}$ Note that, at the level of abstraction of the C^2RM, the structure and dynamics of all independent C^2 systems are generically identical. The underlying dynamics of C^2 system F/G is subject to conflicts $\{Z^{F/G}{}_i\}$, respectively.

ENVIRONMENT

The environment, $E = \{E_i\}$, is depicted as a set of all physical objects, excluding the C^2 systems, in a space-time region of conflict. Thus, physical elements depicting land, air, sea, space and weather are considered as part of the environment. E also includes environmental assets (natural resources) which C^2 systems require for habitat and consumption.

RESOURCE

A primitive resource is a collection of interrelated entities which form a meta-object able to perform a useful set of processes $\{W_j(X^{F/G}_i)\}$ for C^2 system F/G, respectively. Physical processes of F or G generate and receive physical objects in the course of interacting with other resources of either F or G. Physical processes may be grouped and characterized by logical processes. Logical processes generate and receive logical objects. Logical objects represent selected variables of the state of meta-objects, objects or processes. A logical object consists of a set of records or slots from which (and in conjunction with other records or slots) conclusions, recommendations and judgements may be derived and made a part of file or database in support of any decision process.

Each resource implements a set of applications and associated interactions on behalf of its parent C^2 system. Application entities within each resource involve local observation (O), action (A) and decision (D) components which participate in a local and global C^2 Process. Furthermore, within each resource, these components may be hierarchically nested as shown in Figure 7. In nested architectures, lower level resources may become assets of higher level resources. Individual resources may be integrated through links and networks to create large-scale, compound, collective, aggregated or functional processes spanning many resources. Inversely, processes involving different resources may be integrated through links and networks to create large-scale, compound, collective, aggregated or functional resources. This relationship between resources and processes supports a dual view of C^2 systems. The process aspects of the model allow for conceptual visualization whereas the resource aspects of the model allow for design and implementation of the C^2 system.

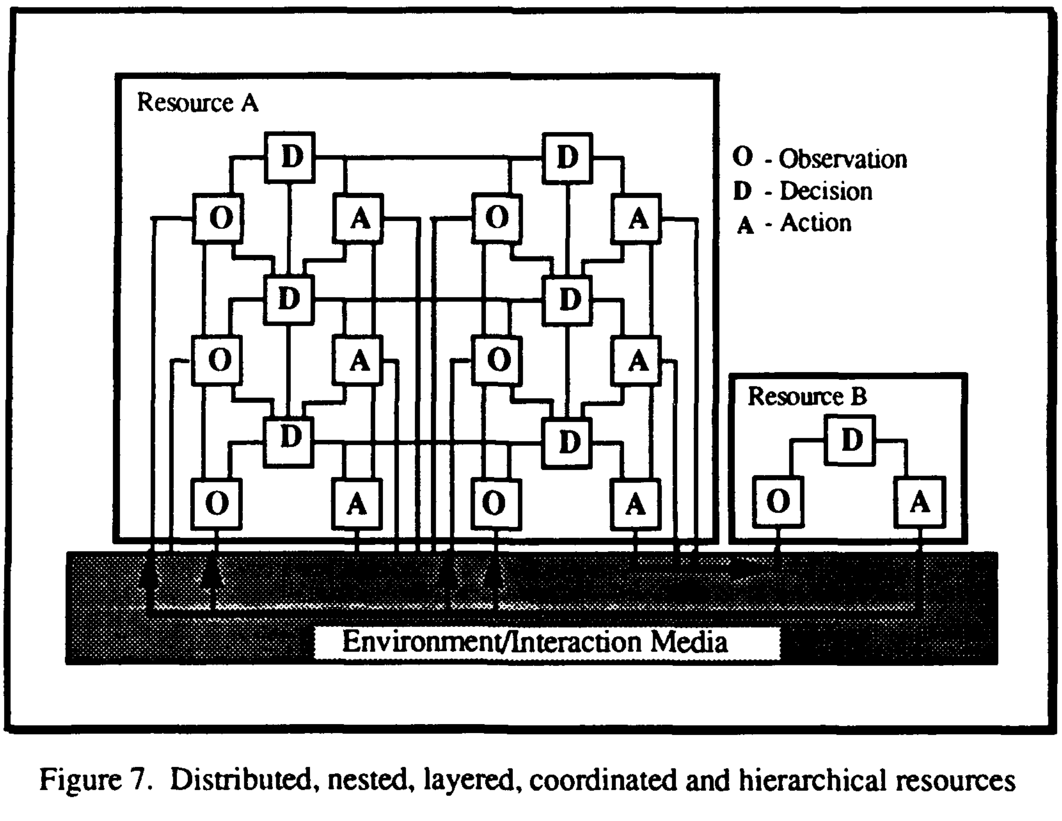

Figure 7. Distributed, nested, layered, coordinated and hierarchical resources

INTERACTION

An interaction provides for any exchange of objects between a resource and its environment. Any exchange of objects between resources, i.e., resource-resource exchange of objects, may be decomposed into a set of primitive resource-environment interactions (aka actions) and environment-resource interaction (aka event) pairs. An object exchange is typically triggered by actions of resources which may lead to potential observations of unilateral or bilateral events and outcomes. The environment plays a key role in influencing interactions. The four fundamental types of interactions are identifications, communications, transportations and inflictions. Topologically, these four types of interaction possess many properties which are analogous and isomorphic. Therefore, their layered structures are also analogous and isomorphic.

The interaction between resources of opposing forces is also described in this layered fashion which provides a common framework for understanding and developing techniques for destruction, disruption and denial of services between combatants as well as for building, facilitating and augmenting of services within a C^2 system.

Identification is an interaction which directly results in the recognition of objects in the environment. Identification is derived from observations by sensors (including radars) and is used to determine compliance with coordination objects.

Communication is an interaction which directly results in an exchange of portable tools, data, information, knowledge or experience. Communication is used to command, control and share products and and methods among the resources.

Transportation is an interaction which is associated with moving assets and maneuvering resources. It results in the motion or exchange of cargo objects. Transportation is used to carry, strengthen, equip and load resources with the necessary physical assets. Transportation interactions also include supplying or replenishing any consumables or perishables that may be needed by a resource.

Infliction is an interaction which may result in adverse impacts leading to destruction or damage of target assets, or to degradation or disruption of operations of target resources. Infliction is used to reduce the capabilities of target adversarial C^2 systems through a variety of means including: a) fire of explosive projectiles, rockets, torpedoes, or missiles, b) radiation of electromagnetic energy, or c) employment of barriers, obstacles or decoys.

Each resource involved in a conflict, as a minimum, must be capable of communications. Data, information, knowledge or experience are continually being exchanged among processes, explicitly or implicitly. Thus, communications in general is critical in coupling between and among processes which are either co-located or distributed within a given resource or across a number of resources. In addition, processes may be coupled strongly or weakly depending upon the need for physical separation between the resources and the potential interactions involved. The capability for other types of interactions depends on the specialty of the resource. For example, weapon resources must be capable of infliction, sensor resources must be capable of identification, and logistic resources must be capable of transportation. An interactive resource as depicted in Figure 8 includes the physical assets to interact with other resources using interactions of all four fundamental types in an integrated manner. Note that the different fundamental types of interactions are depicted by the differently textured thick lines connecting the resource with the environment.

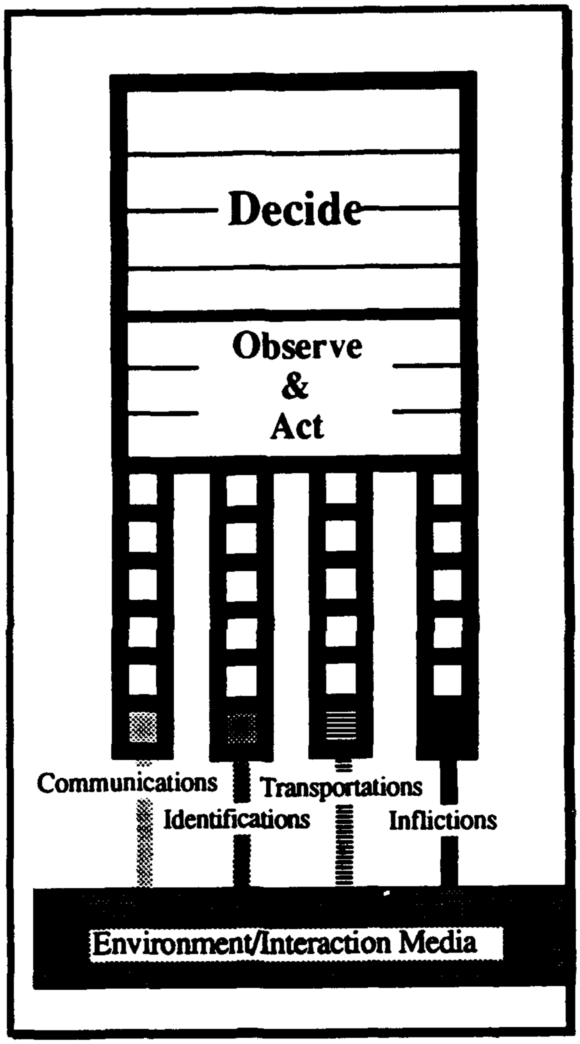

Figure 8. An interactive resource

CONFLICT

Availability of environmental assets is a primary concern for any C^2 system since it may constrain its ability to survive and thrive in peace within E. Peace is the state of a space-time region in E which denotes a perceptively attainable degree of freedom through nonmilitant means, i.e., conflicts between and among coexisting C^2 systems are resolved in a civilized manner through avoidance, negotiation, mediation, arbitration or litigation. Competition for natural resources, however, as well as other apparently rational or irrational C^2 system requirements, may lead to the development of space-time regions of militant conflict. As shown in Figure 9, each space-time region of conflict must be supported by a number of varied resources or processes which depend upon the scale of the conflict. The highest region of militant conflict is a space-time region characterized by the state of war nested in the region of peace. In addition to the region of war, the region of conflict consists of a set of five other hierarchically nested regions with layered relationships. The structure of the environment, E, and its associated regions is also formally defined as an integral part of the C^2RM. As defined herein, any disagreement, dispute or discontent evolving from constraints upon required freedom may lead to a military option for resolution of conflicts over concerns, interests, influences, maneuvers, interchanges and changes. These motivations are nested hierarchically to characterize the space-time regions of war, campaign, battle, combat, engagement and armament, respectively.

CONFLICT REGION OBJECTS

Associated with the four dimensions{E, X, Y, Z} are logical C^2 objects which correspond to the view of the universe maintained internally by each resource X_i. This view of the universe is decomposed into three abstract superclasses named Conflict region objects:

a) Conflict region unit object,
b) Conflict region environment object, and
c) Conflict region coordination objects.

Conflict region objects are also modelled in accordance with the C^2RM abstract object trees and associated layers. Conflict region unit objects correspond to real resource entities. Conflict region environment objects correspond to real environment entities. Conflict region coordination objects correspond to real command and control entities which provide the mapping of real resources to real environment through real interactions.

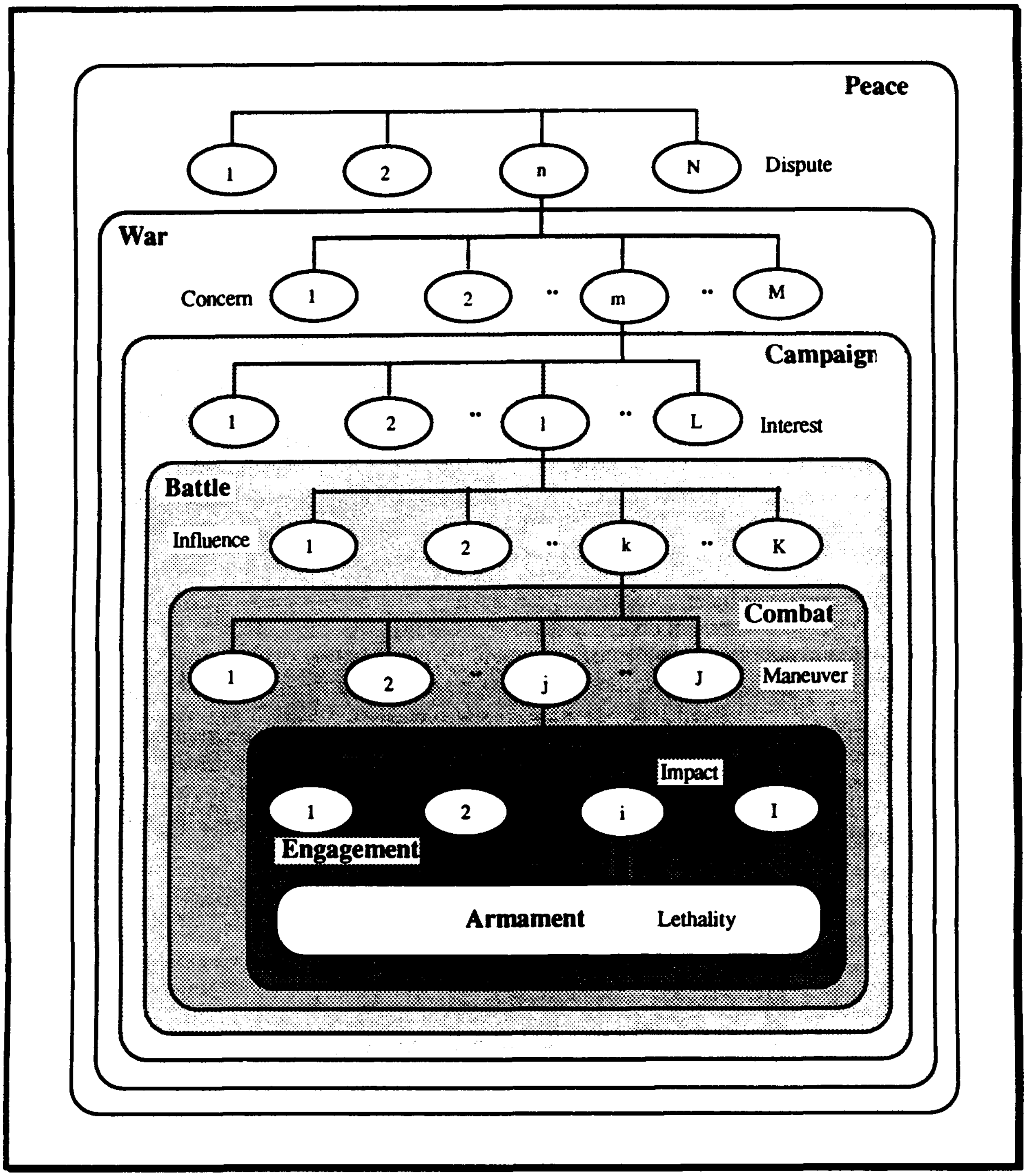

Figure 9. Hierarchical levels and nested regions of Conflict

ABSTRACT OBJECT TREES

Intrinsic to C^2 systems is a set of multidimensional abstract object trees essential to define and refine a coherent, consistent, and recurring set of C^2ed objects. These abstract object trees are listed in Table 3.

Table 3. Classes of abstract object trees

a)	Unit object trees,
b)	Problem/Solution domain object trees,
c)	Implementation/Technology domain object trees,
d)	Method object trees,
e)	Product object trees, and
f)	Official object trees.

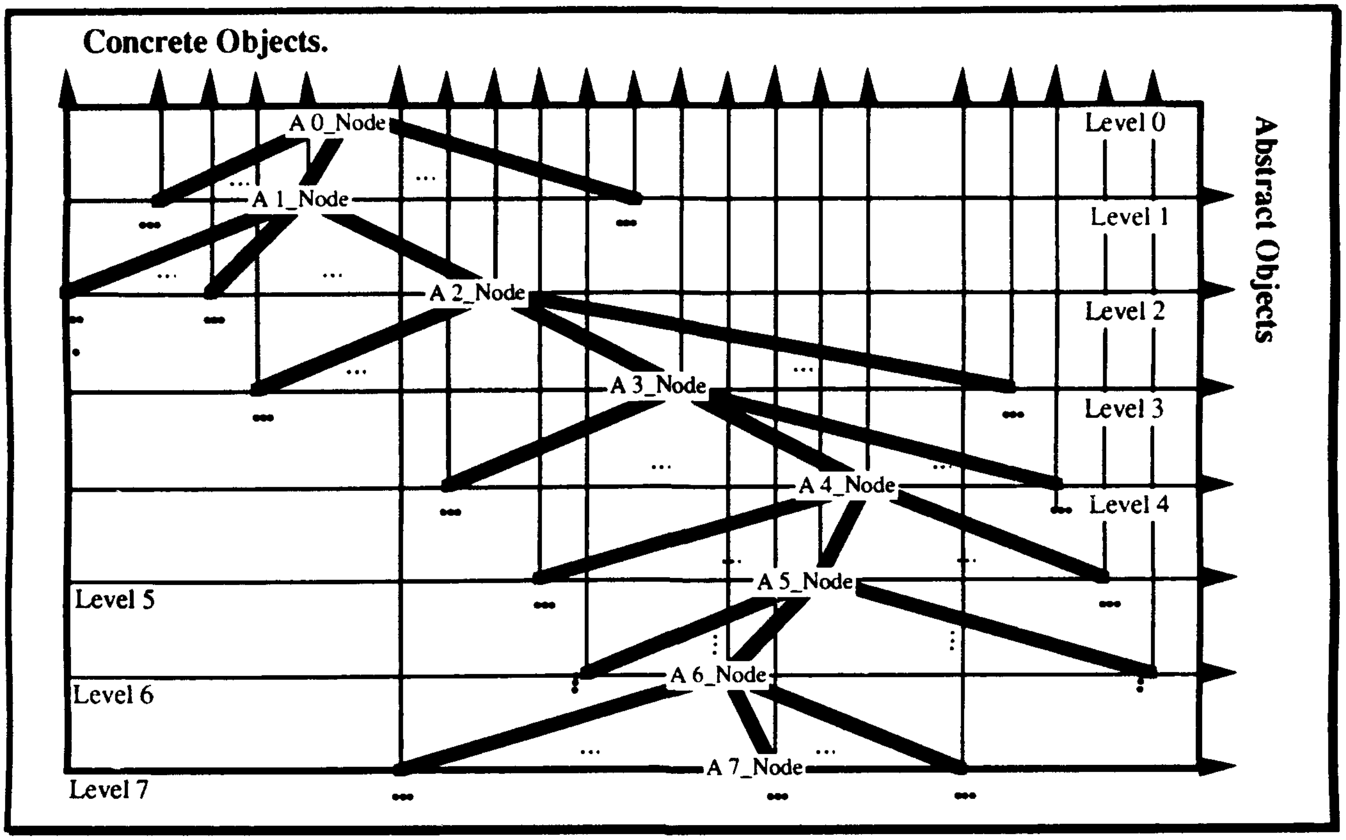

Figure 10. A generic abstract object tree

GENERIC ABSTRACT OBJECT TREES

An abstract object refers to any or all concrete objects at a given level of the abstract object tree. A generic abstract object tree is depicted in Figure 10. Specific concrete objects are referred to as nodes. Specific abstract objects are referred to as levels. Thus a 3_Node is a concrete object which corresponds to the class of Level 3 abstract object. Whereas abstract objects are common to many organizational entities, concrete objects are unique to each organizational entity.

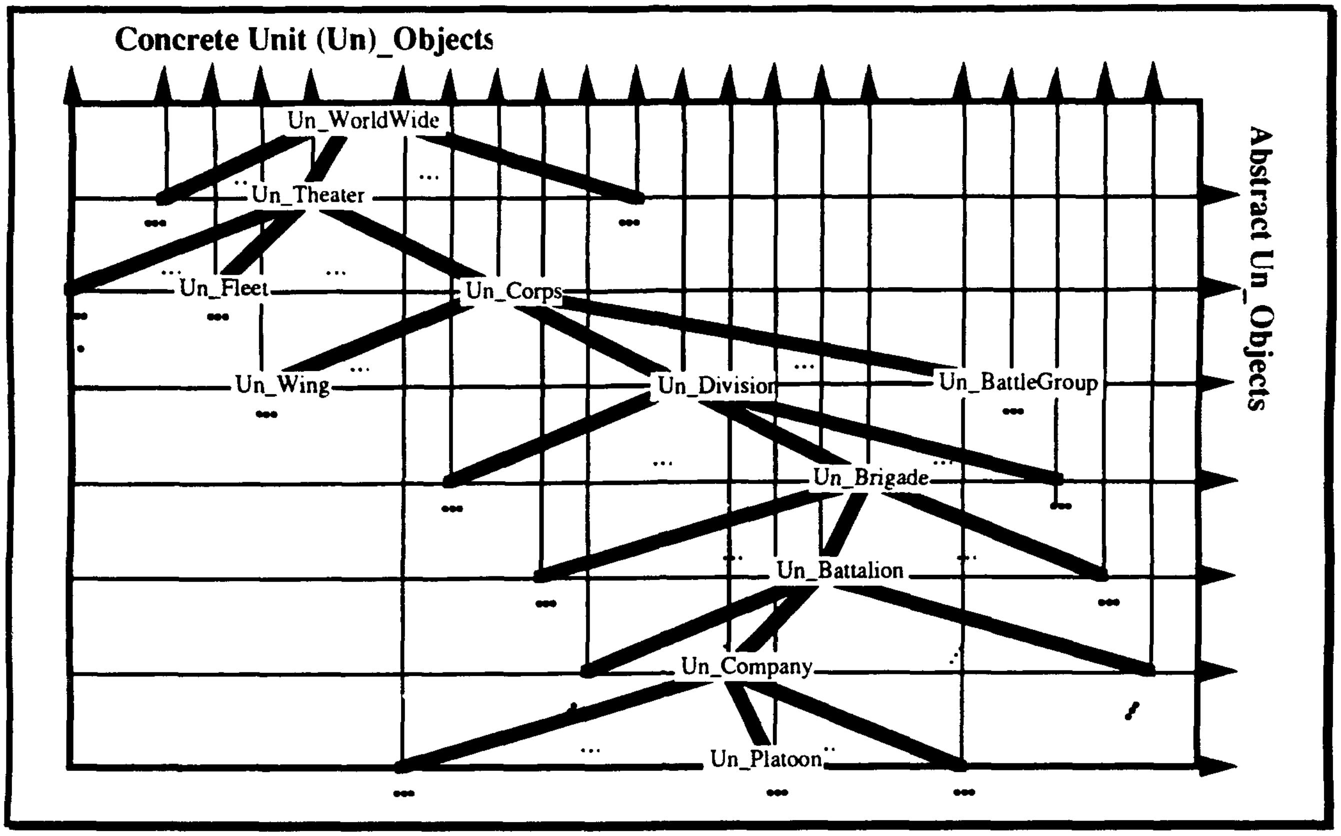

Figure 11. A Unit object tree

UNIT OBJECT TREES

Echelons of resources of a C^2 system are represented by a unit object tree. Army Echelons are shown in Figure 11 as an example. Each echelon down to its primitive resource units provides a service to its parent/command organization. Each echelon may be regarded as a complete C^2 system with a corresponding structure consistent with the same generic metastructure which characterizes the C^2RM layered architecture. Therefore, this model pertains to each echelon to the extent that it is autonomous. Each node of a Unit object tree forms the root of a C^2_problem/solution_domain tree. Each leaf of a Unit object tree is considered an Asset of the C^2 system.

Every unit object has a command object tree as shown in Figure 12. The command object tree is provided with a separate set of resources in addition to subordinate units. Thus, a unit consists of command resources and subordinate units. Such a nested lattice work defines an organization. An organization may be fine tuned to execute specific missions. This is accomplished through attachment or detachment of subordinate units as required from other units. The resulting organization is temporary and defines a mission organization (aka task organization). When a unit is first created its role is characterized by the set of missions which the unit is capable of executing. Abstract units as well as concrete units may be tailored to specific roles by mixing and matching of subordinate units and assets.

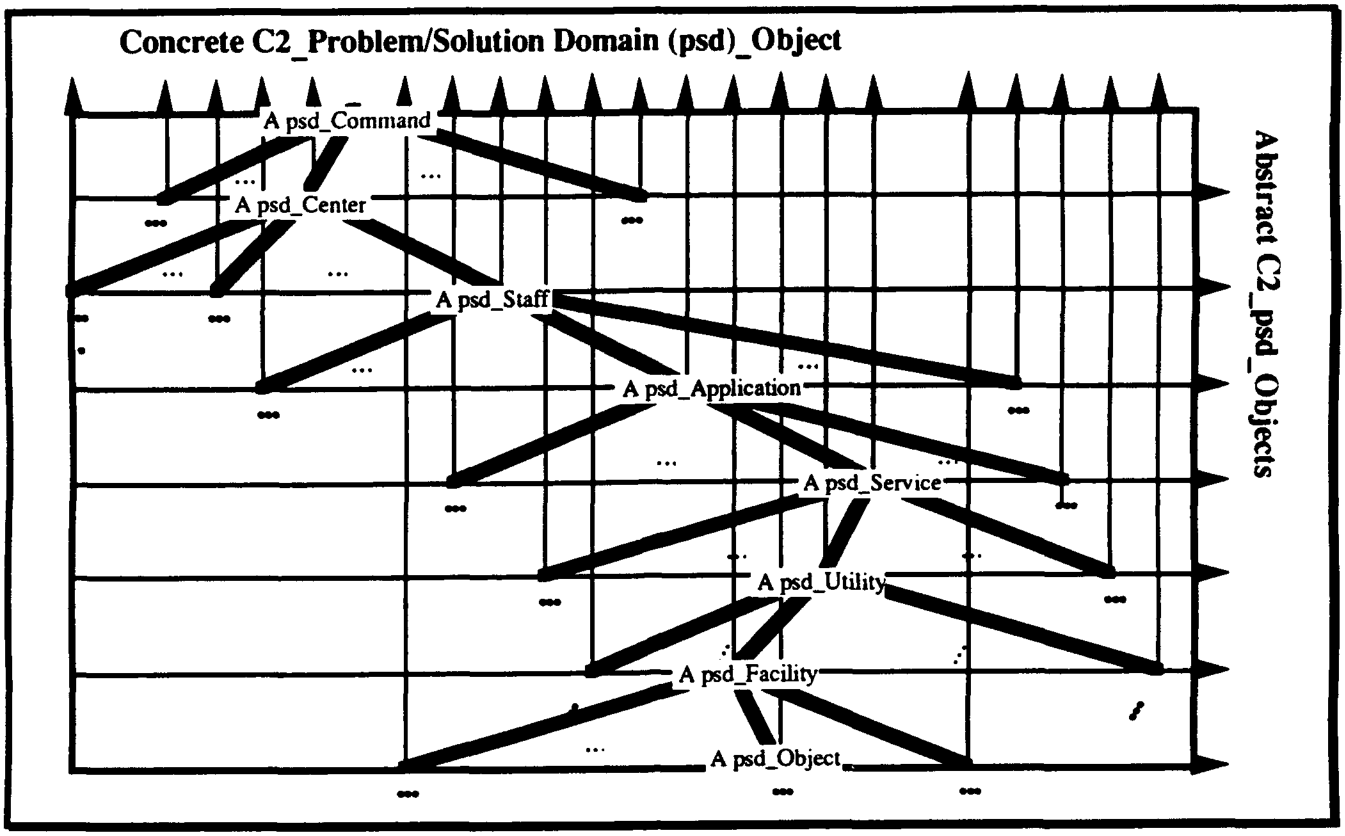

Figure 12. A Problem / Solution domain object tree

PROBLEM / SOLUTION DOMAIN OBJECT TREES

The hierarchy of problems and solutions of C² are represented by a problem/solution_domain object tree as shown in Figure 12. A problem / solution_domain object tree provides the semantics of describing the behavioral structure of resources and their applications. Note that all applications are concrete objects of this tree. Psd_command includes the headquarters of the unit. The psd_command includes one or more psd_centers with a subset of the staff.

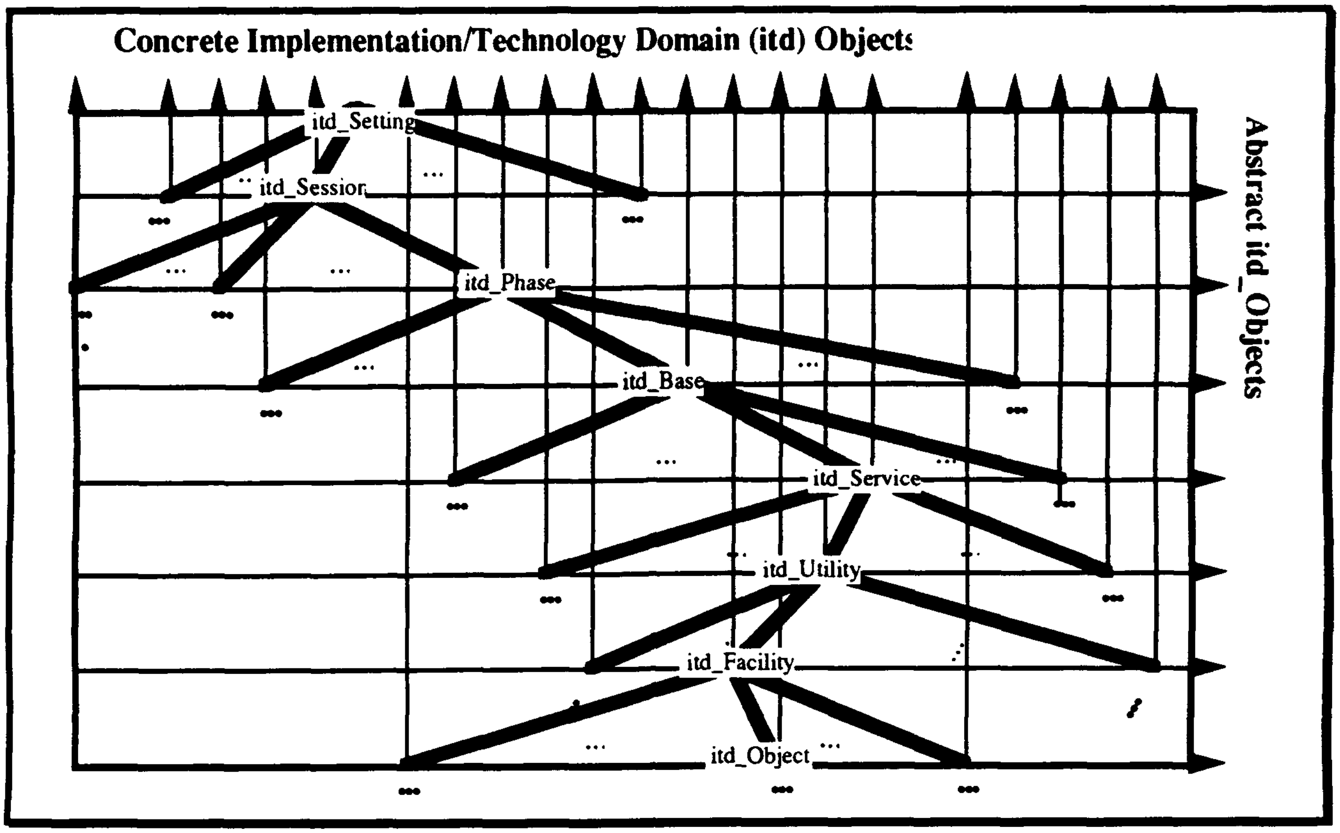

Figure 13. An Implementation / Technology domain object tree

IMPLEMENTATION / TECHNOLOGY DOMAIN OBJECT TREES

The hierarchy of implementations and technology of C^2 systems is represented by an implementation / technology domain object tree as shown in Figure 13. An implementation/ technology_domain object tree provides the syntax of describing the technological structure of resources and their bases. Note that specific bases for implementation are concrete objects of this tree.

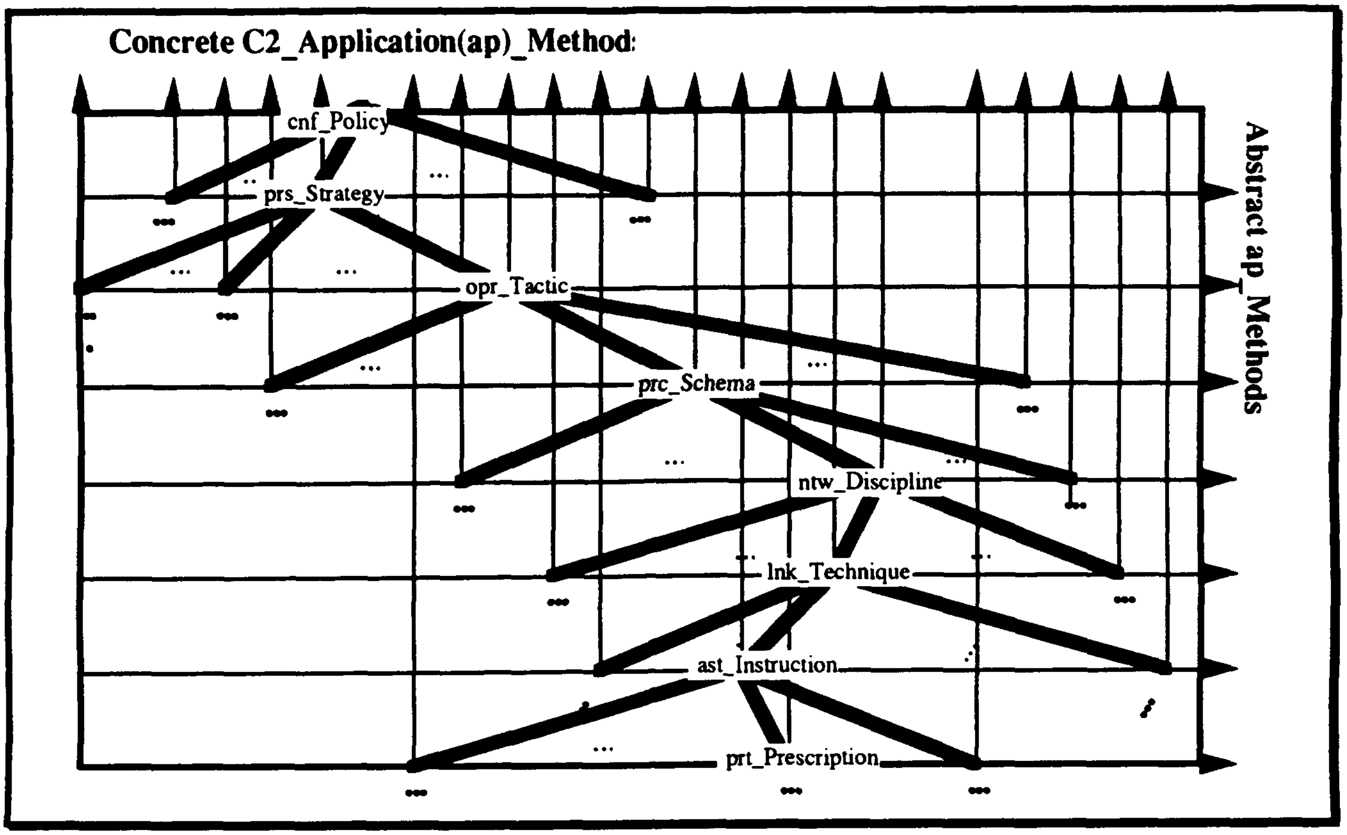

Figure 14. A Method object tree

METHOD OBJECT TREES

Methods associated with each of the layered applications are organized in a Method object tree as shown in Figure 14. Methods are specific operations of application objects within a given application layer.

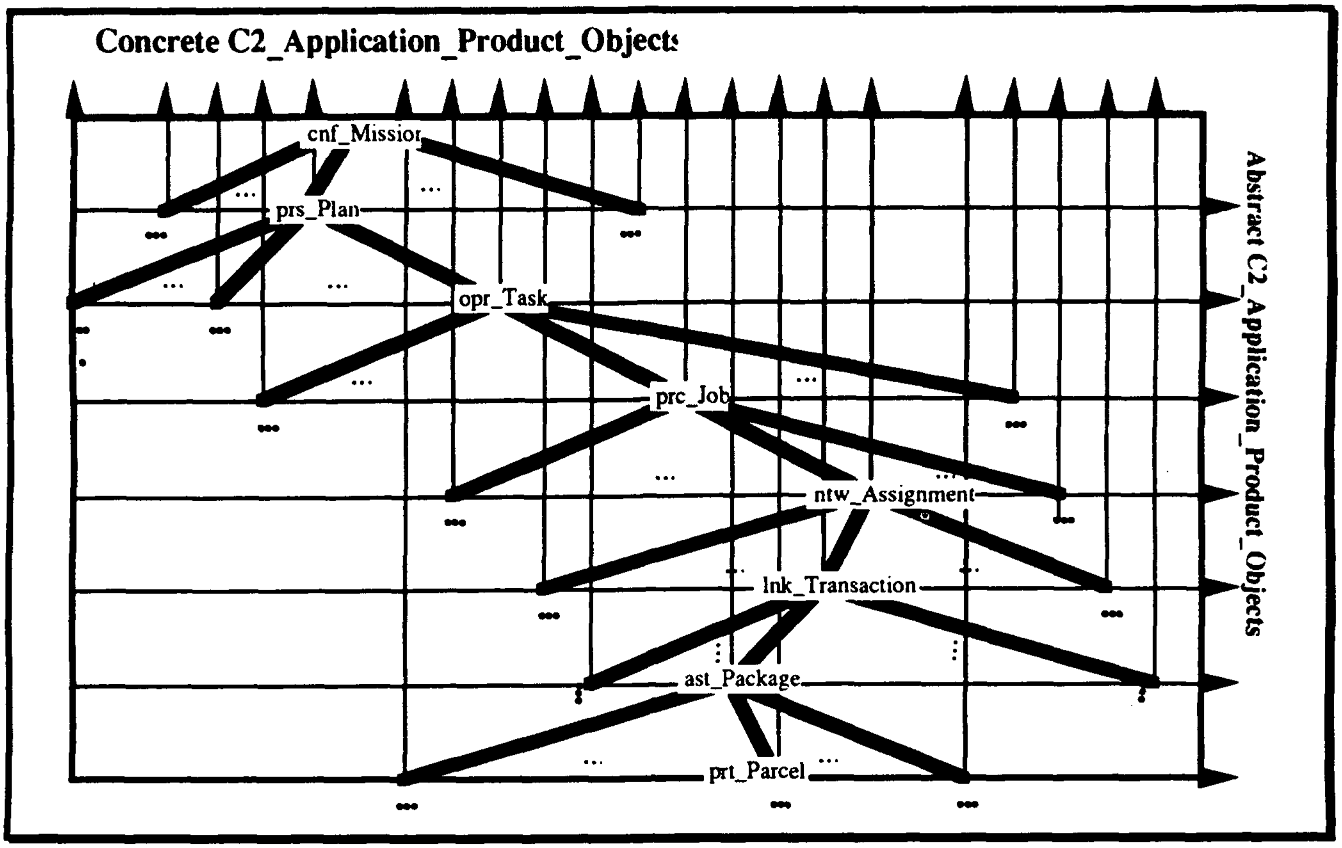

Figure 15. A Product object tree

PRODUCT OBJECT TREES

Products associated with each decision_aid object are organized in a Product object tree as shown in Figure 15. Products are specific objects associated with decision-aid objects. There are two subclasses of each Product which are always paired: Requirement and Status. Requirements evolve from the top C^2 Application layer and spawn lower level Requirements. Statuses evolve from the lowest C^2 Application layer and gets consolidated or fused into upper layer Statuses commensurate with the upper layer Requirements in effect. Products may be factorized (decomposed) into a set of Facts. Facts may be static throughout the lifetime of the factorized Product (e.g., typically asset allocation). Facts may be dynamic (e.g., typically asset location). Example Requirements are commands, orders, or requests. Example Statuses are reports or summaries. A Mission is a key Product of a C^2 system. As such, a Mission abstract object is at the root of Product object trees. The Mission of the C^2 system defines the goal, aim, command, objective, purpose, intent, decision requirement, function, or desired state of the C^2 system. It is the highest level Product of a C^2 system. The mission must be derived from the conflict in which the C^2 system is involved. A Mission may be decomposed into a series of sub-missions. At the system level, the Mission is supported by layered applications global to all Resources and associated entities which are local to each Resource. Entities, in turn, are modularized to constitute local C^2ed objects to be associated with the Mission. These C^2ed objects are the nodes which branch off the psd_Application node on the Problem/Solution domain object tree.

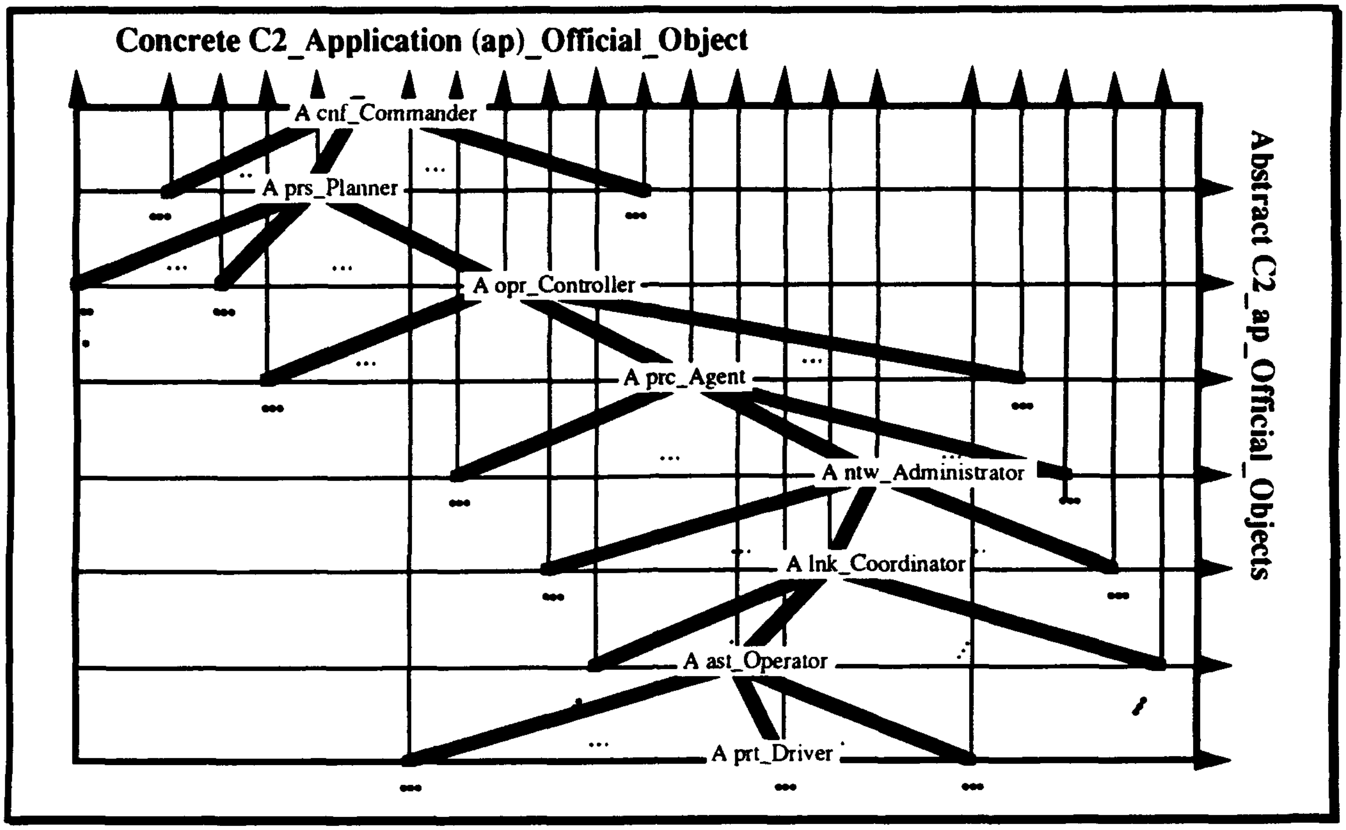

Figure 16. An Official object tree

OFFICIAL OBJECT TREES

Official capacities associated with each layered application are organized in an Offical object tree as shown in Figure 16. One or more human officials may be associated with a given psd_Application and may assume responsibility for one or more of its Products. In general, C²RM Officials represent only a layered part of human officials and extensive augmentations by corresponding layers of associated software services.

Commander

A Commander is the key Official object of a C² system. As such, a Commander abstract object is at the root of Official object trees. The Commander is responsible for every C² system unit of action, force or power used in any space-time region. Conversely, no unit of action, force or power should act without the authority of its Commander. The authority of the commander may be formalized but must always retain the flexibility to take initiatives as required to meet unexpected conflicts and exploit opportunities which enhance the survivability posture of the C² system. Thus, the system which executes commands is strongly coupled to the system which generates commands, and, as such, both may be distributed hierarchically and are treated in an integrated fashion as one C² system. A resource may execute an interaction without a specific or explicit mission or command provided such an interaction contributes to its commander's intent.

A commander's responsibilities should span the seven sublayers of conflict. The encapsulation of these responsibilities within each conflict sublayer constitutes the essence of the associated commander's titles as described for each conflict sublayer. These titles should not be confused with titles of real commanders and should be used to denote abstract commander objects only within the framework of the C^2RM.

A resource may have states which are unknown or unobservable from the point of view of a given official. States which may be irrelevant to a given official, but which may not be ignored in the overall execution of the resource should be identified, observed or monitored by higher, lower or peer level officials. Clearly, a coordination mechanism must be established to enable cooperation among officials. Any conflict between two or more N_officials must be arbitrated by an (N+1)_official. An intelligent N_official must be responsive to an intelligent (N+1)_official. Any conflict between peer-level intelligent officials which may evolve in the course of developing or executing N_products will be resolved through negotiation or brought to the attention of the (N+1)_official for mediation or arbitration. An intelligent official will be able to anticipate potential conflicts ahead of product execution time. When an (N+1)_official is unavailable or does not exist, an N_official must attempt to collaborate if possible with other peer-level N_officials to form a Coalition.

THE DECISION-THEORETIC C^2 PROCESS

A process is defined as a set of entities arranged to carry out a prescribed function. If this set of entities involves more than one resource, then the process is said to be global. Otherwise it is a local process. The common denominator of all C^2 paradigms can be shown to be decision-theoretic in nature. As shown in Figure 17a, to each system-level (global) observation (**O**) there corresponds a system-level action (**A**). A system-level decision-rule (**D**) is invoked to select a system-level action for a given system-level observation. {**O**, **D**, **A**} are classes of a C^2 function. Sequences of global observation-action pairs characterize the dynamics, i.e., the rules of behavior and evolution, of the system-level decision process. This structure and associated overall cyclical process (as conveyed by the arrows) is called the C^2 process. Functionally, each of the **O**, **D**, **A** subsystems of **F** or **G** represents a complex, collective and compound process which must be logically interconnected as shown. When tightly coupled, the physical perspective is identical to the functional perspective shown in Figure 17a. When loosely coupled, the physical perspective becomes more complex as typified in Figure 17b. Due to weight, size, power and survivability considerations, physical resources must be established to effect the functional subsystems in a distributed and dispersed manner. Their logical interconnections become highly uncertain due to the environment through which they must communicate. As the system is stimulated by a set of external events, rules within each resource are applied in succession or in parallel to make decisions which, in turn, initiate a set of desired actions. Due to uncertainty in system reliability and differences between observations and reality, an action may result in a variety of outcomes.

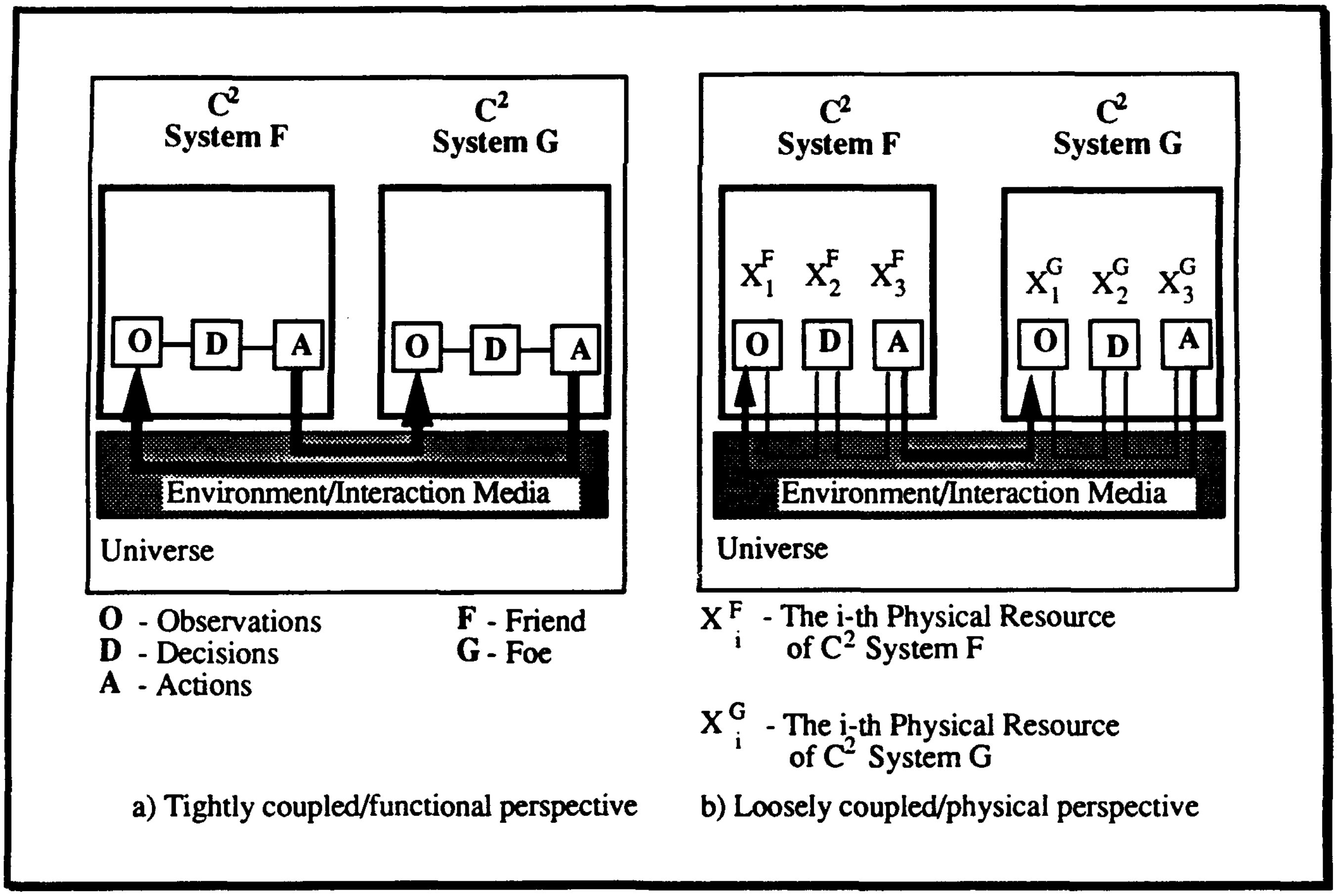

Figure 17. The C^2 process/cycle

Not every C^2RM resource must have the full complement of interaction ports. As shown in Figure 18, local C^2 processes within each C^2RM resource support the loosely coupled global C^2 process of Figure 17b. The C^2RM inherently applies to tightly coupled systems as shown in Figure 18 where communications is assumed to exist but does not play an explicit part and all resources are aggregated to form a single resource C^2 system. The C^2RM also applies to loosely coupled C^2 systems as shown in Figure 19 where communications plays an explicit part of all resources which comprise the C^2 system.

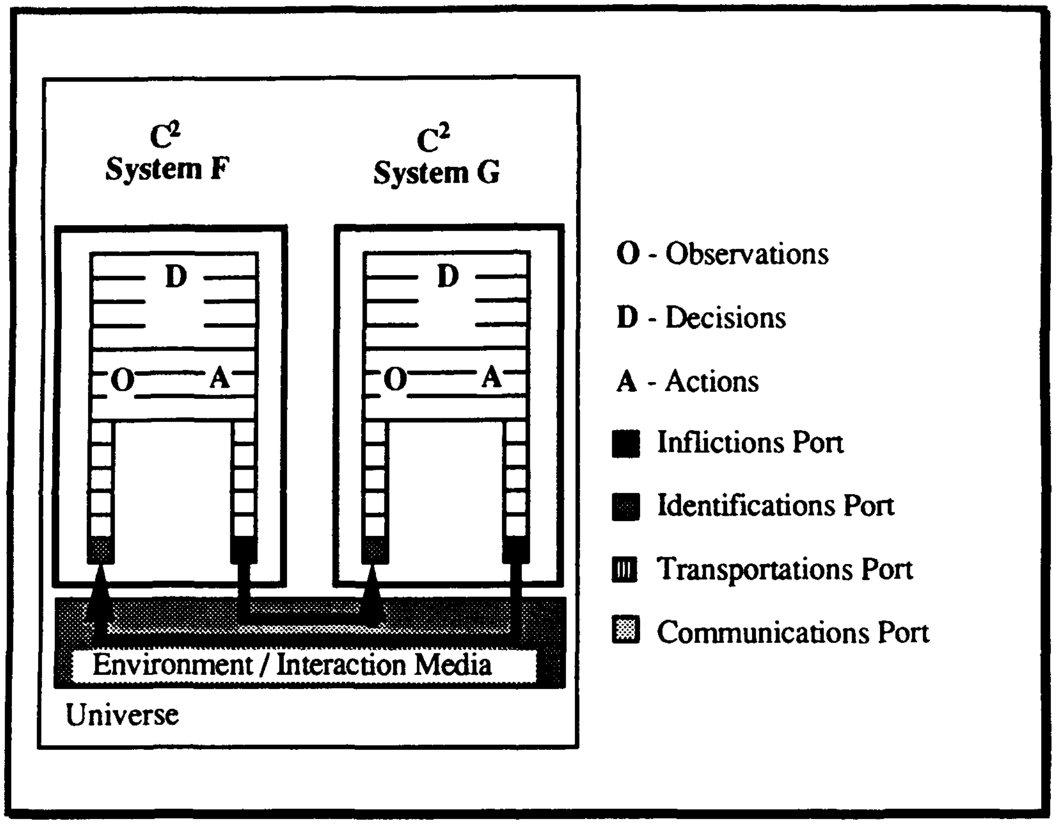

Figure 18. The C^2 process through tightly coupled { **O**, **D**, **A** } subprocesses

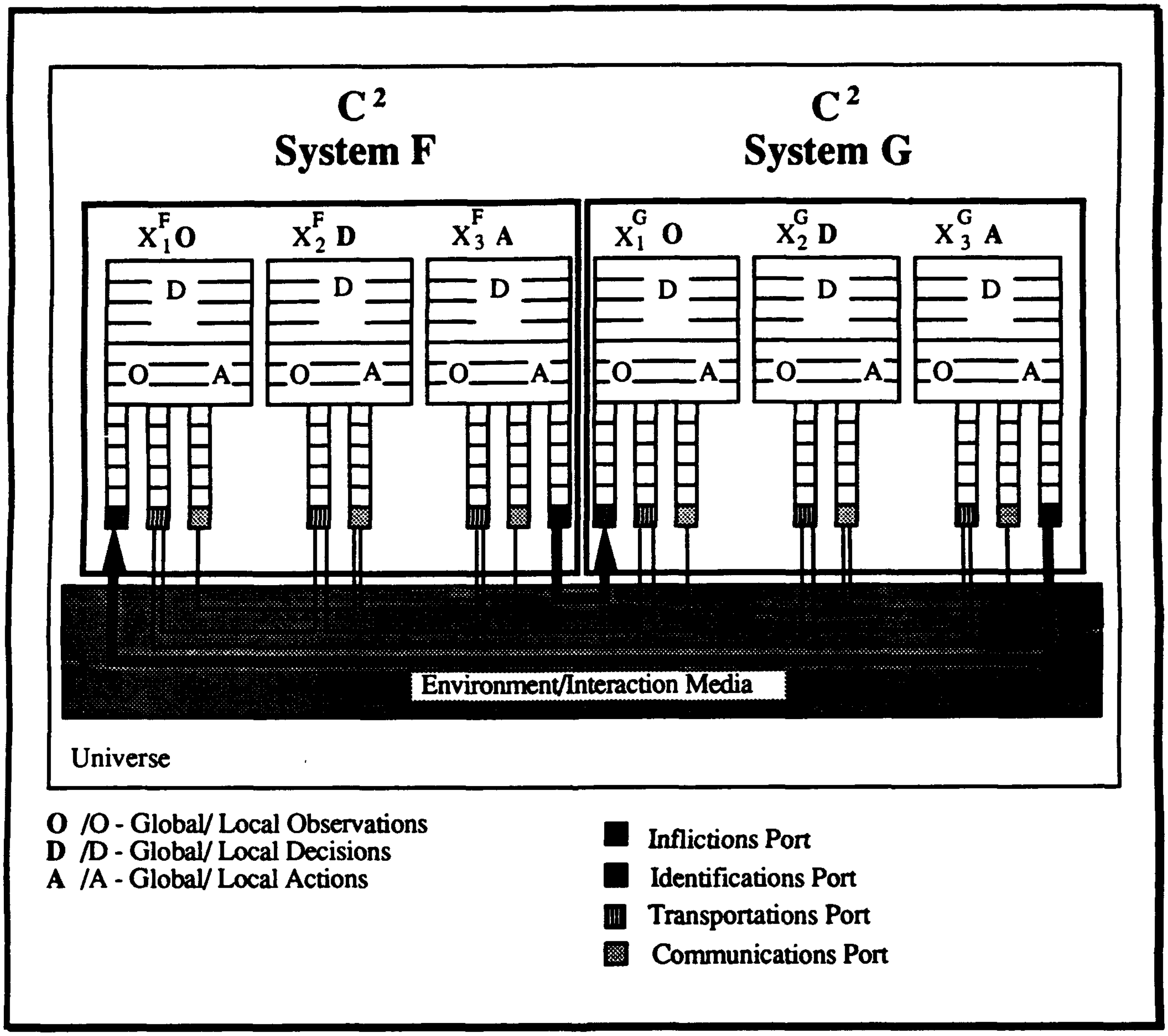

Figure 19. The C² process through loosely coupled { O, D, A } subprocesses

C^2 CYCLE

Each C^2 system must continuously and recursively go through the global C^2 process in cycles regardless of how the subsystems are partitioned. As shown in Figure 20a, it is possible to add feed-forward connections which use the latest observations for adjustment of decisions to provide a priori predictive actions and thereby speed up the main C^2 cycles. It is also possible and often desired or required to incorporate feed-backward connections for fast corrective actions which use previous decisions as desired states to react a posteriori to unpredictable changes in the environment and other systems. This is shown in Figure 20b.

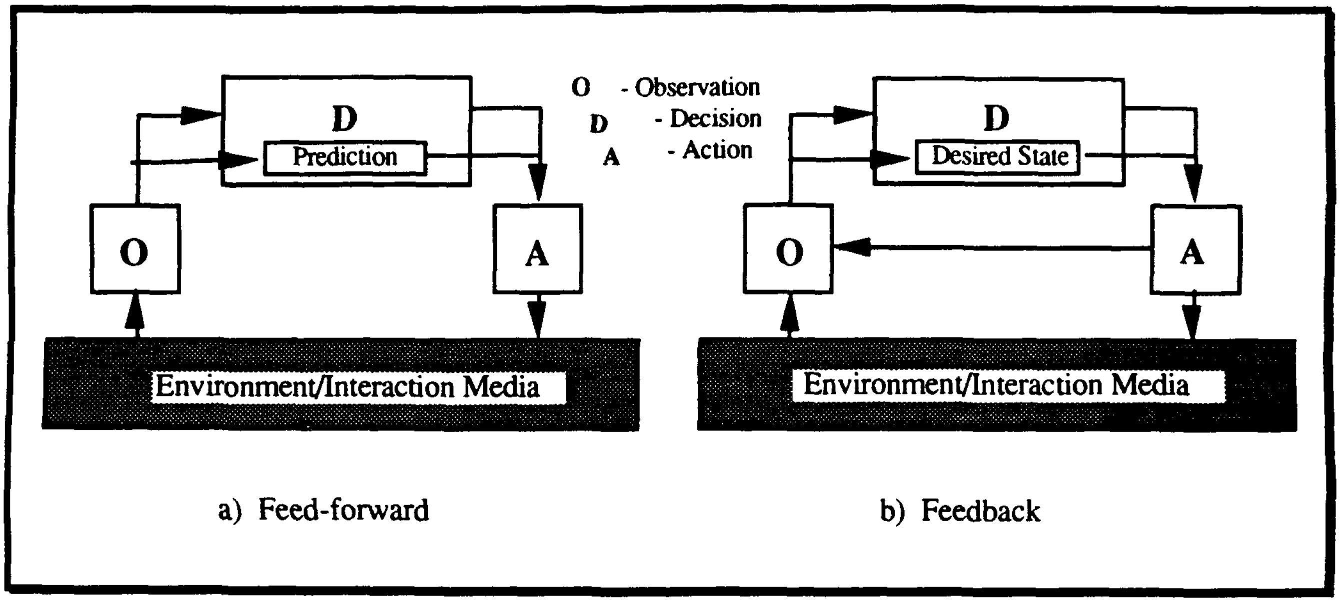

Figure 20. A C^2 system

APPLICATION LAYERS

Application Entities
Application entities are grouped in a layered fashion to exploit a specific interaction capability as shown in Figure 21. From the perspective of a specific interaction port, therefore, the other interaction ports (1.x-6.x) as well as the C^2 application layers (7.x) constitute a complete set of application entities.

The grouping of entities within an application layer must be consistent with the Problem/Solution_Domain object Tree as well as with the Implementation/Technology_Domain_ Object tree. Entities required for a given process are organized into C^2ed objects which are ordered sequentially or in parallel as required by the process. Internally to each Application layer there are four intrinsic C^2ed objects: a) psd_application object, b) psd_service object, c) psd_utility object, and d) psd_facility object. These classes of objects are typically derived from user functional descriptions and requirements specifications. Inherently bound and providing depth to each of these objects are the four intrinsic objects derived from the Implementation layers, namely: a) itd_base object, b) itd_service object, c) itd_utility object, and d) itd_facility object. These classes of objects are typically derived from technical descriptions and design specifications (e.g., in accordance with MIL-STD-2167A). The nested architecture of intrinsic C^2ed objects is depicted in Figure 21. Each intrinsic C^2ed object at layer N is also associated with a product object which captures the latest results for collaboration or sharing with other C^2ed objects at layer N, N-1, or N+1.

Access to Conflict_Region Entities
In addition to intrinsic C^2ed objects, inherent in each application of a resource is an internal representation of extrinsic C^2ed objects encompassing environment objects, unit objects, and coordination objects. Note that, as shown in Figure 22, a facility object is allowed to directly access or activate individual features of specific Conflict region objects which are within the purview of the intrinsic C^2ed object supervising the facility object. A utility object may indirectly access or activate a single feature for any subset of Conflict region objects through associated facility objects. Utility objects are constrained only by the configuration setup and workspace available through the supervising higher level intrinsic C^2ed objects, i.e., service objects or application objects. A service object may access or activate a given set of features for any subset of specific Conflict region objects which are within the purview of the supervising application object. Each intrinsic C^2ed object must provide the context and workspace for processing, storage, entry, display and access to its subordinate objects and their products which may include textual, numeric, tabular, graphic, sound, or image content formatted in accordance with its applicable information exchange standard. Annex C provides an object specification framework for C^2ed objects consistent with the C^2RM.

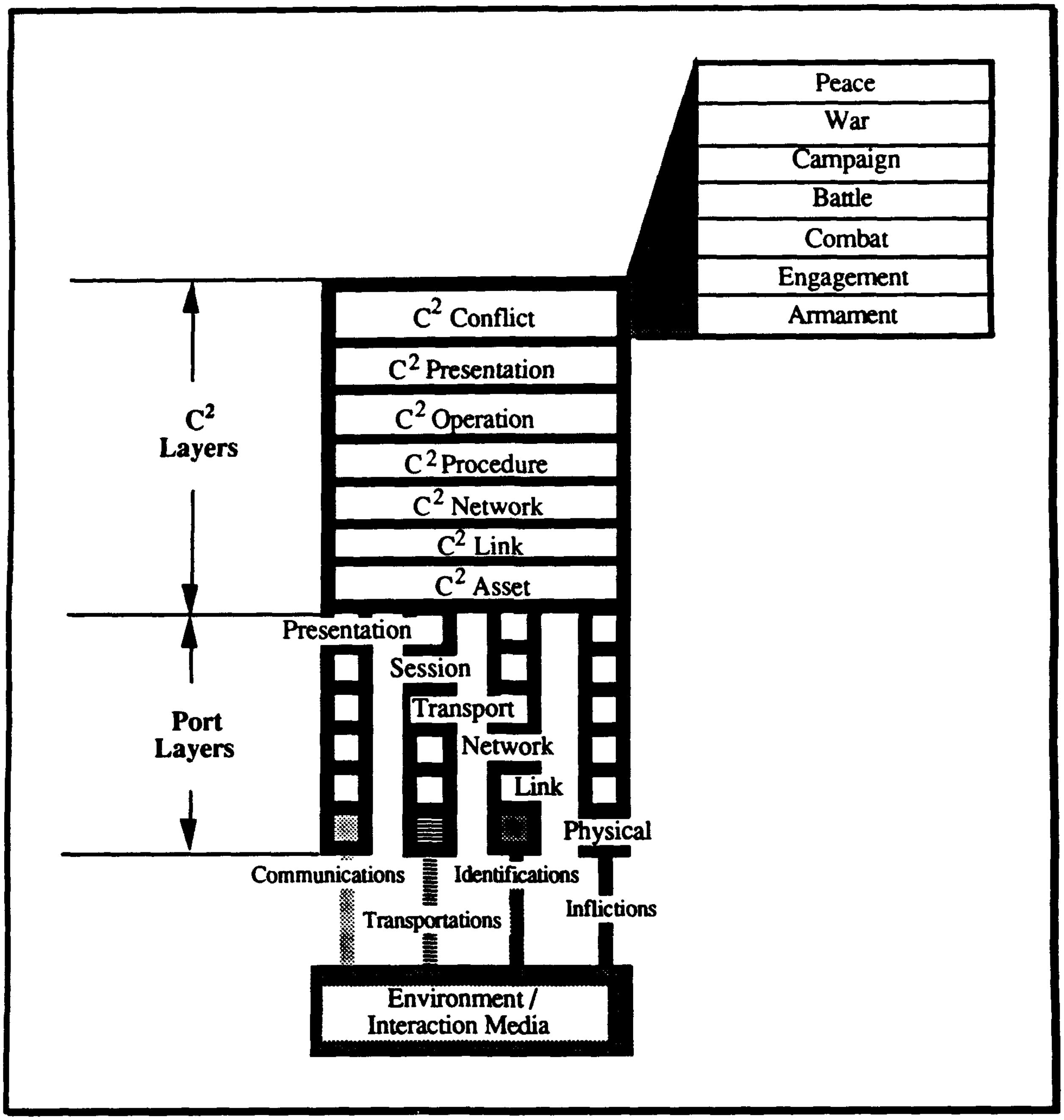

Figure 21. The C^2 application and port layers of a resource

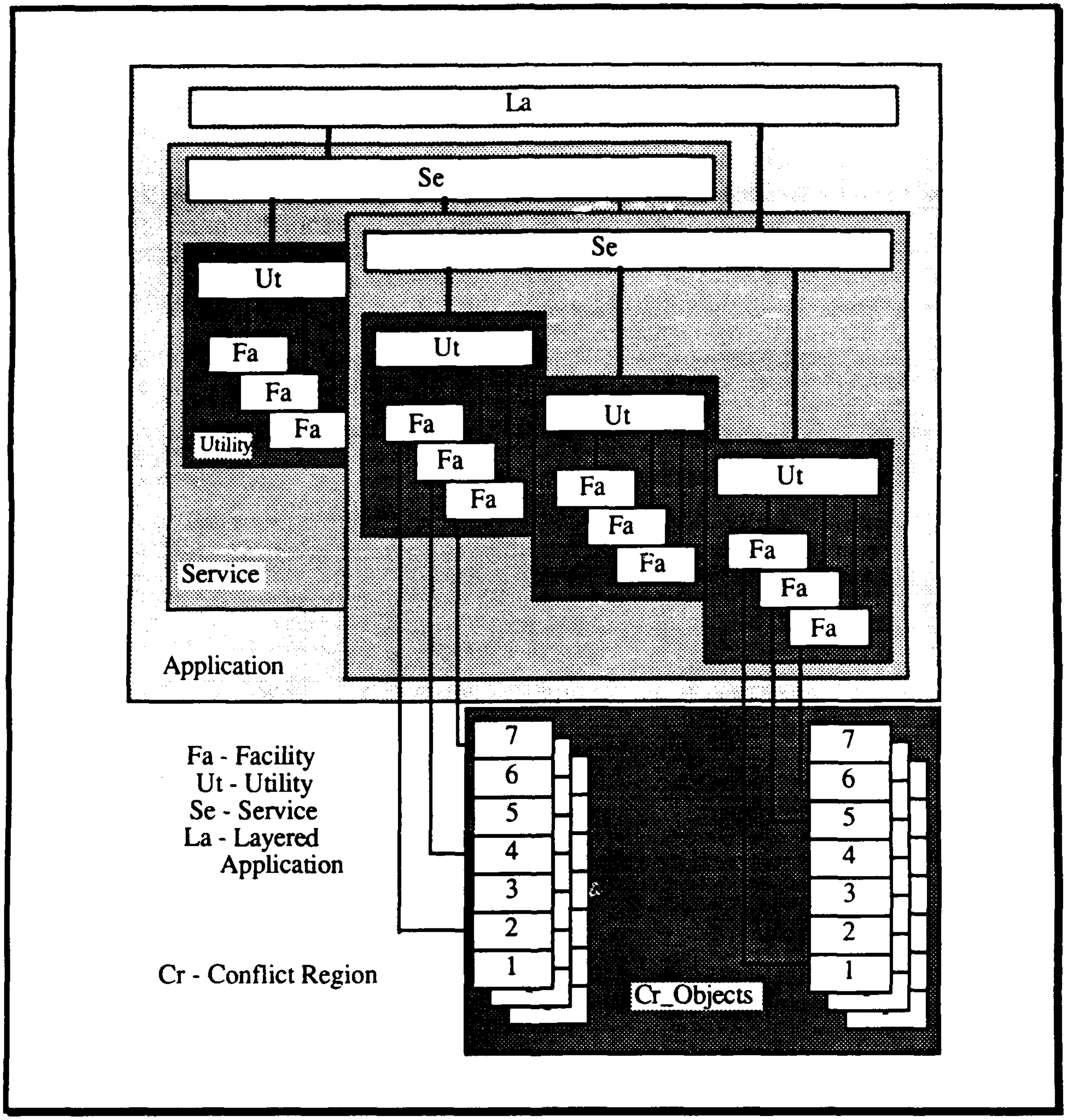

Figure 22. C²ed objects in a given C² application layer

C^2 LAYERS

An Overview

The C^2 layer of a C^2 system consists of seven layers, each of which is involved in peer-to-peer level of relationships or interactions across resources, as illustrated in Figure 23. Thus the ISO OSI RM Application layer is inherently a sublayer of the **C^2 Asset** layer. For a given echelon, the C^2 layer may be distributed to resources at lower echelons in addition to being nested in the lower echelons since lower echelons may provide more specificity and refinement of common goals rather than independent goals which may conflict. In reality, nesting of C^2 can be minimized but not avoided. This may lead to conflicts between local and more global C^2. The role of global C^2 therefore is to minimize such conflicts where possible. At each echelon of command, the highest level of service available to the user is the capability to select the resources necessary to carry out the commander's mission and the establishment of an understanding of the commander's mission by the resources selected to carry it out. The C^2 Conflict layer is the highest sublayer of the C^2 layer. It provides for formulating the mission and communicating the commander's intent, guidance and leadership. The lower layers of the C^2 layer support the C^2 Conflict layer with more specificity and detail of how future missions will be accomplished and how well current missions are being executed.

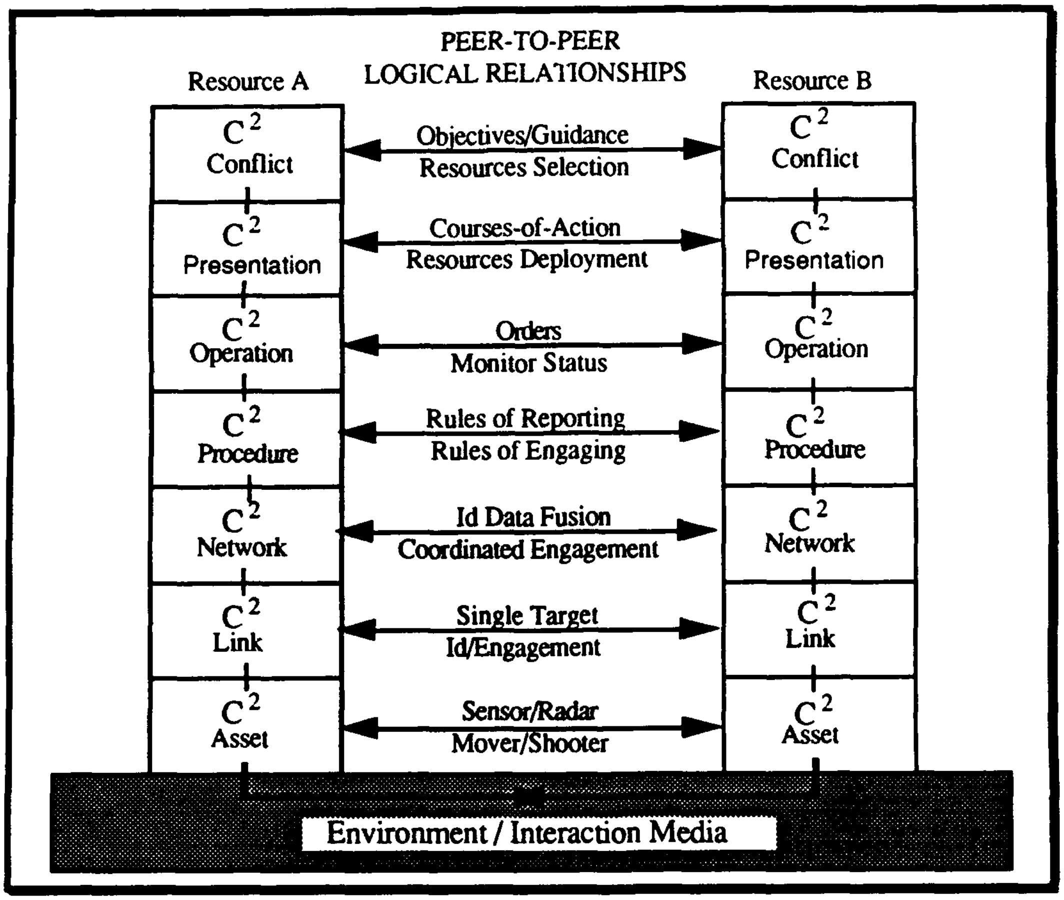

Figure 23. Representative peer layer interactions for C^2 applications

Layer Responsibility

Reflecting the views of many commanders, every resource should have the same view of the universe as the commander. This notion, which is key to promoting greater morale and motivation, implies that ideally, under perfect communications, every resource should be fully aware of the mission in which it plays a part. Thus every new mission may result in a set of peer layer interactions between resources as shown in Figure 23. Once the C^2 Conflict layer has coordinated its mission, it is served by planning functions responsible for the organization and deployment of the selected resources in the best possible posture. The C^2 Presentation layer is responsible for the preparation, coordination, approval and dissemination of the plans in support of the mission. To ensure proper implementation of the plans, the C^2 Presentation layer is serviced by the C^2 Operation layer. The C^2 Operation layer initiates control functions which generate orders to the committed resources and monitors their status. Orders are given in broad descriptions of the intended outcome of sub-missions. The C^2 Operation layer is serviced by the C^2 Procedure layer which implements the doctrine concerning rules-of-reporting status and the rules-of-engagements of targets. C^2 Procedures are detailed prescriptions of various C^2 techniques which may be selected for implementing orders by fully networked resources. The C^2 Procedure layer is served by the C^2 Network layer which provides for the synergistic effect of fusing status, observations, intelligence and communications data to achieve combined, coordinated weapons engagements and maneuver of resources across the theater of operation. The C^2 Network layer is supported by the C^2 Link layer which ensures the reliability of individual pair-wise resource-resource interactions whether: a) friendly-friendly, one-with-one, e.g., sensor-sensor, sensor-weapon, weapon-weapon, or transmitter-receiver, or b) friendly-enemy, one-against-one, e.g., sensor-target, weapon-target, and other such warfare interactions. Finally, the C^2 Link layer is dependent upon the performance of the physical assets in employing mechanical, biological, chemical, nuclear or electromagnetic forces.

Forms of Communications

The communications aspects relevant to C^2 reside in the application layer of the ISO OSI RM. Ultimately, communications relevant to C^2 address and involve the use of all possible means of interactions needed to be ready for, to cope with, or to resolve a given conflict or crisis. Often C^2 is augmented by an additional "C". The third "C" in "C^3" stands for communications of all contextually meaningful information objects necessary to accomplish the mission of the C^2 System. At the physical end of its domain, communications may be carried out through any media in the environment capable of signalling, i.e., forced vibrational behavior propagating conventionally through electronic, photonic, sonic, or audio phenomena and nonconventionally through chemical, biological or nuclear phenomena. At its highest form, it is carried out through the attribute of leadership. Leadership is a quality of motivation which is principally conveyed through communications. The meaning of communications therefore goes far beyond the mere technical aspects of transmitting and receiving bits of information as layered for the communications port (ISO OSI RM Layers 1-6).

Horizontal (peer level) communications between and among (sub)layers represent interoperability of C^2 system resources. Vertical communications represent intraoperability of C^2 system processes within resources. The ISO OSI RM is applicable between any two C^2 (sub)layers, vertically or horizontally, wherever there exists a physical medium for communications. Note that the layered structure depicted in Figure 23 assumes that the C^2 interaction ports as shown in Figure 21 are implicitly available to support the upper layers.

C^2 Application Services

The notion of Service denotes a capability to interface in accordance with a standard protocol. The Army, Air Force, Navy and Marines are called military services because each provides a specific service to the Commander-in-Chief (CINC), i.e., in support of ground, air, sea and amphibious missions. The notion of a Service, therefore, is quite general and applicable at any echelon of C^2 or to any layer or sublayer of a resource of a C^2 system.

A C^2 layer sublayer process typically involves eight classes of Services depicted in Figure 24. The Services provide Utilities and Facilities for three modes of operations Situation Assessment (SA), Product Development (PD), and Execution Monitoring (EM). The services include analyses and syntheses with respect to: a) (N+1)_Requirement and Status Products, b) friendly capabilities essential to support (N+1)_Products, c) adversarial capabilities opposing (N+1)_Products, d) environment associated with the relevant space-time region of conflict, e) key factors in a, b, c, and d which are critical to continue or modify (N)_Products, f) generation of (N)_Product alternatives, g) evaluation of the (N)_Product alternatives, and h) integration, maintenance and dissemination of (N)_Products to applications at (N-1)_layer. Note that each of these Application Services is typically supported by Utilities and Facilities which generate Scenarios, Snapshots, Overlays and Cells for past, present and future situations, as well as for gaming with candidate products to assess potential outcomes. Interactions between Services are highly dependent upon which mode is current and upon the transition from mode to mode. The time period within a given mode is called a phase. Possible configuration of coupling and interdependence of Services is shown in Figure 24a. Services play different roles depending upon the mode and upon the amount of uncertainty associated with their products. As shown, switching between modes increases to a higher intensity level as the level of uncertainty increases. The Services within each layer must be designed with maximum flexibility to accommodate switching between modes for highly fluid situations. Normally, situation assessment, n_SA, is initiated by integrating products using n_EC, n_FC, n_GC, and n_RC Services. Product development, n_PD, usually follows n_SA by integrating products derived from using n_PG and n_PE based upon n_SA inputs. Execution Monitoring, n_EM, then follows n_PD by integrating products derived from n_PR and n_PS. Many other combinations of products may be generated using different Services.

An interface between layers is achieved through Product objects which serve as Service Access Points to mediate between layers. Within each layer, collectively, all the services cooperate to maintain their individual or joint Products. To facilitate and expedite Product Development, applications at a higher layer maintain aggregated (i.e., consolidated, fused, or integrated) objects which represent Products of applications to be developed (in the case of Requirements) or which have been developed (in the case of Status) at a lower layer. Thus as the C^2 process evolves, layered products evolve from past (T/T1) to current (T1/T2) (present) and from current to future (T2/T3) time frames for execution. As shown in Figure 25, each service mode plays an important role in the evolution of C^2 Products. Incremental transitions of C^2 Products follow the causal relationships shown in Table 4.

The following sections present a more detailed description of the Services performed at each C^2 Application layer.

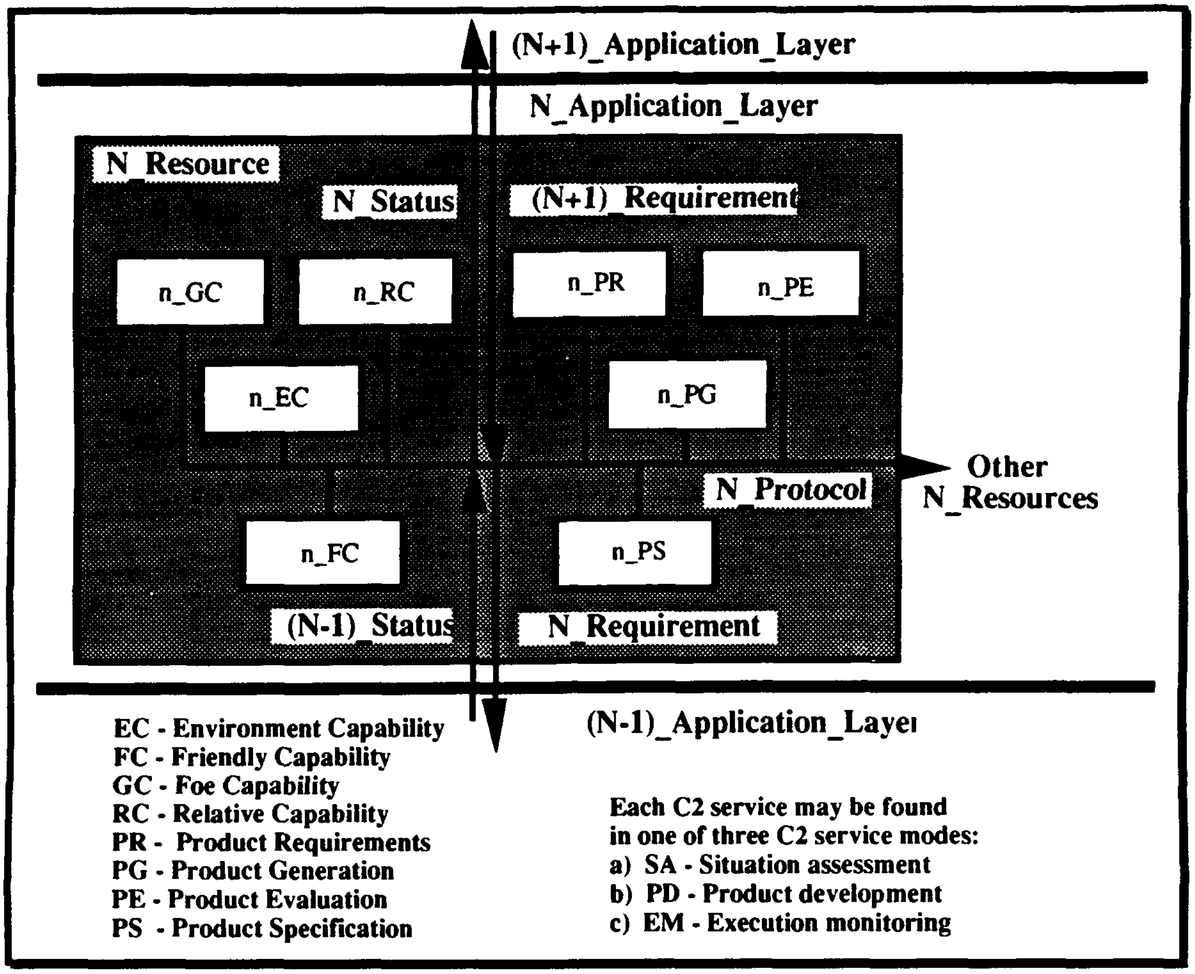

Figure 24. Generic services of a C^2 layer

Table 4. Causal relations for evolution of C^2 Products

(N+1)_requirement(T) + (N)_requirement(T) => (N)_requirement(T1)
(N)_requirement(T) + (N-1)_requirement(T) => (N-1)_requirement(T1)
(N+1)_requirement(T1) + (N)_requirement(T1) => (N)_requirement(T2)
(N)_requirement(T1) + (N-1)_requirement(T1) => (N-1)_requirement(T2)
(N+1)_requirement(T2) + (N)_requirement(T2) => (N)_requirement(T3)
(N)_requirement(T2) + (N-1)_requirement(T2) => (N-1)_requirement(T3)

(N-1)_status(T) + (N)_status(T) => (N)_status(T1)
(N)_status(T) + (N+1)_status(T) => (N+1)_status(T1)
(N-1)_status(T1) + (N)_status(T1) => (N)_status(T2)
(N)_status(T1) + (N+1)_status(T1) => (N+1)_status(T2)
(N-1)_status(T2) + (N)_status(T2) => (N)_status(T3)
(N)_status(T2) + (N+1)_status(T2) => (N+1)_status(T3)

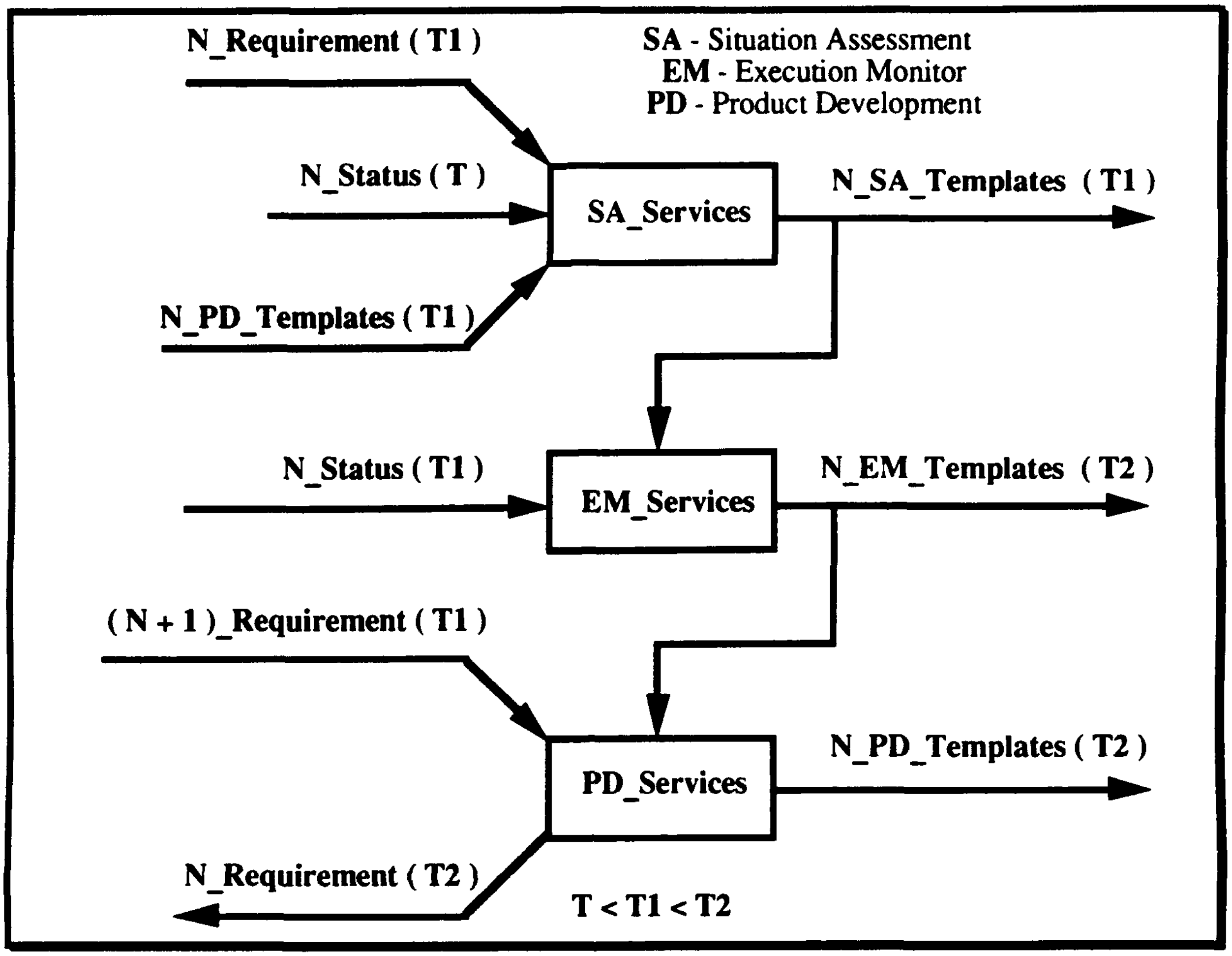

Figure 25. Time phasing C² services between SA, EM, and PD r s

C² CONFLICT LAYER

C² Conflict layer services provide for the creation, management, supervision and execution of Missions to terminate or initiate, minimize or maximize, contain or sustain, escalate or diffuse dynamically evolving conflicts. The C² Conflict layer generates, updates and monitors Missions to ensure that its accomplishment will ultimately end the conflict as intended, perceived, constrained and bounded by the commander. The C² Conflict layer ensures that: a) the mission is communicated and understood by all resources of the system to the maximum extent possible, b) the appropriate resources can be dedicated, committed, activated or placed in reserve and are adequately supplied and equipped to carry out the mission-related interactions, and c) the plans developed by the presentation layer cover all aspects of the mission in an integrated fashion. The mission includes the full definition of success, failure or stalemate. The services provide for the general allocation of resources to perform the mission plans and tasks as well as for the demonstration of leadership and dedication of the resources to the system. The main products of this layer are missions. The Commander is the responsible Official for developing a Mission. The Commander is guided by Policies to define Missions. Note that a formal name is associated with the responsibilities of the Commander at each layer of Conflict. These names are Keywords of the C²RM and, as other keywords, should not be confused with their operational counterparts. The responsibilities of an operational commander may span any subset or superset of the responsibilities identified herein. Given a Mission and associated Intent, The problem of executing the Mission is then delegated to subordinate intelligent Planners responsible for the C² Presentation layer services. Admissible Missions and subsequent layered Products such as Plans and

Tasks must take into account the potential reaction of the adversary involved in the Conflict. Thus C^2 Conflict layer services provide the capability to establish and analyze a set of goals and countergoals in a nested tree fashion and in a manner consistent with the general principles of conflict as postulated in Table 5. Such principles should be appropriate to any echelon of C^2 and may be derived from the principles of war (also shown in Table 5)[14]

Table 5. Principles of Conflict

Action	State	(Principle of War)
1) **Unify**	responsibility	(Unity of Command)
2) **Seize**	initiative	(Surprise)
3) **Reduce**	vulnerability	(Security)
4) **Maintain**	progress	(Offensive)
5) **State**	clearly missions	(Objective)
6) **Simplify**	plans and tasks	(Simplicity)
7) **Improve**	engagement posture	(Maneuver)
8) **Concentrate**	armament	(Mass)
9) **Economize**	resources, assets and supplies	(Economy of Force)

Sublayer Structure of C^2 Conflict Layer

The conceptual layering of adversarial relationships is shown in Figure 26. Each of the conflict sublayers provides services to further define and refine any potential conflict and its associated space-time boundary. The layers follow directly from the hierarchy of the space-time regions shown in Figure 9. C^2 systems must be operational in peace and ready for war. War characterizes a space-time region of conflict which is layered hierarchically and nested to serve the general mission of peace. Any space-time region of peace which is threatened becomes a region of concern subject to war. Threats may be realized by the prospects of takeover, destruction, disruption, damage and degradation of living conditions resulting from the imposition of general constraints upon the freedom and survivability of the C^2 system. From the perspective of the C^2 Conflict layer, war is undertaken to achieve or regain peace in a limited or broadened environment. The mission must also describe any lower layer of conflict in which the C^2 system is to be involved. Ultimately, the importance of accomplishing the mission may be motivated as necessary to bring peace, for the short-term or the long-term. Each C^2 system must define its own peace and related conflicts in which it may become involved. Once a conflict is identified it may be resolved or dissolved at any layer. Resources may be positioned in various states of readiness corresponding to each layer of conflict. The reaction of friendly, neutral and adversarial C^2 systems will depend to a large extent on observing and analyzing the state of readiness at each layer of conflict. Thus a resource may be war-ready but not campaign-ready. Another resource may be combat-ready but not engagement-ready. The ability to diffuse or eliminate a conflict at any layer above the armament layer is called deterrence. If a resource is said to be ready at a lower layer of conflict, it is automatically ready at any of the higher layers. In the general case, all layers of conflict are needed to characterize a conflict. A conflict cannot be characterized without peace. In various simplifying cases, however, a higher layer of conflict may be degenerate with a lower layer of conflict. The perception of peace and the threat of a conflict are relative to each C^2 system. Different C^2 systems may have different perceptions. Regardless of the relative perceptions, the generic characterization of the layer of peace and supportive layers of conflict is common to all C^2 systems.

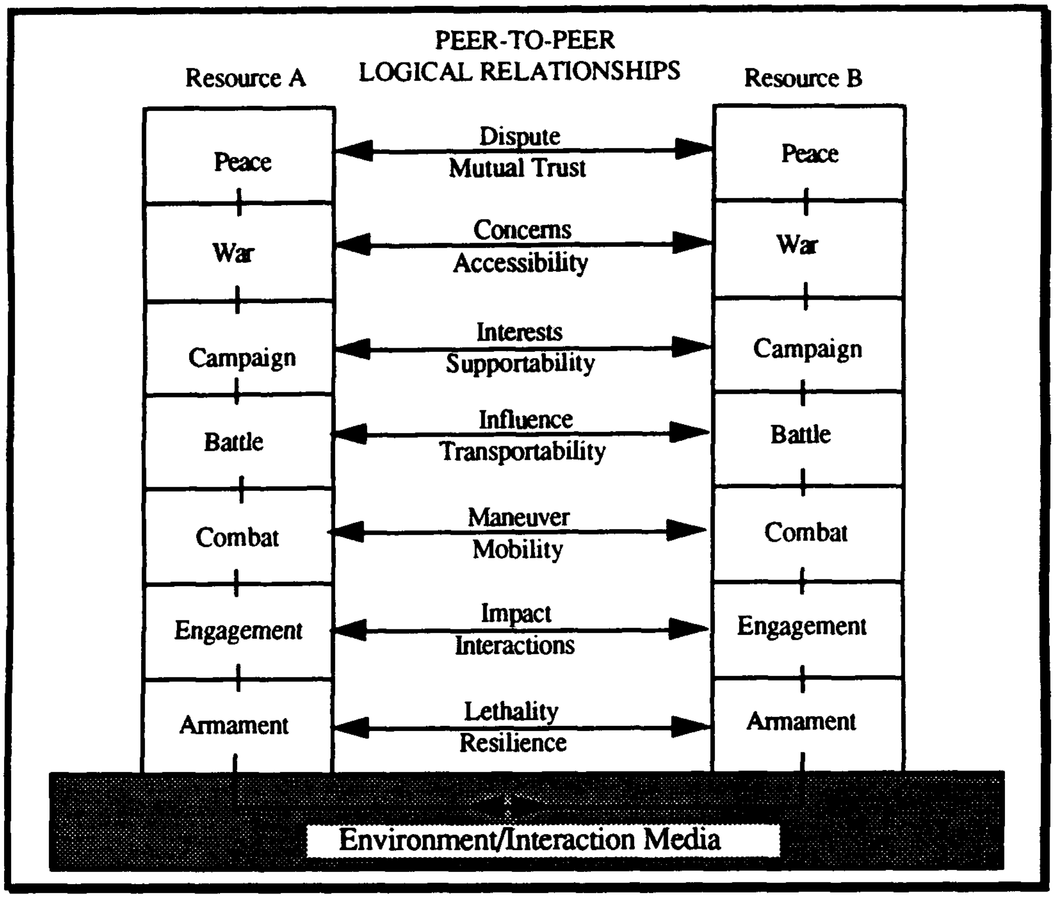

Figure 26. Logical interactions of the C^2 conflict layers

Each layer of conflict may be staged and phased in time. Staging involves decisions with respect to allocation, scheduling, and positioning of resources for that layer of conflict. Phasing involves windows of opportunities for synchronization and execution associated with that level of conflict. Once a phase is initiated it is allowed to run its course. Phases are determined by boundary conditions of time and space where interactions are clearly defined or assumed to be known. Threshold conditions of status are monitored during each phase to determine whether or not to go on to the next phase or to start a new Stage. Each layer of conflict must make decisions with respect to commitment of Reserve and Resupply available at Conflict layer K for layer K-1. In addition, each layer of conflict must make decisions with respect to reprioritizing, reallocation, rescheduling, repositioning and resynchronizing Support, Reinforcement, and Supply. Each conflict may also be characterized by its intensity. The intensity of the conflict is characterized by the number of resources which are made ready to interact in the conflict at each layer as well as the impact of lethality which may be incurred. Resources transition from one layer of conflict into another as a result of physical and associated logical interactions. Note that the layered structure depicted in Figure 26 assumes that the subordinate C^2 layers as shown in Figure 21 are implicitly available to support the conflict layers.

A C^2 Conflict layer process involves the following eight types of SA, PD, and EM services: a) Mission goal and concept definition statement, b) friendly capabilities essential to support the goal, c) adversarial capabilities opposing the goal, d) environment associated with the goal, e) key factors in a, b, c, and d which are critical to continue or modify the current mission, f) generation of evolving missions, g) evaluation of evolving missions, h) dissemination of modified or new missions.

Cr_conflict_layer objects

In general every conflict will have a Conflict_identification (ID) to be associated with a scenario and a given but broad set of issues. More specifically, each layer of conflict will also have an associated distinguishing name followed by its 'parent' and 'child' conflict layers. Thus one may generate a common template for any Cr_conflict layer K object as follows:

Confict layer K_ID
- Confict layer (K +1)_ID
- Confict layer (K -1)_ID

Confict layer K_name (Who)
Confict layer K_explanation (Why)
Confict layer K_description (What)
Confict layer K_coordinates (Where/When)
Confict layer K_status/capability (How)

PEACE LAYER

The peace layer is responsible for all missions essential to prevent, reduce, or eliminate conflicts of all types over the entire space-time region under the purview of the C² system. In particular, the peace layer is responsible for all missions to deter, preclude or prevent potential transitions or escalation of conflicts from the civil domain and methods of resolutions to military confrontations in a space-time region with a potential for adversarial relationships. Such a region is also called a region of peace. Peace-class missions are the responsibility of the President. The President is a leader who must be able to recognize and identify the presence of current, potential, and emerging conflicts. The President must seek to contain, reduce, resolve or eliminate conflicts and generate clear missions to achieve such goals. As a last resort, peace-class missions will identify when to enter into a militant conflict. The President may decide to initiate war when peaceful means fail to deter potential adversary resources from launching into a militant posture. C² resources are said to be in peace if and only if they coexist with mutual trust and support. The products of this layer are peace missions.

Cr_peace_layer objects

Cr_peace_layer objects are responsible for infrastructure **acquisition** features which support the **creation, construction** and **production** capabilities of manmade Cr objects (e.g., units, assets, ports, packages, roads, bridges, and boundaries). Acquisition features, therefore, include training of personnel and manufacturing of materiel and their readiness for integration into force packages (e.g., Organizations & Equipments) appropriate to potential war regions. For example, desert, mountain or jungle warfare would require different ground warfare force packages. In addition, Cr_peace_layer objects enable Cr_interactions between Cr_unit objects F and G in accordance with Cr_coordination objects FG. The produced Cr objects are essential to provide initial deterrence and to sustain losses anticipated through the Cr_war and lower layers of conflict should Cr_coordination objects FG be violated.

Cr_peace objects are distinguished by a set of Cr_identification - Cr_communication interaction pairs. The purpose of Cr_peace objects is to create Cr_units to support a given war and enable global (national or coalition) Cr_units to prepare for early warning operations with adequate resources. Cr_peace objects enable 'action' flow of global surveillance identification and communication interactions to establish thresholds for creating regional (joint or service component) Cr_units as they are needed in anticipation of a given threat of war. A common template for a Cr_peace_layer object includes:

- Peace layer ID
    - Confict layer ID
    - War layer ID
- Peace layer name (Who)
- Peace layer explanation (Why)
- Peace layer description (What)
- Peace layer coordinates (Where/When)
- Peace layer status/capability (How)

WAR LAYER

The war layer of conflict is responsible for missions to protect the concerns essential for the cooperating resources to win a peace. War-class missions are the responsibility of the General. The General is a leader who may launch a number of campaigns, conducted in parallel or sequence, essential to achieve political pressure through an armed force. The General must be able to assure freedom of access from the space-time regions of peace to the space-time region of campaigns under concern. Mobilization and maintenance of an adequate inventory of supply, equipments, and tools are critical to the success of a war. Campaigns are selected by analyzing their accessibility from the region of concern to the region of interest. The products of this layer are war missions.

Cr_war_layer objects

Cr_war_layer objects are responsible for **mobilization** features which support the **allocation**, **configuration**, and **distribution** capabilities of man_made Cr objects from any peacetime storage area to strategic marshalling, assembly, or staging areas appropriate to established campaign regions against a potential threat. Mobilization features, therefore, include integration of personnel and materiel into campaign_ready (i.e., deployable) Cr_units aka campaign packages (e.g., Task Organization or Task Forces). Each campaign package is associated with its own set of Cr_coordination objects. The mobilized Cr objects are essential to provide unquestioned deterrence and to sustain losses anticipated through the Cr_campaign and lower layers of conflict should Cr_coordination objects FG be violated.

Cr_war objects are distinguished by a set of Cr_identification - Cr_communication interaction pairs. The purpose of Cr_war objects is to allocate Cr_units to support a given campaign and enable Cr_units to prepare for regional operations with adequate resources. Cr_war objects enable 'action' flow of global surveillance and communication assets to support mobilized Cr_units as they are committed to and enter a given campaign. A common template for a Cr_war_layer object includes:

War layer ID
- Peace layer ID
- Campaign layer ID

War layer name (Who)
War layer explanation (Why)
War layer description (What)
War layer coordinates (Where/When)
War layer status/capability (How)

CAMPAIGN LAYER

The campaign layer of conflict is responsible for missions to secure interests to win a war. Campaign-class missions are the responsibility of the Director. The Director is a leader who may launch a number of battles, conducted in parallel or in sequence, essential to control and influence a broad region. The Director must secure the access from the space-time region of interest to the space-time region of influence to be occupied. Pre-positioning and sustaining supplies to be made available for ensuing and ongoing battles are critical to the success of a campaign. Battles are selected by analyzing their supportability in the space-time region of interest from the space-time region of influence. The products of this layer are campaign missions.

Cr_campaign_layer objects

Cr_campaign_layer objects are responsible for **deployment** features which support **replenishment**, **reinforcement**, or **projection** (sortie/vector) capabilities of man_made Cr objects from any strategic staging area to tactical assembly areas appropriate to established battle regions. Deployment features, therefore, include reinforcing or reconstituting Cr_units from available or residual Cr_units to support, create and position battle_ready Cr_units aka battle packages including combat, combat support and combat service support capabilities. Each battle package is associated with its own set of Cr_coordination objects. The deployed Cr objects are essential to provide imminent deterrence and to sustain losses anticipated through the Cr_battle and lower layers of conflict should Cr_coordination objects FG be violated.

Cr_campaign objects are distinguished by a set of Cr_identification - Cr_transportation interaction pairs. The purpose of Cr_campaign objects is to enable Cr_units to conduct timely battle operations with adequate resources by enabling 'action' flow of new Cr_units to augment or replace embattled Cr_unit objects as they move from one battle to another. A common template for a Cr_campaign_layer object includes:

- Campaign layer ID
  - War layer ID
  - Battle layer ID
- Campaign layer name (Who)
- Campaign layer explanation (Why)
- Campaign layer description (What)
- Campaign layer coordinates (Where/When)
- Campaign layer status/capability (How)

BATTLE LAYER

The battle layer of conflict is responsible for missions to achieve influence to win a campaign. Battle-class missions are the responsibility of the Manager. The Manager is a leader who may launch a number of combats, conducted in parallel or in sequence, to exploit and occupy certain key areas. The manager must support the access from the space-time region of influence to the space-time regions of maneuverability. Resupplying and reinforcing combat-ready resources are critical to the success of a battle. Combats are selected by analyzing the transportability of combat resources from the space-time region of influence to the space-time region of maneuverability. The products of this layer are battle missions.

Cr_battle_layer objects

Cr_battle_layer objects are responsible for **sustainment** features which support the **maintenance**, **supply,** or **replacement** capabilities of man_made Cr objects from any tactical assembly area to tactical containment areas appropriate to established combat regions. Sustainment features, therefore, include the tactical distribution of logistics and reserve allocations and their synchronization to meet the needs of combat_ready Cr_units in the forward areas. Each combat package is associated with its own set of Cr_coordination objects. The sustained Cr objects are essential to provide final deterrence and to sustain losses anticipated through the Cr_combat and lower layers of conflict should Cr_coordination objects FG be violated.

Cr_battle objects are distinguished by a set of Cr_communication - Cr_transportation interaction pairs. The purpose of Cr_battle objects is to enable Cr_units to conduct continuous combat and combat_support operations by enabling 'action' flow of logistics cargo to meet dynamic demands by consuming and dispensing Cr_unit objects as they move from a current location to a more advantageous location for maneuver. A common template for a Cr_battle_layer object includes:

- Battle layer ID
  - Campaign layer ID
  - Combat layer ID
- Battle layer Name (Who)
- Battle layer explanation (Why)
- Battle layer description (What)
- Battle layer coordinates (Where/When)
- Battle layer status/capability (How)

COMBAT LAYER

The combat layer of conflict is responsible for missions to maneuver to win a battle. Combat-class missions are the responsibility of the Captain. The Captain is a leader who may initiate a number of engagements against multiple targets, conducted in parallel or in sequence. The Captain must be able to maneuver and change force-to-target assignments throughout the entire duration of any given battle.The mobility of engagement-ready resources is critical to the success of combat. Engagements are selected by analyzing the mobility of engagement-capable resources from the space-time region of maneuverability to the space-time region of interactions. The products of this layer are combat missions.

Cr_combat_layer objects

Cr_combat_layer objects are responsible for **containment** or **confrontation** features which support the **movement**, **maneuver**, or **emplacement** capabilities of man_made Cr objects from any tactical containment region to the tactical employment (e.g., meeting engagement) areas appropriate to established engagement regions. Containment features, therefore, include tactical vectoring and positioning of combat and combat support resources and their synchronization to meet the needs of engagement_ready Cr_units in the forward areas. Each engagement package is associated with its own set of Cr_coordination objects. The containing Cr objects are essential to provide subsequent follow-on force deterrence and to withstand losses anticipated through Cr_engagement and Cr_armament layers of conflict should Cr_coordination objects FG be violated.

Cr_combat objects are distinguished by a set of Cr_identification - Cr_transportation interaction pairs. The purpose of Cr_combat objects is to enable Cr_units to approach, cover, launch and adjust the 'action' flow of self-propelled Cr_unit objects from a current location to a more advantageous location for Cr_identification and Cr_infliction. A common template for a Cr_combat_layer object includes:

- Combat layer ID
    - Battle layer ID
    - Engagement layer ID
- Combat layer name (Who)
- Combat layer explanation (Why)
- Combat layer description (What)
- Combat layer coordinates (Where/When)
- Combat layer status/capability (How)

ENGAGEMENT LAYER

The engagement layer of conflict is responsible for missions to win a combat. Engagement class missions are the responsibility of the Partner. The Partner is a leader who may launch a number of armaments to be delivered in parallel or in sequence against individual or aggregated targets. The Partner must be able to apply the forces available against the targets. A single engagement is relative to a single target. Engagements may be composed by the superposition of fundamental topologies such as shown in Figure 27. Note that two types of fundamental interactions, identification and infliction, are required to characterize the simplest class of engagement, i.e., direct engagement as shown in Figure 27a. Direct engagements may be supported by direct support or general support as shown in Figures 27b and 27e, respectively. There is an implicit distinction between general support and direct support engagements. During simultaneous engagements of multiple targets, targets subject to direct support engagements may or may not be subject to general support engagements depending upon the availability of general support resources which span a much wider region of space for engagement. The support of a direct engagement may be subsequently reinforced as shown in Figure 27f. The support of a direct engagement and its reinforcement requires an additional class of fundamental interactions, i.e., communication, to command, control, and coordinate the more complex engagements. The effective range of sensors and the delivery range of armaments are critical to the success of an engagement. Armaments are selected by analyzing the target as a threat potential and the lethality of armament-capable resources as projected from the space-time region of interaction to the space-time region of impact. The products of this layer are engagement missions. Engagement Missions are issued in accordance with Rules_of_Engagements which provide for self defense, matching hostile criteria to a given situation and compliance with Mission_Control guidance.

Cr_engagement_layer objects

Cr_engagement_layer objects are responsible for **employment** features which support **fire**, **detonation**, or **armament** capabilities of man_made Cr objects from any tactical employment area to *the fire areas appropriate to established armament regions.* Employment features, therefore, include tactical distribution of combat and combat support fire power and their synchronization to meet the needs of armament_ready Cr_units in the interception areas. Each armament package is associated with its own set of Cr_coordination objects. The employed Cr objects are essential to provide subsequent deterrence and withstand losses anticipated through the final Cr_armament layer of conflict should Cr_coordination_ objects FG be violated.

Cr_engagement objects are distinguished by a set of Cr_identification - Cr_infliction interaction pairs. The purpose of Cr_ engagement objects is to enable Cr_units to aim, launch, and adjust the 'action' flow of Cr_ordnance from source Cr_unit objects at effective hitting locations to target Cr objects of opposing forces. A common template for a Cr_engagement_layer object includes:

- Engagement layer ID
  - Combat layer ID
  - Armament layer ID
- Engagement layer name (Who)
- Engagement layer explanation (Why)
- Engagement layer description (What)
- Engagement layer coordinates (Where/When)
- Engagement layer status/capability (How)

ARMAMENT LAYER

The armament layer of conflict is responsible for missions to inflict physical and logical damage, permanent or temporary, to win an engagement. Armament class missions are the responsibility of the Expert. The Expert is a leader who can apply available armaments against individual or aggregated targets. The Expert must be capable of activating armaments. At this layer of conflict, actual power is authorized to be unleashed to maximize the effectiveness of an engagement. Any number of armament types may be launched against a single target, depending upon its survivability. The accuracy of sighting and delivery as well as the lethality of the impact are critical to the success of an armament. The destructive or disruptive envelope of the armament defines the space-time region of damage or degradation, respectively. The space-time region of damage is a subset of the space-time region of impact. The products of this layer are armament missions.

Cr_armament_layer object

Cr_armament_layer objects are responsible for **impairment** features which support the **destruction**, **degradation**, **disruption**, **survivability**, **protection**, or **incapacitation** capabilities of man_made Cr objects from any tactical fire area to damage and contamination areas appropriate to established impact regions. Impairment features, therefore, include tactical distribution of explosive damage effects and damage category (e.g., personnel, equipment, logistic, mobility, or electronic) to meet the needs of attrition for future deterrence. Each impairment package will result in a modified set of Cr_coordination objects. The impaired Cr objects are essential to provide deterrence at all layers of conflict should new Cr_coordination objects FG be violated in the future.

Cr_armament objects are distinguished by a set of Cr_ordnance objects. The purpose of the Cr_armament objects is to enable 'event' flow of Cr_ordnance to penetrate protective barriers, explode to impair, damage, suppress, destroy, impact or neutralize intercepted target Cr_unit objects hit from aiming locations of engagement. A common template for a Cr_armament_layer object includes:

- Armament layer ID
  - Engagement layer ID
  - Presentation layer ID
- Armament layer name (Who)
- Armament layer explanation (Why)
- Armament layer description (What)
- Armament layer coordinates (Where/When)
- Armament layer status/capability (How)

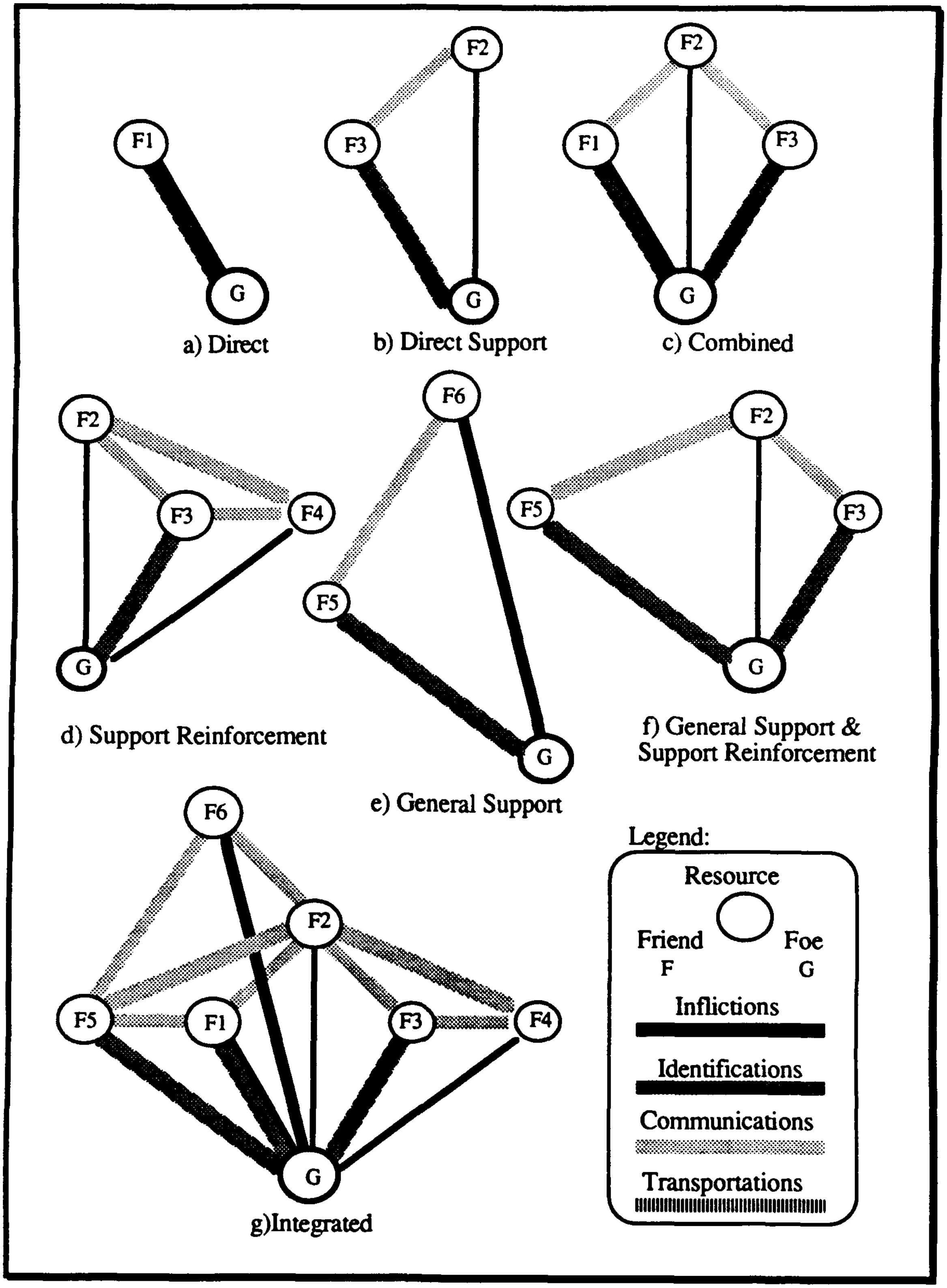

Figure 27. Fundamental types of engagements

C² PRESENTATION LAYER

Presentation services provide for the management, supervision and execution of plans. Presentation services must provide the capability to analyze and understand a given conflict and corresponding missions and, consequently, to draft, coordinate, and finalize their product in support of various potential missions of contingent conflicts. Planning consists of two phases. During the first phase, allocation and adequacy of existing resources, their assets and organizational structures are assessed. Presentation services may reorganize subordinate resources, their distribution of assets, and justify the need for any changes in the quantity and quality thereof. The products from such services are called organization plans. During the second phase, plans are made for a coordinated sequence of operations within the established organizational structures. Such plans are known as operations plans. An Operation plan sets schedule constraints to achieve key milestones within a given mission. It also provides guidance for maneuver and thresholds for expenditures of resources and their assets. The main products of this layer are called plans. The planner is the responsible official for developing a plan. The planner is guided by strategies to generate and evaluate plans. The problem of executing the plan is delegated to subordinate intelligent controllers responsible for the Operation layer services.

A Presentation layer process involves the following eight types of SA, PD, and EM services: a) Mission requirements definition statement, b) friendly capabilities essential to support the mission, c) adversarial capabilities opposing the mission, d) environment associated with the mission, e) key factors in a, b, c, and d which are critical to continue or modify the current plans, f) generation of evolving plans, g) evaluation of evolving plans, h) dissemination of modified or new plans.

C² OPERATION LAYER

Operation services provide for the management, supervision and execution of staff functions to generate clear and concise orders to a wide variety of subordinate resources and assets. Operation services must provide the capability to analyze and understand a given mission and corresponding plans and, consequently, to draft, coordinate, and finalize their product in support of various mission and plan options. Orders are generated and updated to select appropriate procedures and synchronize their execution sequentially or in parallel. Operation services are highly heuristic and typically call for the initialization of concentration or relief of effort by available resources at given times and places. They may also undertake to adapt, modify or rehearse any existing set of procedures. Operational and higher level services evolve directly as a result of the mission in a particular scenario. The main products of this layer are called tasks. The controller is the responsible official for developing a task. A controller uses / consider. tactics to generate and evaluate tasks. Tasks may be issued to prepare for an imminent / subsequent mission, plan or task or to provide a correction / update to a current task. The problem of executing the tasks is delegated to subordinate intelligent agents responsible for the C² Procedure layer services.

An Operation layer process involves the following eight types of SA, PD, and EM services: a) Plan requirements definition statement, b) friendly capabilities essential to support the plan, c) adversarial capabilities opposing the plan, d) environment associated with the plan, e) key factors in a, b, c, and d which are critical to continue or modify the current tasks, f) generation of evolving tasks, g) evaluation of evolving tasks, h) dissemination of modified or new tasks.

Controllers are becoming increasingly more sophisticated and intelligent as high performance computers are exploited to augment the process of command and control. The purpose of a controller is to guide the evolution of system state variables along a desired trajectory or path of execution as defined by a plan which delineates the overall system performance requirements. To carry out a plan, controllers initially generate and evaluate alternative tasks. Subsequently, they should be able to generate desired corrections and updates to the selected task handled by their agents. The Controller's Product may also be fed as guidance to a lower level network of Resources which involve the

Administrators. Intelligent Controllers must be capable of estimation and prediction appropriate to their level of instantiation and commensurate with the level of intelligence of their subordinates. Operating in an environment with stochastic disturbances and time constraints, intelligent Controllers utilize estimation and investigate predictions to produce relevant task descriptions in a dynamic fashion so that output from tasked Resources will follow a given Plan in a robust and timely fashion.

An intelligent controller is typically challenged first with being able to interpret a given plan in terms of permissible and feasible operations and procedures and then to predict resource outputs given a stochastic subset of parameters of uncertain values, i.e., at each time instant a different combination of stochastically varying parameters may be made available. In addition, intelligent controllers may be called upon to make estimates of key parameters based upon incomplete and statistically deficient observations. In prediction, the prime focus is on the next event or set of events which may be produced by a given set of parameters and their values. In estimation, the prime focus is on extracting parameters and their values which are most likely to have produced a given observation of a sequence of events. There is a very close connection between the general problem of estimation and the general problem of prediction. The similarity in the statement of such problems should enable the development and exploitation of common algorithmic tools to support control services involving both aspects in their tactics. For example, traditional Kalman filters or enhancements thereof may be augmented by AI techniques for combination of information to allow a wide range of applications.

A resource may have states which are uncontrollable and/or unobservable from the point of view of a given controller. However, these states, which may be irrelevant to the given controller, may not be ignored in the overall execution of the resource and should be observed and controlled by higher, lower or peer level controllers. Consider a vehicle controller, for example. One controller may be put in charge of monitoring the course and steering. Another controller may be put in charge of monitoring the fuel level and refueling. Clearly, some mechanism must be established to enable cooperation between the two. Any conflict between two or more controllers must be arbitrated by the planner. An intelligent controller must be responsive to an intelligent planner. Any conflict between peer-level intelligent controllers which may evolve in the course of developing or executing tasks will be resolved through negotiation or brought to the attention of the planner for mediation or arbitration. An intelligent controller will be able to anticipate potential conflicts ahead of task execution time.

C^2 PROCEDURE LAYER

Procedure services manage, supervise and execute well-defined, pre-established procedures to implement orders derived by operational services. Procedure services must provide the capability to analyze and understand a given plan and corresponding tasks and, consequently, to draft, coordinate, and finalize their product in support of various plan and task options. Procedural services may be established scientifically through theoretical reasoning and experimental calibrations. The same procedural services may be called upon to support operations derived from a wide range of missions. For example, procedures include all well-defined tactical maneuvers, target reporting and weapon engagement selection disciplines which are incorporated in training as part of doctrine. Any improvisation in procedure should be well-coordinated to ensure that potential conflicts are avoided or minimized. Any well-rehearsed operation may be instituted at the procedure layer. Procedure services are generally scenario independent and allow for the internetting of complementary resources to bridge among transportation networks, communication networks, logistic networks, sensor networks, weapon networks and other types of network assets. Procedural and lower level services are generally known and anticipated. They are prepared well before a specific mission is generated for a given scenario. The main products of this layer are jobs. The agent is the responsible official for developing a job. An agent uses schemata to generate and evaluate jobs. The problem of executing the jobs is delegated to subordinate intelligent administrators responsible for the C^2 Network layer services.

A Procedure layer process involves the following eight types of SA, PD, and EM services: a) Task requirements definition statement, b) friendly capabilities essential to support the evolving task, c) adversarial capabilities opposing successful completion of task, d) environment associated with the task, e) key factors in a, b, c, and d which are critical to continue or modify the current job, f) generation of evolving jobs, g) evaluation of evolving jobs, h) dissemination of modified or new jobs.

C^2 NETWORK LAYER

Network services manage, supervise and execute multilateral, collective interaction of supplementary resources (with similar assets) to create a synergistic effect when compared with individual and link level independent services. Network services must provide the capability to analyze and understand a given task and corresponding jobs and, consequently, to draft, coordinate, and finalize their product in support of various task and job options. Network services include any network of interacting resources such as communications networks, transportations networks for logistics and maneuver, sensor networks, intelligence networks, weapon networks and navigation networks. Network services ensure that end-to-end procedures are supported by participating resources. The main products of this layer are Assignments. The administrator is the responsible official for developing an assignment. An administrator uses disciplines to generate and evaluate assignments. The problem of executing the assignments is delegated to subordinate intelligent coordinators responsible for the C^2 Link layer services.

A Network layer process involves the following eight types of SA, PD, and EM services: a) Job requirements definition statement, b) friendly capabilities essential to support the evolving job, c) adversarial capabilities opposing successful completion of job, d) environment associated with the job, e) key factors in a, b, c, and d which are critical to continue or modify the current assignments, f) generation of evolving assignments, g) evaluation of evolving assignments, h) dissemination of modified or new assignments.

C^2 LINK LAYER

Link services manage, supervise and execute one-on-one and one-on-many interactions. Link services must provide the capability to analyze and understand a given job and corresponding assignments and, consequently, to draft, coordinate, and finalize their product in support of various job and assignment options. Services include communications links, transportation links, logistics links, sensors links, weapons links, sensor-target detections, weapon-target engagements, sensor-weapon coordination and relative navigation. Link interaction response times, throughputs, errors, failures and recovery are of primary concerns. The main products of this layer are transactions. The coordinator is the responsible official for developing a transaction. A coordinator uses techniques to generate and evaluate transactions. The problem of executing the transactions is delegated to subordinate intelligent operators responsible for the C^2 Asset layer services.

A Link layer process involves the following eight types of SA, PD, and EM services: a) Assignment requirements definition statement, b) friendly capabilities essential to support the evolving Assignments, c) adversarial capabilities opposing successful completion of Assignments, d) environment associated with the Assignments, e) key factors in a, b, c, and d which are critical to continue or modify the current Transactions, f) generation of evolving transactions, g) evaluation of evolving transactions, h) dissemination of modified or new transactions.

C^2 ASSET LAYER

Asset services manage, supervise and activate individual assets and associated physical processes within each resource. Asset services must provide the capability to analyze and understand a given assignment and corresponding transactions and, consequently, to draft, coordinate, and finalize their product in support of various assignment and transaction options. Asset services are directly involved with the available types of interaction, i.e., communications, transportations, identifications, or inflictions. They include supplies, equipments, and tools. Supplies include any combination of ammunition, fuel, oils and lubricants, rations, and spare parts. Equipments include any combination of weapons, sensors, communications gear and transportation carriers. Carriers include any class of personnel, vehicle, vessel, air or space craft. Tools include any specialized capabilities embedded in hardware, software or man-machine interface. Asset services must ensure that individual assets attain a high state of preparedness and operational readiness. Services at this level are evaluated with respect to: a) weight, size, power and other compatibility requirements necessary to sustain and interconnect the resources, b) maneuverability and navigation necessary to respond to marching orders, and c) degree of physical destruction, disruption, interruption or damage which may be incurred or imparted in the course of any given interaction. The actual management, supervision, and execution of each individual physical asset is carried out by layers 1 through 6 corresponding to each interaction as shown in Figure 21. The main products of this layer are packages. The operator is the responsible official for developing a package. An operator uses instructions to generate and evaluate packages. The problem of employing the packages is delegated to subordinate intelligent action officers responsible for the interaction-specific Presentation layer services.

An Asset layer process involves the following eight types of SA, PD, and EM services: a) Transaction requirements definition statement, b) friendly capabilities essential to support the evolving Transactions, c) adversarial capabilities opposing successful completion of Transactions, d) environment associated with the Transactions, e) key factors in a, b, c, and d which are critical to continue or modify the current Packages, f) generation of evolving Packages, g) evaluation of evolving Packages, h) dissemination of modified or new Packages.

C^2 asset classes

C^2 asset classes include Weapon (W), Sensor (S), Transceiver (T), and Vehicle (V) classes. Stand-alone reference to an asset is meaningless. Collateral impact to an asset usually affects more than one co-located asset class. C^2 asset classes are only meaningful when matched or related to interact with each other as shown in Table 6.

Table 6. Classes of C² assets

	Asset Subclass	Concrete Asset Object
1)	Weapon-Weapon	Anti-Missile Missile
2)	Sensor-Sensor	Radar Detector, Laser Detector
3)	Transceiver-Transceiver	Modem, Radio, Relay, Router, Bridge
4)	Vehicle-Vehicle	Aircraft Carrier, Transport Plane
5)	Weapon-Sensor	Missile Detector, Incoming Round Detector
6)	Weapon-Transceiver	Fire Control System
7)	Weapon-Vehicle	Self-Propelled Gun, Gun Ship, Battle Ship
8)	Sensor-Weapon	Chaff Dispenser, Laser Gun, Smoke Gun
9)	Sensor-Transceiver	Telemetry System
10)	Sensor-Vehicle	UAV, Scout Helicopter
11)	Vehicle-Weapon	ATM, Torpedo, SAM
12)	Vehicle-Sensor	MTI Radar, Motion Detector
13)	Vehicle-Transceiver	Crew Intercom
14)	Transceiver-Weapon	Jammer
15)	Transceiver-Sensor	Radio Interceptor
16)	Transceiver-Vehicle	Mobile Radio

In general, more complex associations may be made. The order of association of asset classes typically affects the meaning of the interaction as shown below:

((Transceiver-Weapon)-Vehicle)	≠	(Transceiver-(Weapon-Vehicle))
((Sensor-Weapon)-Vehicle)	≠	(Sensor-(Weapon-Vehicle))
((Vehicle-Weapon)-Vehicle)	=	(Vehicle-(Weapon-Vehicle))

AGGREGATION OF CR_INTERACTIONS

A Cr_unit object corresponds to each C^2 object, and a Cr_interaction object (involving pairs of Cr_unit objects and Cr_package objects) corresponds to the four classes of C2_interactions (involving C2_packages) which occur between any two C2_unit objects: C2_identification (involving C2_images), C2_communication (involving C2_messages), C2_transportation (involving self-propelled or transportable C2_cargo) and C2_infliction (involving C2_ordnance). Each C2_interaction has a source and a target C2_unit object. For each C2_unit object and within each C2_layer object, there corresponds a Cr_unit object. For each C2_conflict between any two C^2 unit objects F and G, therefore, there corresponds a Cr_conflict object involving Cr_unit objects F and G. Cr_conflicts may be synthesized using any combination of these interactions. A Cr_conflict may be decomposed into seven layers just as the C^2 conflict layer is decomposed for C2 objects. While the same Cr objects persist across Cr_conflict layers, their view of aggregated features and their interface will most likely be different to correspond to the different responsibilities associated with each of the C^2 conflict layers as described below.

Cr_interactions may also be aggregated into a pair of key interactions. Such a pairwise aggregated interaction may be paired with a third interaction or another pair-wise aggregated interaction to produce the effect of three or more basic interactions. For example, to effect attrition, one may encapsulate identification with infliction. To effect surveillance, one may aggregate transportation and identification. To effect navigation, one may encapsulate identification and communications. To effect damage assessment, one may pair infliction and identification.

PORT LAYERS

At the highest level of abstraction, the ISO OSI RM Application/Application process layer is also extended sideways and downward to include additional **ports** which are associated with the other types of interactions for C^2, i.e., transportations, identifications and inflictions. The layers of these additional ports are analogous and isomorphic to the ISO OSI RM port for communications. Such a metastructure for all types of ports may be motivated in two ways: a) by the general metamodel for generic problem-solving as described in the Approach section, or b) by the less abstract paradigm of having user-oriented services supported by the upper three layers, network-oriented services supported by the lower three layers, and the user-network interface facilitated through a middle layer. Thus, the C^2 layers which are incorporated within the application layer of the port provide integrated command and control over all types of subordinate interactions. The four fundamental types of interactions, identifications, communications, transportations and inflictions are described in more detail in the following subparagraphs.

Corresponding to each class of interactions, i.e., identification, infliction, transportation, or communication, there exists a set of ports which are contained as part of an asset, i.e., sensor, weapon, vehicle, or transceiver. Each asset is responsible for using its ports to send or receive a set of packages, i.e., image, ordnance, message, or cargo. Analogous to the ISO OSI RM where each communication layer is responsible for providing certain data handling services in support of the application, each port layer is responsible for providing certain services corresponding to the type of package handled by the port. The key functions which are generic to any port at the appropriate layer are listed below.

Port	Functionality
Layer 7	Batch transfer access, virtual asset, package repository and distribution
Layer 6	Package-transaction, -packing, -security, -encryption, -labeling
Layer 5	Port connection (logical) for orderly package exchange including initialization (Log on / Sign on) and termination (Log off / Sign off)
Layer 4	Transparent transfer of packages between end-to-end port entities ensuring that packages are exchanged with reliability (error-free) and integrity (intact, temper-proof)
Layer 3	Establish, monitor, maintain, terminate network connections for routing of heterogeneous packages across multiple heterogeneous networks
Layer 2	Access and sharing of the physical environment (media) to minimize collisions
Layer 1	Physical connection to the environment through electrical, mechanical, nuclear, biological, chemical and other environmental characteristics of the network of assets

IDENTIFICATION LAYERS

Identification is an interaction which directly results in the recognition of objects in the environment. Signals retrieved from the environment result in the recognition of objects in the environment. The identification port is responsible for correlation, fusion, aggregation, and exploitation of signal parameters processed by its own sensor assets. The identification port interactions are typified in Figure 28. A Package of identification is called an Image. Identification port interactions range from signal acquisition from objects in the environment through the exploitation of associated data, information, knowledge, and individual experience of target identification. The identification port interactions are subject to further study. The identification port interactions will be described in a common framework developed and coordinated herein or by reference. Identification ports generally fall into one of the categories listed in Table 7.

Table 7. Classes of identification ports

	Port Class	Concrete Port Object
a)	Vehicle-Sensor	MTI Radar, Motion Detector, Tracer, Illumination Flares
b)	Sensor-Sensor	Radar Detector, Laser Detector
c)	Weapon-Sensor	Missile Detector, Incoming Round Detector
d)	Transceiver-Sensor	Radio Interceptor

Examples of identification ports include any one of the following sensor ports: radar, sonar, electro-optical, laser rangefinder, celestial navigation, and compass navigation.

Examples of Images are recordings or telemetry from photographic, audio, video, sonar, radar, or seismic sensors.

Port	Functionality of Identification
Layer 7	Batch image transfer access, virtual sensor, image repository and distribution
Layer 6	Image-transaction, -packing, -security, -encryption, -labeling, -source encoding
Layer 5	Port connection (logical) for orderly image exchange including initialization (Logon) and termination (Logoff)
Layer 4	Transparent transfer of images among multiple sensors and target entities ensuring that images are exchanged with reliability (error-free) and integrity (intact, temper-proof)
Layer 3	Establish, monitor, maintain, terminate sensor network connections for routing of heterogeneous images across multiple heterogeneous sensor networks
Layer 2	Access and sharing of the physical environment (media) to minimize collisions or loss of images
Layer 1	Physical connection to the environment through electrical, mechanical, nuclear, biological, chemical and other environmental characteristics of the network of sensors and targets

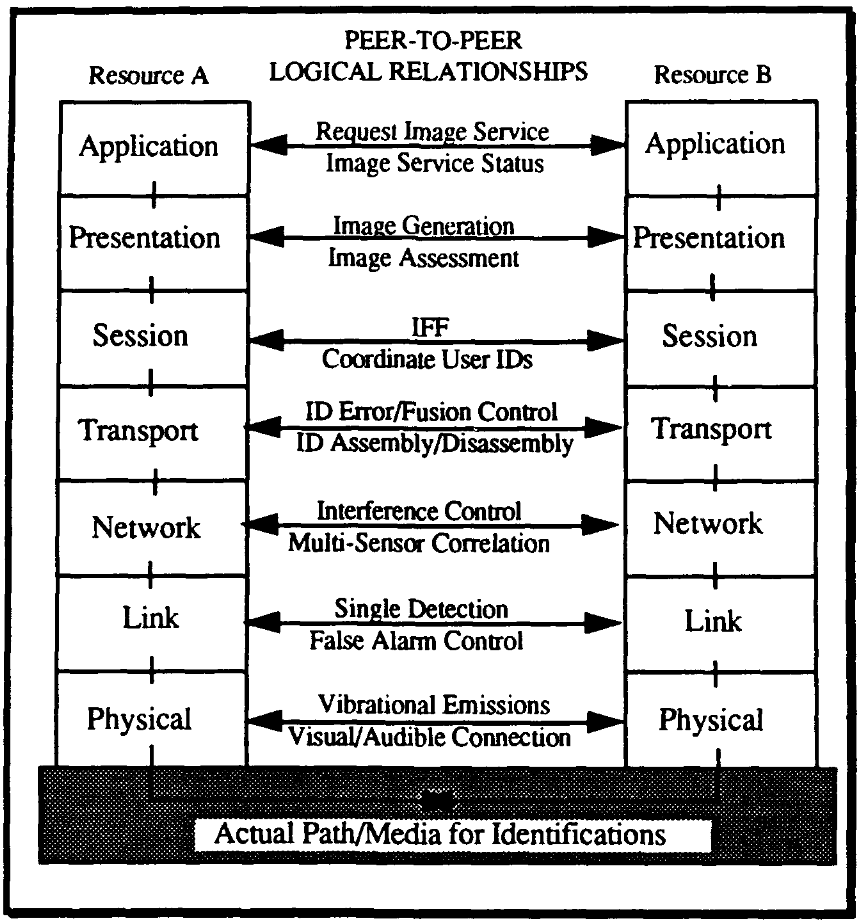

Figure 28. Representative peer layer interactions for identifications

INFLICTION LAYERS

Infliction is an interaction which directly results in the destruction, damage, degradation or disruption of unit objects or environment objects. Infliction is used to reduce the capabilities of adversarial C^2 systems through a variety of means called armaments which may cause infliction upon target resources involved in the conflict. Armaments may cause infliction upon physical or logical objects through direct or indirect impact on target resources. The infliction port interactions are typified in Figure 29. A Package of infliction is called an Ordnance. Infliction port interactions range from the use of armaments in single shots at isolated targets to the general use of armaments in coordinated engagements. The infliction port interactions are subject to further study. The infliction port interactions will be described in a common framework developed and coordinated herein or by reference. Infliction ports generally fall into one of the following categories:

Table 8. Generic classes of infliction ports

	Port Class	Concrete Port Object
a)	Vehicle-Weapon	Launcher of ATM, Torpedo, SAM, Emplacers of Mine, Barbed Wire, Abatis, Ditch digger, Berm tractor
b)	Weapon-Weapon	Launchers of Anti-Missile Missile
c)	Sensor-Weapon	Chaff Dispenser, Laser Gun, Smoke Gun
d)	Transceiver-Weapon	Jammer

Examples of infliction ports include any one of the following weapon ports: cannon, launcher, gunner, emplacer, high-energy laser, high-energy electromagnetic pulser and jammer. Operationally, jammers are weapons that pollute the electromagnetic environment. From a modeling point of view, however, jammers are equivalent to transmitters of noise or interference. Note that from similar considerations an emplacer or dispenser of obstacles is also a weapon port.

Examples of Ordnance include projectiles (e.g., bullets, grenades, bombs, shells), torpedoes, rockets, missiles, mines, high-energy electromagnetic radiation, and scatterable or dispensable obstacles.

Port	Functionality of Infliction
Layer 7	Batch ordnance transfer access, virtual weapon, ordnance repository and distribution
Layer 6	Ordnance-transaction, -packing, -security, -encryption, -labeling
Layer 5	Port connection (logical) for orderly ordnance exchange including initialization (Log on / fuse setting) and termination (Log off / trigger off)
Layer 4	Transparent transfer of ordnances between multiple weapons and target entities ensuring that ordnances are exchanged with reliability (error-free) and integrity (intact, temper-proof)
Layer 3	Establish, monitor, maintain, terminate weapon network connections for routing of heterogeneous ordnances across multiple heterogeneous weapon networks
Layer 2	Access and sharing of the physical environment (media) to minimize collisions or loss of ordnance
Layer 1	Physical connection to the environment through electrical, mechanical, nuclear, biological, chemical and other environmental characteristics of the network of weapons and targets

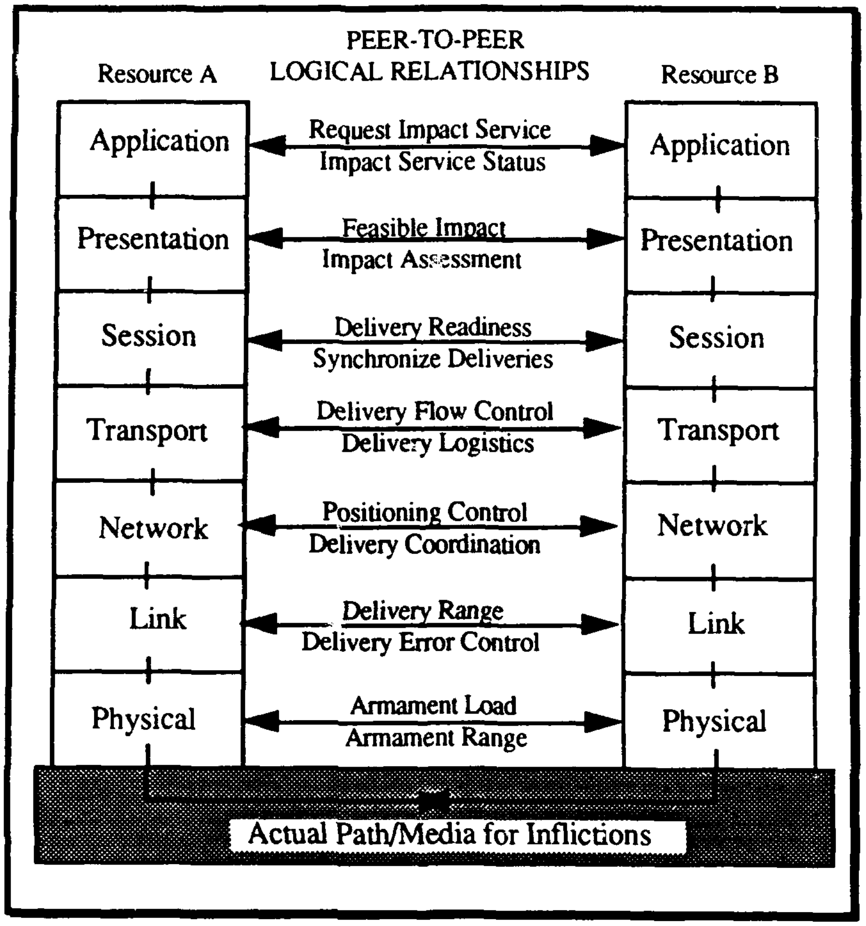

Figure 29. Representative peer layer interactions for inflictions

TRANSPORTATION LAYERS

Transportation is an interaction which directly results in the motion of objects. Transportation is used to propel, carry, supply, strengthen, equip and load resources with the necessary physical assets. Vehicles, manual or motorized, provide for the translation and rotation of physical objects as well as the movement of resources through the environment. The transportation port interactions are typified in Figure 30. A Package of transportation is called a Cargo. The transportation port interactions range from packaging of physical objects to the unpacking for expenditure, consumption or coordinated maneuver. The transportation port interactions are subject to further study. The transportation port interactions will be described in a common framework developed and coordinated herein or by reference. Transportation ports generally fall into one of the following categories:

Table 9. Generic classes of transportation ports

	Port Class	Concrete Port Object
a)	Vehicle-Vehicle	Aircraft Carrier, Transport Plane
b)	Weapon-Vehicle	Self-Propelled Gun, Gun Ship, Battle Ship
c)	Sensor-Vehicle	UAV, Scout Helicopter
d)	Transceiver-Vehicle	Mobile Radio

Examples of transportation ports include any one of the following vehicle ports: truck, craft, train, vessel, satellite. Vehicles are typically classified as ground, maritime, amphibious, air, or space. Given an initial reference point, a transportation port based navigation system such as gyro or autopilot is provided as a transportation link service.

Examples of Cargo are assets, food, fuel, ammunition, equipments and other items or supplies of various categories.

Port	Functionality of Transportation
Layer 7	Batch cargo transfer access, virtual vehicle, cargo repository and distribution
Layer 6	Cargo-transaction, -packing, -security, -encryption, -labeling
Layer 5	Port connection (logical) for orderly cargo exchange including initialization (Logon) and termination (Logoff)
Layer 4	Transparent transfer of cargo among multiple vehicle entities ensuring that cargo is exchanged with reliability (error-free) and integrity (intact, temper-proof)
Layer 3	Establish, monitor, maintain, terminate vehicle network connections for routing of heterogeneous cargo across multiple heterogeneous vehicular networks and stations
Layer 2	Access and sharing of the physical environment (media) to minimize collisions or loss of cargo
Layer 1	Physical connection to the environment through electrical, mechanical, nuclear, biological, chemical and other environmental characteristics of the network of vehicles and stations

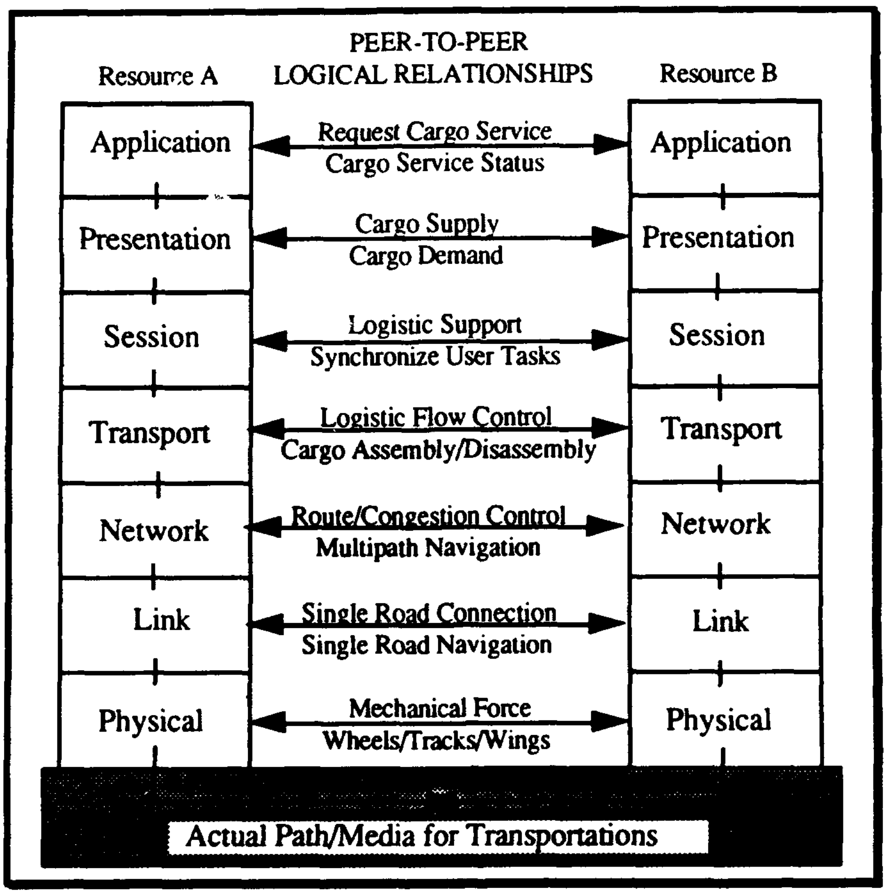

Figure 30. Representative peer layer interactions for transportations

COMMUNICATION LAYERS

Communication is an interaction which directly results in an exchange of data, information, knowledge and experience. Communication is used to command, control and coordinate among the resources. Signals transmitted and received from other resources allow for the exchange of information through the environment. The communication port interactions are typified in Figure 31. A Package of communications is called a Message. Communications port interactions range from signal generation to data updates as described in a common framework developed and coordinated in accordance with the ISO OSI RM [[1, 2]]. Communication ports generally fall into one of the categories listed in Table 10.

Table 10. Generic classes of communication ports

	Port Class	Concrete Port Object
a)	Vehicle-Transceiver	Crew Intercom, GPS Navigation, Beacon
b)	Weapon-Transceiver	Fire Control System
c)	Sensor-Transceiver	Telemetry System
d)	Transceiver-Transceiver	Modem, Radio, Relay, Router, Bridge

Examples of communication ports include any one of the following transceiver ports: radio, wire, fiber optics, audio, infrared, or optical.

Examples of Messages are commands, requests, reports, status or facts thereof.

Port	Functionality of Communication
Layer 7	Batch file transfer access, virtual terminal, data repository and distribution
Layer 6	Data-transaction, -packing, -security, -encryption, -labeling
Layer 5	Port connection (logical) for orderly data exchange including initialization (Logon) and termination (Logoff)
Layer 4	Transparent transfer of messages between multiple transceiver terminal entities ensuring that messages are exchanged with reliability (error-free) and integrity (intact, temper-proof)
Layer 3	Establish, monitor, maintain, terminate transceiver network connections for routing of heterogeneous messages across multiple heterogeneous transceiver networks
Layer 2	Access and sharing of the physical environment (media) to minimize collisions or loss of data/messages
Layer 1	Physical connection to the environment through electrical, mechanical, nuclear, biological, chemical and other environmental characteristics of the network of transceivers

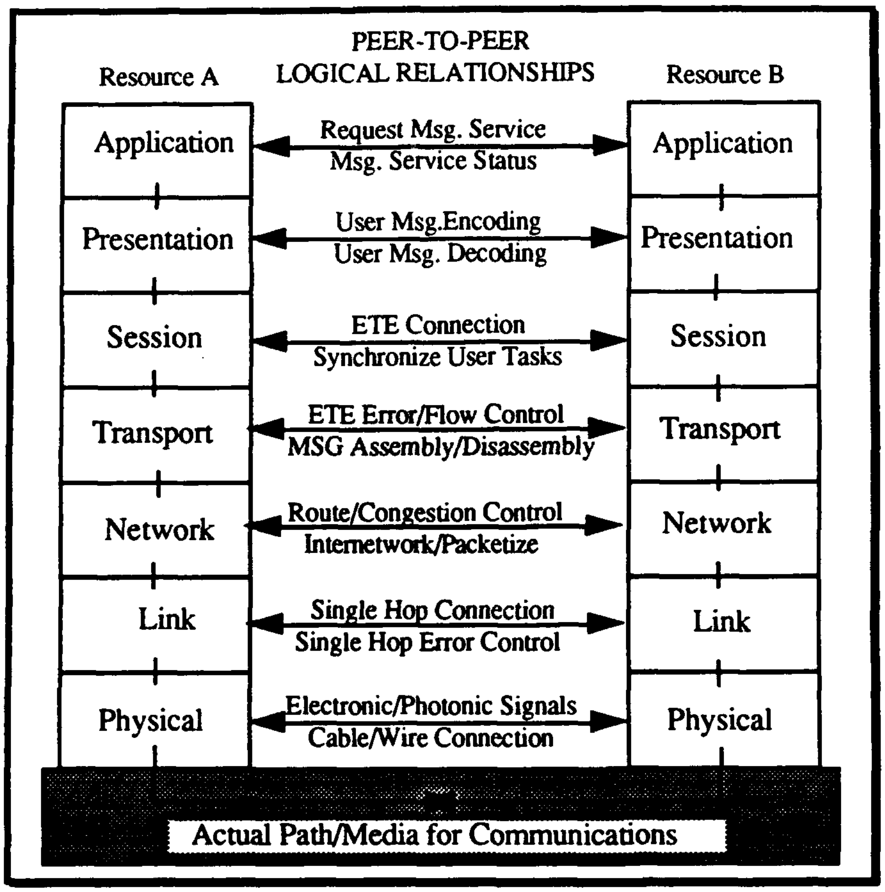

Figure 31. Representative peer layer interactions for communications

BASE LAYERS

The entities which implement C^2 applications and associated interaction ports are the physical implementations which characterize the performance of a resource. These entities constitute a set of base layers which provide a resource with the capabilities to utilize its supplies, equipments, and tools to build and use its own base of objects, information, knowledge, and experience. Whereas a problem/solution domain (psd) provides the semantic shell for C^2 applications and associated interaction ports, the implementation/technology domain (itd) provides the syntactic shell as a base for C^2. Thus, base layers are associated with or are made available to each of the C^2 layers and associated port layers as shown in Figure 32. Note that since products may be factorized (decomposed) into atomic products of services to define facts, itd products may be factorable into atomic products of itd services to define itd facts. For each psd fact there may be more than one itd fact. It is also possible, however, for one class of itd facts to represent a multiple set of psd facts. Messages or images are examples of psd facts. Message formats or image formats are examples of itd facts. Interoperability and open architectures are critically dependent upon commonality of both itd facts and corresponding psd facts.

Each of the itd services may encapsulate its own reasoning utilities or facilities capable of multistage decision support. When input events are characterized by limited accuracy, and questionable relevance, a confidence factor may be attributed to every output action. Multiple classes of events and associated uncertainties may feed a single itd entity. Multistage decision services may employ automated reasoning utilities for deriving information conclusions, knowledge recommendations, and experience judgements in support of object interaction of all classes (e.g., detection, reception, motion, elimination).

Examples of well known classes of reasoning utilities are: a) probabilistic (e.g., Bayesian), b) possibilistic (e.g., Dempster-Shafer), and c) Empiric (Fuzzy logic). Probabilistic methods utilize Bayes theorem to update the probability of a particular hypothesis given the probability of a newly acquired event and the *a priori* probability of the hypothesis before the arrival of the newly acquired event. The Confidence Factor is a probability. Possibilistic utilities such as Dempster-Shafer may avoid Bayes theorem by treating hypotheses independently. The Confidence Factor is the strength of belief in the validity of the hypothesis. Empirical utilities such as Fuzzy Logic ascribe discrete membership, association, and composition values for use in approximate inferencing. The Confidence Factor is a fuzzy measure.

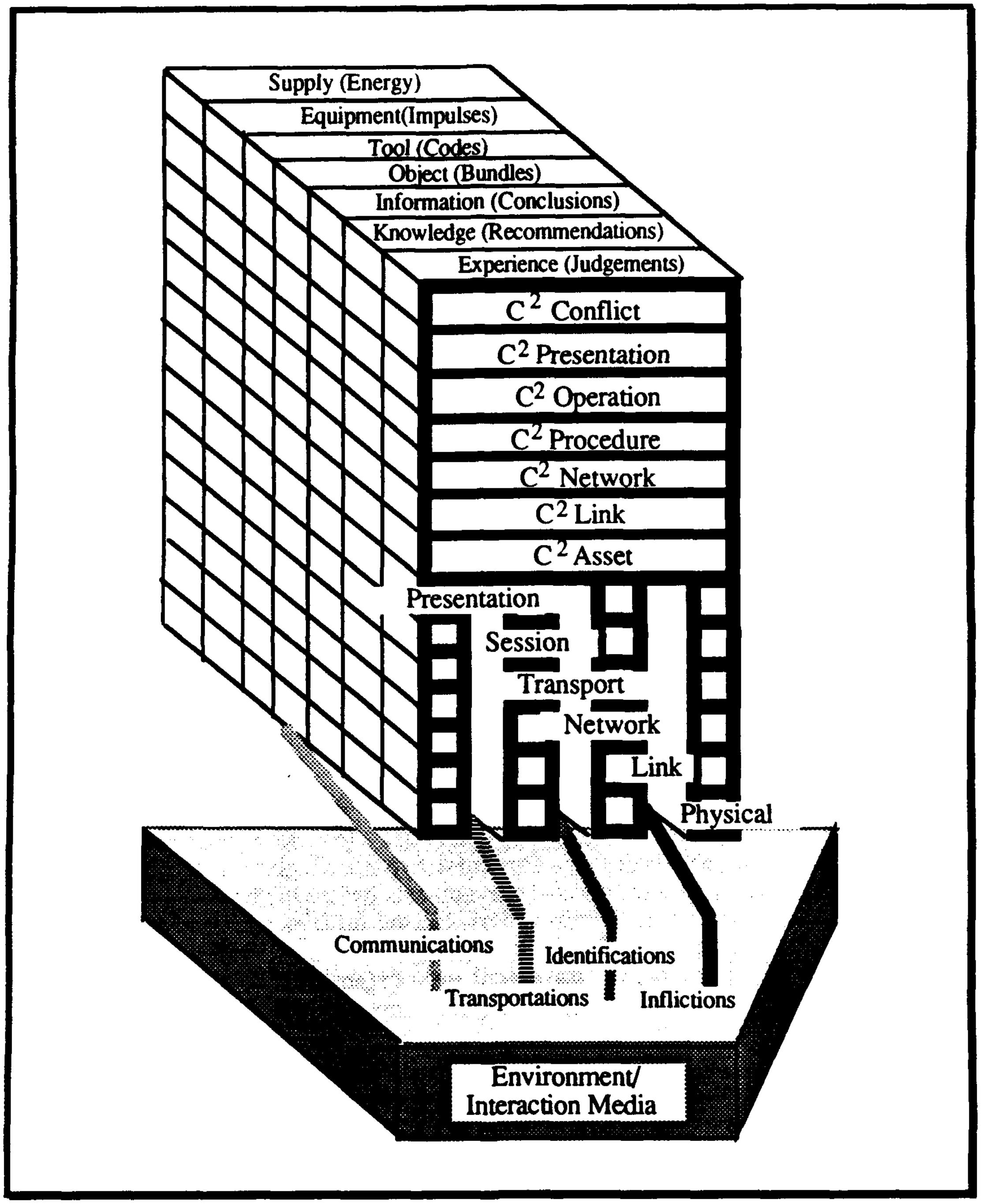

Figure 32. Projecting C^2 layers onto base layers

From an implementation/technology domain perspective, base layers are also involved in peer level interactions as shown in Figure 33. Resources are capable of consuming or producing a wide variety of energy forms through a diversified set of supply services. Supportability of any capability is dependent upon equipment services which generate carriers to facilitate the transition of energy involved in various types of interactions. Flexibility and portability are inherent through onboard tool services. A variety of modes or representations may be admissible. An impact in the environment, for example, may propagate and enter through the Object layer of any one of the C^2 Ports. The associated carrier is processed by the tool services and converted into an input mode for more efficient processing and storage. Tool bases require a high degree of portability to facilitate the leveraging of implementation and technology across resources. Admissibility into the object-base and the accuracy of the representation are taken into account by the object services. The object services manage arrays of similar objects. A new array is analyzed by information services in conjunction with other information to provide new information in the form of a candidate conclusion about the relevancy, certainty and significance of the observed impact. The new conclusion is consolidated by the knowledge services with previously derived conclusions to generate new knowledge in the form of a candidate recommendation. Recommendations should be practical and predict potential causal consequences. The new recommendation is subsequently reviewed by experience services in conjunction with previously accumulated experience, motivation and known intentions to determine whether or not to accept the recommendation and prioritize it with respect to previously accepted recommendations.

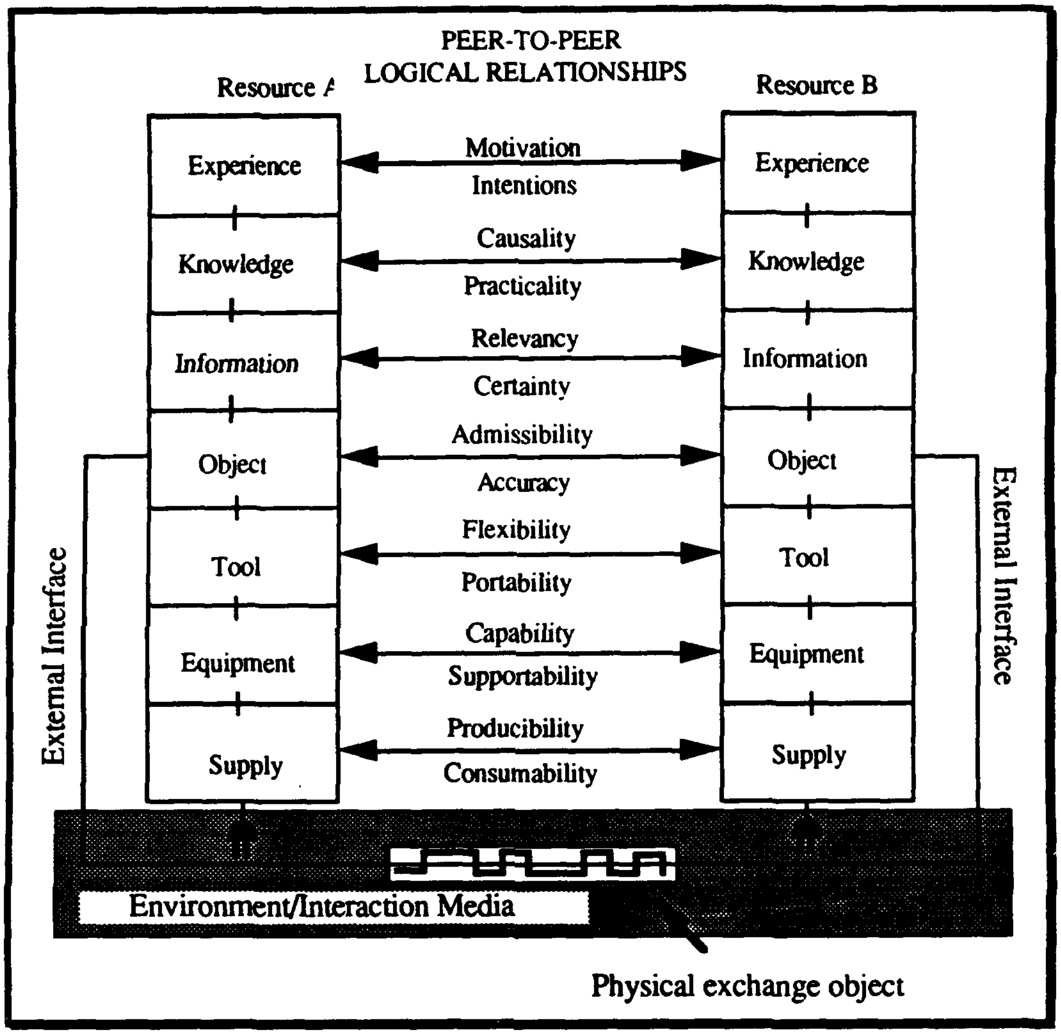

Figure 33. Representative peer level interactions for base layers

Base Services

A base layer typically involves five classes of services as depicted in Figure 34. These services provide utilities and facilities for accessing, displaying, editing, processing and storing (N+1), N, or (N-1) product objects. Each Service is typically supported by utilities and facilities which include distributed processing, multimedia visualization, multidevice editing, multilogic processing and multistorage persistence. m_Flow services provide for sharing and querying utilities for C²ed objects within a layer or across layers. m_Storage services employ random or sequential access utilities to primary (main-memory) or secondary storage devices (disc). m_Display services provide window viewing utilities with raster or vector drawing facilities. m_Process services provide computational or inferential utilities. m_Entry services respond to manual commands associated with keying, touching, clicking, or speaking devices.

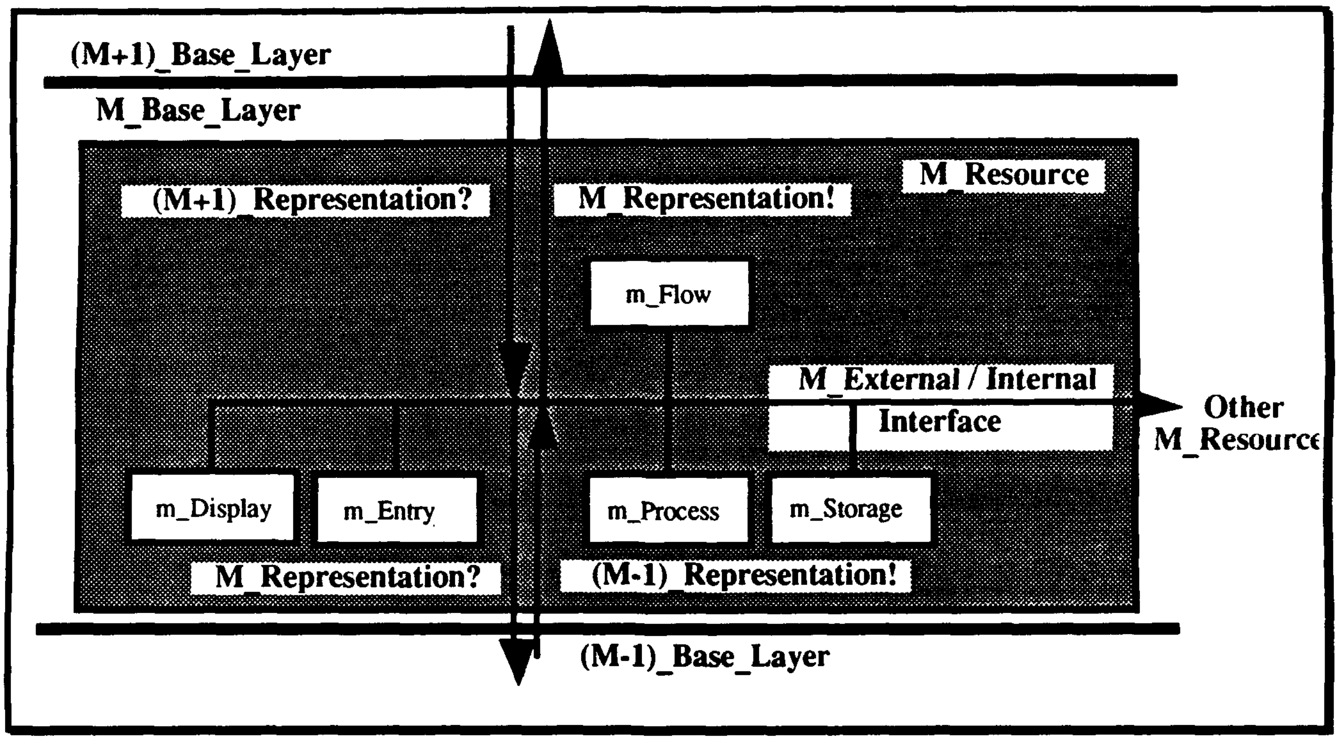

Figure 34. Generic services of base layers

EXPERIENCE LAYER

The Experience layer provides judgements from available recommendations. Experience services evolve from exercising artistic talent capable of one of a kind nonrepetitive, nonroutine exceptions and surprise handling. It may be supported by inductive reasoning and cognitive pattern recognition techniques such as neural networks and genetic algorithms. Descriptions of various cases are obtained from practical experience which evolves from unique scenarios. It employs evidential reasoning to handle scenario-dependent situations which cannot be anticipated ahead of time by knowledge services. Experience services are responsible for learning. As new cases are experienced, they are recorded in a descriptive language which can capture the context of the facts of the situation surrounding noteworthy events. As more facts are processed, they may not fit the assumptions or criteria of built-in hypotheses tests, as conducted by knowledge services. Experience services, therefore, will determine the degree to which history may have been repeated and the degree to which previously known cases are relevant to provide insight on which to base a judgement of the present situation. A human official may be able to motivate and justify his intentions with respect to a CoA correction on the basis of his intellectual capacity as described by the Experience layer. The Experience layer provides the highest level of rationalized decisions which require understanding and cognizance. In human officials, the experience services may be influenced, however, by emotions and instincts. The Experience layer provides for the ability to sort through the Knowledge layer, chain selected inferences, and apply them in finalizing a judgement solution from a given problem-generic layer.

The Experience layer services are responsible for validating choices of actions made available through the Knowledge layer. Most generally, Experience layer services provide inferencing capability upon available knowledge. Knowledge may be aggregated or deaggregated, correlated or decorrelated, associated or deassociated by Experience layer services. Having a comprehensive knowledge model and knowledge access is key to the success of Experience layer services.

KNOWLEDGE LAYER

The Knowledge layer provides recommendations from available conclusions. The Knowledge layer provides well-established rules of behavior which may be obtained from experts with extensive experience supported by scientific experiments. This layer evolves from exercising scientific talent capable of credible, accurate, reliable, routine, and expected event handling. It allows for handling generic, scenario-independent situations which may be anticipated ahead of time with some probability. The Knowledge layer is responsible for time-proven training and memory refreshing with school solutions to previously known and satisfactorily solved problems. The Knowledge layer may be supported by *a priori* defined pattern matching techniques aided by prediction, estimation and optimization techniques. It is primarily based upon deductive and abductive heuristics as employed by conventional artificial intelligence (AI). As relevant information is extracted and reviewed concerning new events, it is cast into premise statements of conclusions from which recommendation statements can be drawn. The process of drawing recommendations from conclusions may use any one of several calculi of logic which may be appropriate. Examples of such tools include production rules, goal-directed backward chaining, demon-driven forward chaining, Bayesian inferencing, and hypothesis testing. Production rules use deductive reasoning which allows recommendations to be drawn as necessary consequences of the truth of conclusions. Bayesian inferencing allows recommendations to be drawn from the statistical nature of the conclusions, combining premises with previously derived prior and conditional probabilities.

The Knowledge layer services are responsible for rank ordering choices of actions as supported by information available through the Information layer. Stated most generally, Knowledge layer services provide an inferencing capability upon available information. Information may be aggregated or deaggregated, correlated or decorrelated, associated or deassociated by Knowledge layer services. Having a comprehensive information model and information access is key to the success of Knowledge layer services.

INFORMATION LAYER

The Information layer provides conclusions from available data. As records of data are extracted and reviewed concerning new events, they are cast into premises statements of input data from which conclusion statements can be drawn. The process of drawing conclusions from input bundles, files or records of data may use similar tools made available to the Knowledge layer. For example, production rules may use deductive reasoning to allow conclusions to be drawn as necessary consequences of the truth of the premises included in the input data records. Bayesian inferencing allows conclusions to be drawn from the statistical nature of the input data records, combining premises with previously derived prior and conditional probabilities. In addition, The Information Layer includes traditional procedural programming techniques, mathematical analyses and simulations which extract information from available data about key variables, parameters, measures-of-performance, measures-of-effectiveness and premises which may be computed and evaluated upon request by higher services.

Most generally, Information layer services provide inferencing capability upon available data. Data may be aggregated or deaggregated, correlated or decorrelated, associated or deassociated by Information layer services. Having a comprehensive data model and data access is key to the success of Information layer services.

OBJECT LAYER

The Object layer provides the capability to handle packages of various classes (e.g., messages, cargo, ordnance, or images) as input or as output from a resource. The Object layer provides for the extraction and management of data from input objects. At the Unilateral (Asset/Physical) layer, objects may be collected from other resources or from the environment. At layers above the Unilateral layer, objects are shared with other resources. Note that what is information, knowledge or experience to one resource may be an input object for another. The converse may be true also depending upon the application and perspective. The relative nature of interoperability services between layers should be evaluated from resource to resource.

The services of the Object layer are particularly important for automated or semiautomated resources where computer data may support a wide variety of users and applications. A resource may be inundated with data and algorithms. Data and corresponding algorithms must be properly formatted, encapsulated, compressed, uncompressed, filtered, sorted, merged, and edited before it may be useful. Object layer services may also provide the capability to convert Data and algorithms from one structure to another, e.g., text to graphics, or audio to text and vice versa. Object layer services should also support multimedia association between data objects and various data object queries, e.g. Structured Query Language (SQL)

TOOL LAYER

The Tool layer provides the processing environment necessary to support the higher layers as well as to monitor the status of the lower layers. It is highly software-, firmware- and operator skills-oriented. It includes application-independent programs and libraries supported by operating systems, languages, utilities, storage management, man-machine entry and display packages as well as manual Standard Operating Procedures (SOPs).

The services of the Tool layer are particularly important for automated or semiautomated resources where computer software may support a wide variety of users and applications. The operating system provides the bulk of the tool layer services. It includes the system clock, a software drivers for each hardware drive, partitioning and alocation of RAM and disc space, and providing access to system resources through system administration support. The Tool layer is responsible for installation of the tools themselves as well as the applications and their databases. The Tool layer should provide for file management, directory assistance, persistence of software system configuration and backup provisions. The Tool layer is responsible for orderly powerdown under normal conditions as well as in emergency situations such as problems with power or equipment failure. The Tool layer is should also provide for reconstituting the latest viable configuration salvageable from emergency interruptions.

EQUIPMENT LAYER

The Equipment layer allocates and provides the physical space and media necessary for hosting the higher technology layers. For software and firmware tools it consists of electronic, electromagnetic, magnetic and optical media and hardware. For interactive tools it consists of personnel augmented by display-oriented hardware. Equipment services may include built-in test equipment (BITE). For machines, the equipment layer embodies the body of the machine, i.e., machine hardware. For humans, the equipment layer embodies the body of the human, i.e., human 'fleshware.'

The services of the Equipment layer are particularly important for automated or semiautomated resources where computer hardware may include a wide variety of boards or cards with large scale integrated circuits, e.g., CPU, RAM, ROM, arithmetic logic, symbolic logic, and peripheral interfaces and devices such as keyboards, pointing devices, monitors,disc drives, tape drives, scanners, printers, and plotters. Equipment layer services support putting the devices on line through appropriate formatting and mounting procedures. Equipment layer services include hardware diagnostics, and notifications of availability or utilization of any board or device. Equipment layer services should also provide warnings to insure proper equipment maintenance,and to alert of dangerous equipment status such as overheating, excessive vibrations, or loose connections,

SUPPLY LAYER

The Supply layer allocates and provides energy and power for equipment services. The Supply layer becomes important in equipment overloaded situations. For example, a single source of power may not be capable of turning on all equipments at once or at full operational capability. The Supply layer may have to budget available supplies. In most operational environments this is a real concern. If however such a problem does not exist for a given official, the Supply layer services may be assumed to be provided automatically, instantaneously, or as required. Examples of supply include food rations for personnel, electrical power for electronic gear, fuel for vehicles, electromagnetic energy for sensors, and propellent for armaments.

The services of the Supply layer are particularly important for automated or semiautomated resources, i.e., resources with computers. A computer may be supplied with power from a variety of sources, e.g., 110/220 volt, 50/60 Hz, AC power distribution line , a gas powered field generator, or a battery power pack. Supply sources may not always match or meet the specified supply requirements such as power consumption rates, e.g., wattage, or input supply levels, e.g., voltage thresholds. Furthermore, a power distribution line may be noisy, and vulnerable to surges induced by other non organic devices on the line or near the line, or by the weather, e.g., lightening. A battery power pack may need to be recharged every few hours. Supply layer services therefore must provide protection against a variety of dangerous or spurious supply types and levels and warnings and alerts of various types to prompt corrective actions. Typical portable or laptop computers include a variety of power monitoring and power conservation utilities. It is left to the Tool layer with the support of the Equipment layer to properly utilize the services of the Supply layer. The importance of the Supply layer can not be underscored any more than by considering the consequences of the disruption of a mission due to unexpected interruption of operations for lack of supply. Improperly designed Supply layer services can lead not only to an asset downtime but to asset equipment failure or malfunction. When a power outage occurs, a time critical supply service may be capable of sustaining operations without or with minimum interruption by shifting to alternate standby power sources.

IMPLEMENTING APPLICATIONS

The two-layered architectures for applications and bases form a two-dimensional matrix of resource services as shown in Figure 35. This matrix of layers allows resources to have distributed applications across distributed bases for any given layer. Thus, for example, when an official must provide a service to decide upon its next product correction or adjustment, efficient access mechanisms may be provided to all seven base layers for support. Note that this structure is highly flexible since it permits access to base services of adjacent application layers. The two-dimensional (nxm) service access is needed to exploit the status maintained by a common set of base layers for multiple application layers. In this respect the C^2RM requires a structured interface for both vertical (application layers) and horizontal (base layers) boundaries to facilitate open system interconnection within a resource.

Base services may be activated within each layer of C^2, with a request which propagates in either real or virtual time through the seven base layers. Requests typically start top-down with the Experience layer and culminate in the Supply layer. Responses to requests proceed also in real or virtual time utilizing supply services, and ultimately reach the services of experience. Responses may take on a variety of forms. The request for a service may be conditionally or unconditionally: a) subject to acknowledgement, b) granted, c) denied or d), delayed.

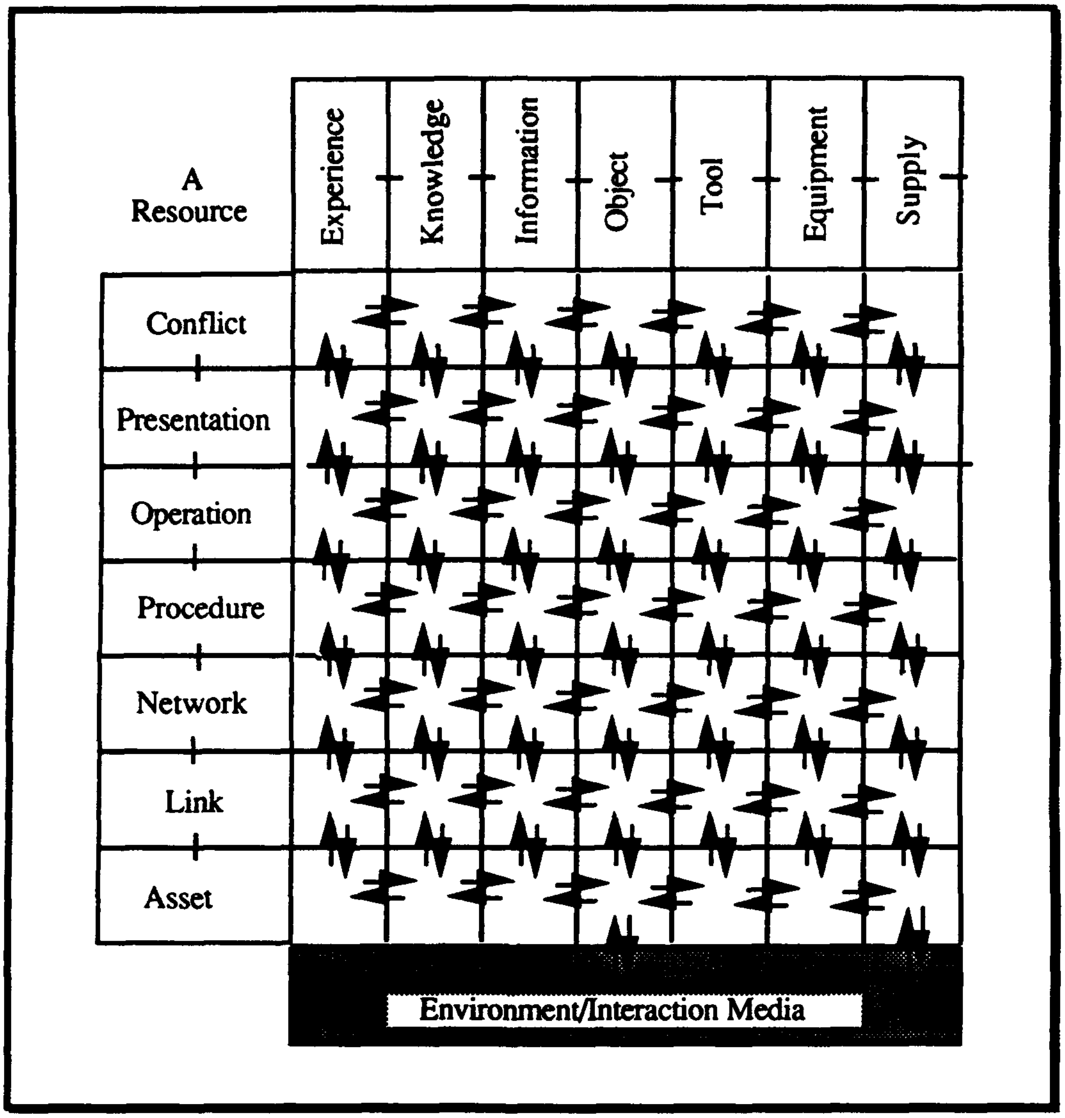

Figure 35. Coupling (typing/binding) layered bases with layered applications

ANNEXES

The following Annexes form a part of of the C²RM.

ANNEX A: APPLICABLE DOCUMENTS

1. ISO 7498(Draft) Information Processing Systems - OSI Basic Reference Model, 1984.

2. ISO 7498(Draft) Information Processing Systems - OSI Proposed Draft, Addendum 2 to ISO 7498 on Security Architecture, 30 January 1986.

3. ISO ASN.1

4. *Technical Reference Model for Corporate Information Management*, Version 1.0, Center for Information Management, U.S. Defense Information Systems Agency, 4 November 1991.

5. ISO/IEC JTC1/SC21/WG7, Basic Reference Model of Open Distributed Processing (ODP-RM), Contact Address: Secretariat: Netherlands Normalisatie-Institute (NNI) or Accredited Standards Committee, X3, Information Processing Systems, X3T3.Operating under ANSI.

ANNEX B: GLOSSARY

1. DEFINITION SETS

ARCHITECTURAL TUPLETS

A.	Port	the layers associated with a particular class of interaction
B.	Interaction	a set of actions, events and reactions which involves an exchange of packages
C.	Official	an asset responsible for a given application layer
D.	Method	a set of rules for generating and evaluating object states and their transitions
E.	Leader	an asset providing the initiative, motivation and will to solve a conflict
F.	Product	an output of a layer which may be a representation of requirements or status
G.	Conflict	a set of adversarial relationships which threaten or actually disrupt security
H.	Representation	a form or an instance of an implemented product
I.	Base	a set of implementation / technology services necessary to realize an application
J.	C^2 Application	a set of problem/solution services necessary to support a C^2 system process
K.	Organization	a set of resources and their structural interrelationships
L.	C^2 Service	a set of interrelated utilities
M.	C^2 Mode	a phase of interrelated services
N.	Package	a set of interrelated utilities
O.	PSD	Problem/Solution Domain generic organization and capabilities
P.	ITD	Implementation /Technology Domain generic organization capabilities
Q.	Scenario	a depiction of the evolution of a conflict in a given conflict region

A. PORT SEPTUPLETS

A.1. Physical	pertaining to personnel or materiel objects
A.2. Link	a direct coupling relation between two objects
A.3. Network	an association of physical links for the purpose of realizing logical links
A.4. Transport	a coupling relation between two objects mediated by other objects in a network
A.5. Session	a set of interactions essential to initiate and complete the execution of a product
A.6. Presentation	a set of representations associated with form, fit and functional relevance of products
A.7. Application	the use of port to support asset

B. INTERACTION QUADRUPLETS

B.1. Communications	interactions resulting in the exchange of packages called messages
B.2. Transportations	interactions resulting in the displacement of packages called cargo
B.3. Identifications	interactions resulting in the recognition of images derived from units and the environment
B.4. Inflictions	interactions resulting in impact to target objects. An impact is an amount of destruction, damage, degradation or disruption caused by ordnance.

C. OFFICIAL SEPTUPLETS

C.1. Operator	an official for asset instructions and associated packages
C.2. Coordinator	an official for linking techniques and associated transactions
C.3. Administrator	an official for network disciplines and associated assignments
C.4. Agent	an official for procedural schemata and associated jobs
C.5. Controller	an official for operational tactics and associated tasks
C.6. Planner	an official for presentation strategies and associated plans
C.7. Commander	an official leader for conflict resolution, policies, and associated missions

D. METHOD SEPTUPLETS

D.1. Instruction	a method for packaging assets. Instructions are ways of making a resource operational.
D.2. Technique	a method for transacting across links. Techniques are specialized bilateral protocols for coupling resources.
D.3. Discipline	a method for assigning responsibility and priority. Disciplines are well-defined protocols for using resources.
D.4. Schema	a method for working out jobs. Schemata are ways of using basic skills.
D.5. Tactic	a method for tasking. Tactics are ways of optimizing short-term risks.
D.6. Strategy	a method for planning. Strategies are ways of optimizing long-term risks.
D.7. Policy	a method for setting missions. Policies are ways of imposing constraints

E. LEADER SEPTUPLETS

E.1.	Expert	a commander for armament services
E.2.	Partner	a commander for engagement services
E.3.	Captain	a commander for combat services
E.4.	Manager	a commander for battle services
E.5.	Director	a commander for campaign services
E.6.	General	a commander for war services
E.7.	President	a commander for peace services

F. PRODUCT SEPTUPLETS

F.1.	Package	a product of the Asset layer identifying a specific asset port necessary to execute a given transaction for interaction
F.2.	Transaction	a product of the Link layer identifying specific resources selected to carry out a given assignment
F.3.	Assignment	a product of the Network/Internetwork layer associating specific resources with given jobs
F.4.	Job	a product of the Procedure layer identifying specific work units necessary to execute a given task
F.5.	Task	a product of the Operations layer prescribing a near-term course-of-action
F.6.	Plan	a product of the Presentation layer prescribing a long-term course-of-action
F.7.	Mission	a product of the Conflict layer identifying and motivating the goals and objectives to be met by the C^2 system

G. CONFLICT SEPTUPLETS

G.1.	Armament	a conflict of interacting assets
G.2.	Engagement	a conflict of interacting links
G.3.	Battle	a conflict of supporting combat
G.3.	Combat	a conflict of interacting networks
G.5.	Campaign	a conflict of sustaining battles
G.6.	War	a conflict of concerns for peace
G.7.	Peace	the state of a space-time region void of militant conflicts

H. REPRESENTATION SEPTUPLETS

H.1. Energy	a product of supply services which process fuel essential to generating impulse objects
H.2. Impulse	a product of equipment services consisting of signals essential to generating codes
H.3. Code	a product of tool services consisting of packed impulses arranged into efficient units of capability
H.4. Bundle	a product of object services including well-structured parcels relevant to information objects and suitable for interaction
H.5. Conclusion	a product of information services including evaluated assumptions, facts and propositions relevant to recommendation objects
H.6. Recommendation	a product of knowledge services including generated candidate actions
H.7. Judgement	a product of experience services including a set of key events encountered and stored in a resource for future reference

I. BASE SEPTUPLETS

I.1. Supply	a consumable and expendable source of energy essential for the proper functioning of a resource
I.2. Equipment	a physical platform or frame capable of hosting an integrated set of tools
I.3. Tool	a capability to handle and manipulate objects
I.4. Object	a capability to import or export package objects for interaction
I.5. Information	a capability to generate and evaluate objects
I.6. Knowledge	a capability to generate and evaluate information
I.7. Experience	a capability to generate and evaluate knowledge

J. C² APPLICATION SEPTUPLETS

J.1. Asset	a capability to act iaw specific transactions
J.2. Link	a capability to interact iaw specific assignments
J.3. Network	a capability to link multiple assets iaw specific jobs
J.4. Procedure	a capability to network in support of tasks
J.5. Operation	a capability to synchronize tasks
J.6. Presentation	a capability to plan operations for specific CoAs
J.7. Conflict	a capability to deal with confrontations through specific missions

K. ORGANIZATION SEPTUPLETS

K.1.	Item	any part of an object
K.2.	Component	a configuration of items capable of executing actions
K.3.	Entity	a configuration of components grouped to perform functions
K.4.	Element	a configuration of entities grouped to perform procedures
K.5.	Resource	a configuration of elements grouped to perform tasks
K.6.	Unit	a configuration of resources grouped to perform missions
K.7.	Enterprise	an organization of units

L. C^2 SERVICES OCTUPLETS

L.1.	Environment	an assessment of the qualitative capability of the environment to support given interactions
L.2.	Friend	an assessment of the qualitative capability of the own assets
L.3.	Foe	an assessment of the qualitative capability of threat assets
L.4.	Relative	an assessment of the relative strengths of linked assets
L.5.	Requirement	a restatement of requirements derived from a command for execution or a request for service from the next higher layer
L.6.	Generation	a synthesis of requirements
L.7.	Evaluation	an analysis of synthesized requirements
L.8.	Specification	a formulation of a detailed requirement suitable for command for execution or request to service at the next lower layer

M. C^2 MODE TRIPLETS

M.1.	Assess	evaluate and compare situations (capabilities and topology) of cooperating and noncooperating/adversary resources, and the environment
M.2.	Develop	require, generate, evaluate and specify products
M.3.	Monitor	compare current execution (actions, events, and outcomes) with expected execution

N. PACKAGE QUADRUPLETS

N.1.	Ordnance	an item for infliction port action
N.2.	Image	an item for identification port action
N.3.	Message	an item for communication port action
N.4.	Cargo	an item for transportation port action

O. PROBLEM/SOLUTION DOMAIN SEPTUPLETS

O.1. Command	a headquarters resource
O.2. Center	a command resource responsible for mission, plans and tasks
O.3. Staff	a center resource responsible for situation assessment, product development, and execution monitoring
O.4. Application	a decision-aid and associated conflict region objects
O.5. Service	a decision-aid module providing a capability to monitor, assess, generate, forecast, evaluate, require, specify products
O.6. Utility	a service module providing a capability to snapshot, template, overlay, allocate, schedule, associate, synchronize conflict region objects
O.7. Facility	a utility module providing a capability to directly access conflict region objects and assess or modify their capability to interact , e.g., observe, fire, cover, conceal, communicate or move

P. IMPLEMENTATION/TECHNOLOGY DOMAIN SEPTUPLETS

P.1. Setting	a capability to configure and assign personnel and tools to equipment available to a psd command tree
P.2. Session	a capability to maintain contiguity across and between shifts of psd centers, staff and application layers
P.3. Phase	a capability to switch between modes of operations through initialization, orientation, execution, and finalization of objects, information, knowledge, and experience relevant to a given session
P.4. Base	a capability to provide access to each technology and implementation of a given application
P.5. Service	a capability to display, enter, store, process and flow product representations in a given base layer
P.6. Utility	a capability to manage and manipulate product representations
P.7. Facility	a capability to directly access or create representations of any object associated with a given base layer

Q. BASE SERVICES QUINTUPLETS

Q.1. Display	an output of a product representation to a user interface
Q.2. Enter	an input to a product representation from a user interface
Q.3. Process	a set of transformation functions applied to a product representation
Q.4. Store	a set of repository persistence and access functions applied to a product representation
Q.5. Flow	a set of collaboration and exchange functions applied to a product representation

R. SCENARIO QUINTUPLETS

R.1. Scenario	a set of snapshots related by a common course-of-action, frame of reference and background which span the globality of the Conflict region
R.2. Snapshot	a set of overlays registered with respect to a common space-time region and spanning a subregion of the Conflict region
R.3. Overlay	a frame I a template of the Conflict region which is registered relative to a background and which provides a functional view of the Conflict region
R.4. Cell	a coordination object bounding a set of interacting unit objects
R.5. C^2ed Cr object	a representation of a unit, coordination, or environment object within a C^2 resource

2. OTHER DEFINITIONS

Action	an observable output of an organizational entity
Activity	an observable set of actions or functions
Aggregated resource	a set of independent resources
Algorithmic	the application of a set of rules to a domain of problems where a correct solution can always be guaranteed
Alignment	the use of background data to achieve a common frame of reference
Ambiguity	an inherent property of data which can support multiple hypotheses
Analytic	the application of a set of rules to a domain of problems where solutions exist in closed forms
Application	a problem domain decision-aid and associated Conflict region objects
Architecture	the way in which a system may be configured
Asset	an irreducible (primitive) resource
Background	a static frame of reference
Blob	a Binary Large Object
C^2 Process	a C^2 system-global process involving observations critical to a conflict, associated decisions and actions
Canonical	generally recognized and accepted
Capability	an attribute of an object
Case	a set of experiences to include lessons learned
Characteristics	a set of parameters associated with a feature
Child	an object which inherits one or more features from another object. A child may augment or override inheritance.
Class	a generic name for a logical object with inheritance properties which characterize a set of physical objects. A class object contains data elements, which may be of different types, and a set of operations to manipulate the data.
Coalition	a set of two or more C2 systems which agree to cooperate and interoperate mutually as a unified C2 system against a common foe. Cooperation and interoperation of member C2 systems, however, depend upon good will rather than legal bindings.
Collection	a group object
Command	the actions of initiating missions
Command and Control	aka C2, integrated command and control with feedback
C2 and Communications	command and control mediated by communications
Communicator	a resource with primary responsibility for communications
Composite resource	a compound resource capable of autonomous activities
Compound resource	a set of interdependent resources
Conclusion	a product of information services

Concurrency	maintaining object_integrity when multiple object_operations are encountered
Control	the actions of insuring that the operations are being carried out according to plan
Coordination	a process of establishing space-time coordinates for unit objects with admissible actions
Course-of-Action	a sequence of actions
Da_ object	hasa Da_La object, Da_Se object, Da_Ut object, Da_Fa object
Data	a coded representation pertaining to input, output or internal object components
Data_base	an implementation where data is stored
Decision	an outcome from a rule
Decision rule	a rule for generating a decision
Demonstration	a non-operational implementation of a model with limited flexibility
Dynamics	the state as a function of time
Effectiveness	the degree to which outcomes of a lower (sub)layer product contribute to the success of a higher (sub)layer product
Environment	the physical surroundings such as land, sea, air, space and associated space-time region which characterize the channel or conduit for real interaction between resources
Equipment	a physical platform capable of running a set of tools, e.g., hardware
Evaluation	a phase of a method to assess utility and select options for transitions from one state of an object to another
Event	an observable occurrence in the environment
Exchange object	an object which may be exchanged by interacting resources
Experience	a set of judgements which may include measures of intuition
Experience_base	an implementation where experience is stored
Fact	an assertion of an atomic psd or itd product based upon known observations, decisions or actions
Field	a local region of the Conflict region
Font	a scalable implementation of a typeface
Frame	an itd object which encapsulates and represents a type of a psd object
Frame-of-Reference	a range and scale of permissible values of attributes which are common to a set of C2ed objects (e.g., space/time)
Function	a class of performance features supported by an entity
Functional	pertaining to the C^2 Application or Asset Port layers or to a given perspective thereof
Generation	a phase of a method to initiate transitions from one state of an object to another
Guidance	a general policy and associated boundary conditions

Hardware	a non-reprogrammable technology
Heuristic	the application of a set of rules to a domain of problems where a correct or optimal solution cannot always be guaranteed to be found in available time
Horizontal interaction	an interaction within a given layer across resources
Impact	a fact of outcome
Independent resource	a resource capable of autonomous activities
Indicator	a notation which precedes and identifies the intended use of associated entity or item
Inference	a postulated relationship
Information	a set of conclusions which may include measures of uncertainty (e.g., declarative inferences)
Information_base	an implementation where information is represented and st
Inheritance	the features of a child object which may be attributed to a parent object
Input	a stimulus event
Intent	perceived I interpreted I described I clarified as part of the stated mission
Interdependent resource	a resource participating in coordinated activities
Interface	an entity of a composite object which is directly responsible for its observed attributes and behavior
Irreducible	aka Primitive
Item	any part of an object, a variable, also a constant, or a value of a variable of an object
Keyword	a string of ASCII characters beginning with a capital letter A-Z formally defined in this Annex
Knowledge	a set of recommendations to include measures of risk (e.g., procedural /goal inferences)
Knowledge_base	an implementation where knowledge is represented and stored
La_ object	hasa La_Se object, La_Ut object, La_Fa object
Layer	a set of services which are grouped and phased at a given level of a hierarchy and which span all resources of a system. Unless explicitly noted all sublayers are also referred to as layers.
Logic	heuristic or algorithmic mechanisms for generating solutions or parts thereof
Logical	virtual, pertaining to a generically realizable feature of a real object
Measure of Effectiveness	aka MOE, an inherent capability parameter which can only be measured in the course of accomplishing a mission
Measure of Performance	aka MOP, an inherent capability parameter which may be measured independently of the mission
Meta-class	a class whose instances are classes with the same generic inheritance properties

Meta-knowledge	a meta-class of knowledge
Meta-model	a meta-class of models
Meta-process	a meta-class of processes
Meta-structure	a meta-class of structures
Method	a set of computational heuristics and algorithms
Model	a formal description of a class of systems which provides consistent structural and behavioral detail of embedded entities and processes
Module	a generic name for a physically or functionally aggregated set of activities, processes, services, layers or resources or some combination thereof
Node	a resource or a part of a resource which may be localized
Ob_action	represents an operation upon an Ob_target which may change the state of one or more of its Ob_base-attributes
Ob_argument	represents a variable with one or more admissible Ob_values characterizing an Ob_feature
Ob_attribute	an abstraction of analogous properties within an Ob_class. Ob_attribute is aka Ob_property. Ob_attribute isa Ob_base-/Ob_derived-attribute
Ob_base attribute	an Ob_attribute which is only changeable by an Ob_action operation
Ob_base-attribute-dynamic	isa Ob_base-attribute which may be changed by an Ob_action.
Ob_base-attribute-static	isa Ob_base-attribute which cannot be changed by any Ob_action
Ob_class	isa set of objects with common features. Ob_class represents a discrete, distinguishable set of objects characterized by a single set of Ob_features. Ob_class may be an Ob_instance of its Ob_Peer. Ob_class is hierarchically related to other Ob_classes through Cl_associate arguments.
Ob_client	isa Ob_class requesting an Ob_operation from an Ob_server
Ob_derived attribute	an Ob_attribute which is only changeable by an Ob_query operation
Ob_feature	isa Ob_attribute/Ob_operation. Ob_feature may be a Cl_feature inherited through Cl_inherit
Ob_icon	isa graphic symbol which represents the Ob_class
Ob_instance	isa single object created from an Ob_ class with one or more unique features
Ob_method	isa Ob_operation by a given Ob_class which may depend upon Ob_arguments
Ob_operation	an abstraction of analogous behavior or method within an Ob_class. Ob_operation is aka Ob_behavior. Ob_operation isa Ob_action-/Ob_query-operation. Ob_operation may apply to more than one Ob_class (polymorphism). Ob_Operation may take on different forms of implementation aka Ob_methods depending upon the Ob_class.

Ob_parameter	isa Ob_argument of an Ob_operation which does not affect the choice of an Ob_method
Ob_query	represents an operation upon an Ob_target which preserves its Ob_base-attributes
Ob_server	isa object executing an Ob_operation requested by an Ob_client
Ob_target	isa Ob_class submitting to an Ob_operation by an Ob_server
Ob_value	is an admissible representation of an Ob_argument for a given Ob_instance
Object	an organized set of entities assembled as a member of a class
Observation	the process leading to the identification of a set of events
Operational	ready for action|deployed|in effect|part of current doctrine
Order	an unconditional requirement
Outcome	a consequence or result of an event
Paradigm	a conceptual model, a metaphor
Parent	an object which provides one or more features to another object. A parent may augment or modify characteristics of features for inheritance.
Pattern	a template for a set of facts pertaining to one or more C2ed objects
Perspective	a set of variables considered in the description of an Object
Phase	a time interval identified with processing a given set of activities
Physical	pertaining to inherently realizable attributes of performance
Physical Asset	personnel and/or materiel objects
Physical layer	provides services which prepare and ready the physical assets
Port	the layers through which an entity of one resource interacts with the environment.
Power	the ability to expend given assets per unit time
Predicate	an expression used to compute a Boolean value
Primitive	an attribute of an entity which indicates that it cannot be decomposed into usable subentities
Primitive	elementary, irreducible word or a symbol which is fundamental to command and control
Program	an executable body of code
Programmable	capable of being encoded to execute more than one program
Prototype	a component of a system segment with limited operational capabilities
Rasterization	representation of an image by a bitmap
Record	a unit of storage
Registration	a set of points in a coordinate system used to provide a common origin and scale for templates and their background
Regular resource	a resource regulated by a mission

Request	a conditional/optional requirement
Requirement	a product propagating from layer N to layer N-1
Representation	an arrangement of selected features of an object
Resource	a set of entities aggregated in a physical object or subject such as a person, team, crew, or any other size unit capable of partaking in a mission with all assets under its responsibility
Rule	any logic derived from laws of nature, policy, strategy, tactics or doctrine which governs the output of an application, service, utility, or facility.
Se object	hasa Se_Ut object, Se_Fa object
Segment (of a system)	a subsystem of a system made up of an integrated/interoperable collection of prototypes
Simulation	An abstract representation of a set of entities through activation of interrelated models of constituent entities.
Slot	a representation of a feature in a frame
Software	a body of code intended for programmable hardware
Software package	a body of code delineated to provide a given functionality through an external interface
State	a set of variables which characterize object entities and span all possible Perspectives
Status	a product propagating from layer N-1 to layer N
Structure	the static interrelationship among objects (e.g., composition and containment)
Subsystem	a set of resources or entities from a circumscribed partition of the system
Subject	an object identified with taking an action
Supply	consumable and expendable sources of energy essential for the proper functioning of a resource
System	an operational configuration of a set of resources from a circumscribed partition of the universe or configured from system segments
Template	a representation of a set of C2ed objects within a common frame of reference
Theory	a coherent body of knowledge which can justify the utility of a model
Threat	any potential compromise of a capability
Tool	an implementation component which can make direct use of a given equipment (e.g., software driver or a hardware drive)
Tree	a direct (hierarchical) acyclical (loop-free) graph
Type	the syntax/format in which a class or object is specified
Typeface	a graphic representation of a symbol

Universe	the entire Conflict region including all systems: friendly, adversary, and neutral and the environment
Universal truth	aka ground truth, pertaining to coordinates and status of a simulated object, i.e., from the point of view of a simulation controller
Ut object	hasa Ut_Fa object
Value	an admissible instance of a variable
Variable	a dynamic characteristic of an entity
Vectorization	represention of an image by an outline
Vertical interaction	an interaction within a given resource across layers
View	an instance of a perspective
Virtual asset	a process which can exercise control over a physical asset
Virtual resource	a resource limited to selected layers

ANNEX C: OBJECT SPECIFICATION FRAMEWORK

PURPOSE AND SCOPE

The purpose of this annex is to specify a framework for modeling Conflict_Region resources for the purpose of command and control decision-aids and simulations. The material presented herein is intended for use by designers and researchers in developing and analyzing object-oriented specifications for command and control systems in the context of operational scenarios. Evolving specifications of selected objects are intended for incorporation into a Joint DoD library of Decision-Aid/Simulation Application Objects referred to herein as C2ed objects.

This annex is modeled in principle after the reference document by OSI/Network Management Forum entitled Object Specification Framework, Forum 003, Issue 1.0, September 1989. It provides an extension to it in the area of the man-machine interface and in the area of command and control. A key element to understanding this document is to consider the notions associated with managed objects for communications as described in the reference and apply them to managed objects for command and control (C2 objects).

THE OBJECT MODEL FOR C2

There are two superclasses of application-oriented C2ed objects: **Da objects** and **Cr objects**. Cr objects represent and model real-world resources and assets of the universe for command and control. Da objects represent the framework for modeling, creating and activating interrelationships and interactions between and among Cr objects.

Typically Da objects provide frameworks for layered services which are productized to provide observation support (situation monitor/capability assessor), decision support (alternative formulator/selection evaluator) and action support (execution synthesizer/action disseminator) for command and control of Cr objects. As an integral part of an open system architecture, C2ed Objects are layered in accordance with the C2 Reference Model.

There are four superclasses of Da objects which occur at four levels with respect to Cr objects: Fa objects, Ut objects, Se objects and La objects. Fa objects have direct access to Cr object features. Fa objects are constrained to access or activate individual features of Cr objects. Fa objects are therefore level 1 Da objects. Fa objects may be associated with Ut objects, Se objects or La objects which are at level 2, 3 or 4, respectively. Each level n Da object must include at least one level n-1 Da object. A level n Da object may also include one or more level n-2 Da objects. Ut objects have only indirect access to Cr objects through Ut_Fa objects. Ut objects may access or activate a single feature for any subset of Cr objects as constrained only by the configuration setup and workspace available through the supervising Se/La object. Se objects provide direct access to Cr objects through Se_Fa objects and indirect access to Cr objects through Se_Ut objects. Se objects may access or activate a given set of features for any subset of Cr objects as constrained only by the configuration setup and workspace available through the supervising La object.

Examples of Da objects include:

Da_layer (e.g., mission, plan, task, job, assignment, transaction, package)
Da/La_service (e.g., monitor, assess, generate, forecast, evaluate, require, specify)
Da/La/Se_utility (e.g. snapshot, template, overlay, allocate, schedule, associate, synchronize)
Da/La/Se/Ut_facility (e.g., observe, fire, cover, conceal, communicate, move, query, compute)

The hierarchy and nestability of these levels and their correspondence to Cr objects is delineated below:

Da object (Situation_Assessment, CoA_Development, Execution_Monitoring)

1.1 Da_**layer object** (For all Cr objects relevant to a given C2 object layer:
Cr_unit objects, Cr_coordination objects, Cr_environment objects)
1.2 Da_**service object**
1.3 Da_**utility object**
1.4 Da_**facility object**
2.1 Da layer **service object** (Product_requirement, Product_generation,
Product_evaluation, Product_specification)
2.2 Da layer **utility object**
2.3 Da layer **facility object**
3.1 Da layer service_**utility object** (Cr_interaction)
3.2 Da layer service_**facility object**
4.1 Da layer service_utility_**facility object** (Cr object_layered coordinates, status and capabilities)

Each Da object must provide workspace for processing, storage, entry, display and access to its lower level Da objects and their products which may include textual, numeric, tabular, graphic, sound or image content formatted in accordance with its corresponding industry-standard (to be selected).

There are three superclasses of **Cr objects** which are layered in accordance with the C2 Reference Model: **Un objects**, **En objects** and **Co objects**. Un objects represent friend, foe or neutral resources and assets. En_ objects represent the Conflict_Region air, land, sea and space regions including terrain and weather. Co objects represent objectives and constraints in space and time which must be enforced in mapping/overlaying Un objects to En objects in a manner which is consistent with the Un object capabilities and command and control. Logical interactions between Cr_Un objects are always mitigated through Cr_Co objects and Cr_En objects. Logical interactions must occur in a layered fashion at each of the seven layers of Cr objects or at an aligned aggregation thereof.

Examples of Cr objects include:

Cr_unit (e.g., division, brigade, company)
Cr_environment (e.g., region [e.g., hill, river, urban, field],
structure [e.g. bridge, obstacle])
Cr_coordination (e.g., line of-departure, -contact, phase-, delay-line, objective area)

The hierarchy and nestability of these levels of aggregation is delineated below:

Cr object (Unit, Coordination, Environment)

1. Cr_**unit** object (associated with 'action' flow)
1.1 Cr_**unit_asset** object
1.2 Cr_**unit_port** object
1.3 Cr_**unit_package** object
1.4 Cr_**unit_action** messages

2. Cr_**coordination** object (associated with 'control' flow)
2.1 Cr_**coordination_asset** object
2.2 Cr_**coordination_port** object
2.3 Cr_**coordination_package** object
2.4 Cr_**coordination_control** messages

3. Cr_**environment** object (associated with 'event' flow)
3.1 Cr_**environment_asset** object
3.2 Cr_**environment_port** object
3.3 Cr_**environment_package** object
3.4 Cr_**environment_event** messages

An example of a Ut object product depicting each of the above Cr objects through a common display framework is shown in Figures An.C.1, An.C.2, An.C.3, and An.C.4.

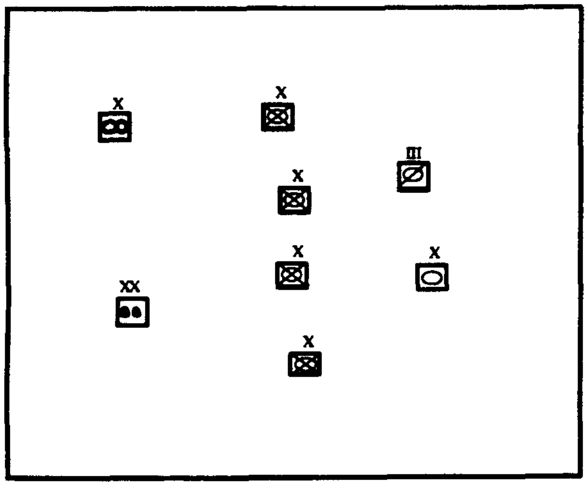

Figure An.C.1. An overlay of Unit objects

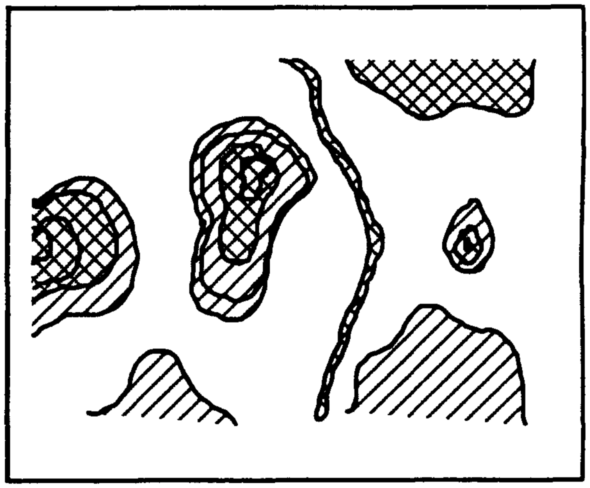

Figure An.C.2. An overlay of Environment objects

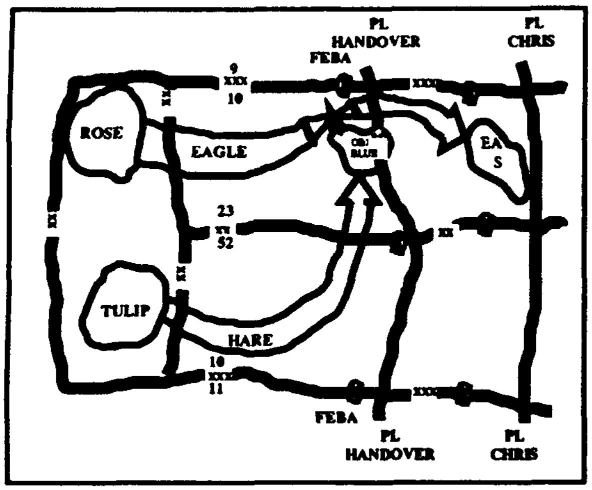

Figure An.C.3. An overlay of Coordination objects

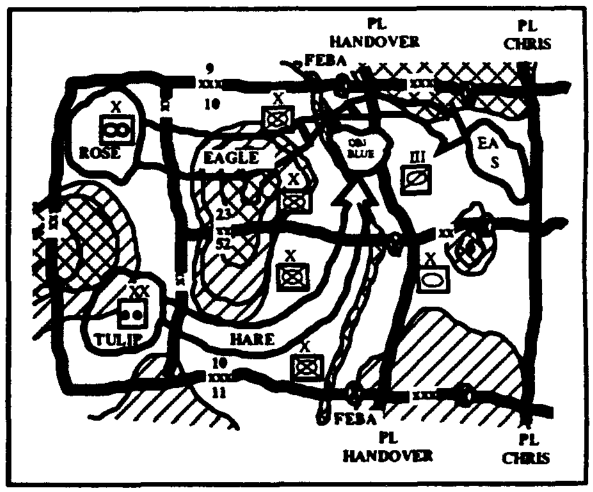

Figure An.C.4. An overlay of Conflict region objects

Examples of Cr_* objects where * ::= Un/En/Co include:

Cr_*_conflict (Mission -state/restate [Peace, War, Campaign, Battle, Combat, Engagement, Armament])
Cr_*_presentation (Plan -CoA template, -resources allocate/deallocate)
Cr_*_operation (Task -schedule/synchronize, -activate/deactivate)
Cr_*_procedure (Execute -assemble/disassemble, -deliver/position)
Cr_*_network (Assign -connect/disconnect, -enable/disable, -route)
Cr_*_link (Transact action/event -transmit/receive, -depart/arrive, -look/see, -fire/hit, -intercept/vector, -detect/jam)
Cr_*_asset (Port/Package-communications, -transportations, -identifications, -inflictions)
Cr_*_port (transceiver, vehicle, sensor, weapon)
Cr_*_package (message, cargo, image, ordnance)

Note that each C2ed object provides at least one key product.
An object at one level supports one or more objects at the same or higher level.
An object at one level coordinates with one or more objects at the same level.

CR_INTERACTIONS DIAGRAMS

Consistent with the OO message passing paradigm, Cr objects interact through the generation and flow of three categories of messages: 'action,' 'control,' and 'event' messages. Cr_interactions are best portrayed by object interaction diagrams (OIDs). An example of an OID depicting these messages is shown in Figure An.C.5.

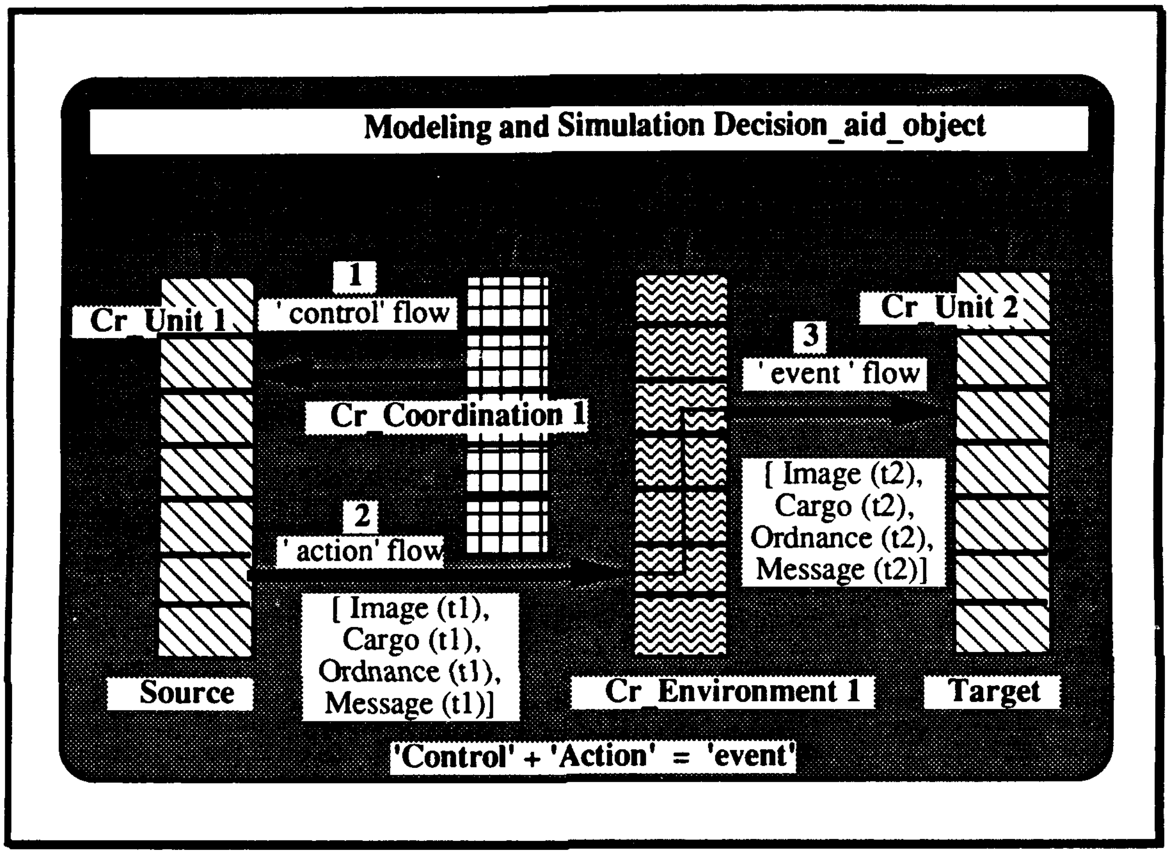

Figure An.C.5. A Cr object interaction diagram (OID)

'Action' flow: Cr_unit objects create package objects as message objects which provide for the initial 'action' flow to potential recipient Cr objects. 'Action' flow messages are received by Cr_environment objects in the path of the Cr_package object.

'Control' flow: Cr_coordination objects create package objects as message objects which act as 'filters' to 'control' the flow of 'action' in the Conflict region. 'Control' flow messages are received by Cr objects associated with the originating Cr_coordination object.

'Event' flow: Cr_environment objects also create package objects which act as 'modifiers' to create 'events' as received by other Cr_unit objects. An 'event' may be created by a Cr_environment object as a natural phenomenon or in response to an 'action' message enabled by a Cr_coordination object.

C2ED OBJECT SPECIFICATION FORMAT

Object specification requires that provisions be made for Object Identification, Classification, Abstraction, Polymorphism, Inheritance and Hierarchy. The following formal structure applies to C2ed objects:
(examples shown in () may pertain to a Cr object, or a Da object, or both, as appropriate)

Ob_header features (attributes including abstract /concrete object characteristics)
- Ob_class (e.g., Class-, Subject-, Block-Name: 'nth ID')
  - Cl_catalog (Da_layer, -service, -utility, -facility,
    Cr_unit-, -environment-, -coordination-,
    -package, -port, -asset, -link, -network,
    -procedure, -operation, -presentation, -conflict)
  - Cl_associate (Cl_super: 'Corps', Cl_subordinate: 'Bde', Cl_peer: 'Div')
  - Cl_inherit (Cl_feature to be shared from other Ob_classes)
  - Cl_instantiate (Ob_feature selected for implementation)
  - Cl_model (model describing dynamic behavior of Ob_abstract_attributes)
- Ob_name
  - Nm_register (ASN.1)
  - Nm_bind
  - Formal_name identifier (Unique-Label/Handle: '35th ID')
  - Name_modification (Group Reference: 'InfDiv')
  - Alias(Nickname)
  - Nomenclature
- Ob_explanation
  - Objective
  - Source of object information
  - Use/Usage (operational)
  - Importance
- Ob_Description
  - Multimedia
  - Image
  - Signature
  - Symbol
- Ob_define
  - Ob_model (model describing dynamic behavior of Ob_concrete_attributes)

Typically, each class inherits all of the Ob_features of Ob_classes identified by Cl_super and adds its own unique Ob_features. Thus, the Ob_features of the Ob_classes named by Cl_super need not be repeated in each Ob_Class. However, Cl_inherit allows an Ob_class to be set up with specific Ob_features for inheritance from any of the Ob_classes identified in Cl_associate.

C2ed object Specifications Format (Continued)

Ob_Body features (attributes/operations pertaining to concrete object characteristics)
- Ob_attribute----may include "universal truth" and/or "perceived truth"
  - Coordinates(attributes)
    - At_datetime (Greenwich Mean Time, D-day/H-hour/M-Minute/S-Second)
    - At_location/position
    - At_direction/orientation
  - Status (corresponding to capability operations)
- Ob_operation
  - Capabilities (corresponding to status attributes)
    - Products (for each layer as required)
    - Carrier (for each port/package)
    - Target (for each asset/port/package)
    - Environment (constraints for each port/package)
    - Capacity (for each port/package)
    - Power (for each port/package)
    - Rate (for each port/package)
    - Range (for each port/package)
    - Response_time (for each port/package)
    - Accuracy (for each port/package)
    - Reliability (for each asset/port/package)
    - Availability (for each asset/port/package)
    - Maintainability (for each asset/port/package)
    - Survivability/Protection (for each asset/port/package)
  - Op_create (New Instance, Initialize Instance, Name/Rename Instance)
  - Op_retrieve (Open, Load, Decompress from Archive)
  - Op_display (Background, Overlay, Flash [Alert], Print, Plot, Animate)
  - Op_edit (Cut-, Copy-, Duplicate-, Paste-, Modify-Subordinate Object)
  - Op_format (Comment[Text]-, Draw [Vector]-, Paint [Raster]-Subordinate Object))
  - Op_share (Update-, Distribute_to-C2ed objects)
  - Op_save (Replace, Store New, Dump, Compress to Archive)
  - Op_clone (Duplicate, Copy)
  - Op_change (Get-, Set- Attribute, Select _ob_attribute)
  - Op_activate (Start-, Stop-, Resume-, Interactive-, Allocate-, Deallocate-ob-attribute)
  - Op_notify (Report-event, -result)
  - Op_deactivate (Close, Suspend)
  - Op_delete (Memory, Storage)
  - Op_transfer (Transfer control of active view to another C2ed object)

Ob_implement (Mandatory/Optional/Mandatory Conditional/Optional Conditional)
Ob_use (Instances incorporated in known implementations)

Note that above terms are overloaded, i.e., the Cr object ID will determine the context of the Cr object Header and Body features. Recall that a 'feature' may be either an 'attribute' or an 'operation.' Also note that a 'capability' is a method or a set of methods which update 'status.' A 'status' is an attribute or a set of attributes which are realized or realizable by a capability as a function of space and time. In other words 'status' is an actual or projected outcome of a 'capability.'

CR OBJECT TEMPLATES AND CONTAINMENT DIAGRAMS

Cr object templates provide, as a minimum, four aggregation templates to put into context the wide variety of units, assets, ports, or packages. These levels of aggregations are shown in Figure An.C.6. The following set of ID_templates applies to Cr objects where the character '*' is replaceable by one of the following strings: 'Unit,' 'Coordination,' or 'Environment.' Note that any Cr object may be decomposed or aggregated along any set of contiguous layers as described in the C^2RM. Other decompositions will clearly result in additional libraries.

*_template identification:

*_ID

Superior_*_ID

Subordinate_*_ID/*_asset_ID

1. *_asset_template identification:

    *_asset_ID

    *_ID

    *_port_ID

2. *_port_template identification:

    *_port_ID

    *_asset_ID

    *_package_ID

3. *_package_template identification:

    *_package_ID

    *_port_ID (if nested)

    *_Subordinate_port_ID. Note that a package may have internal ports (e.g., missile may have a homing sensor or a guidance transceiver; similarly a mine may have a proximity sensor or transceiver for remote control).

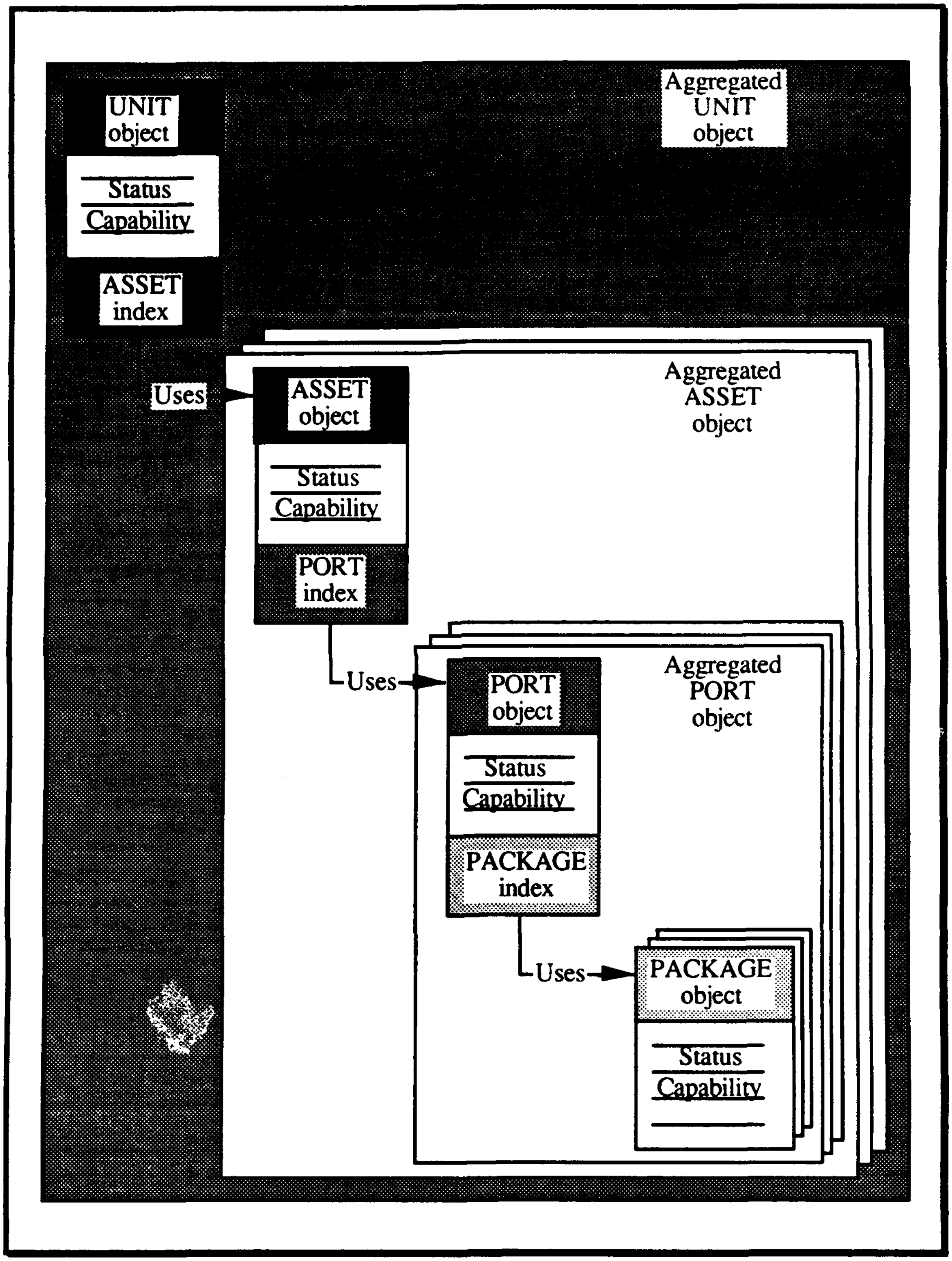

Figure An.C.6. A Cr object containment diagram (OCD)

NAMING

The name of an object is an integral part of its definition. A name is a linguistic construct which corresponds to an object in some universe of discourse [ISO DIS 7498-3, Naming and Addressing]. Names of C2ed objects may be ambiguous when taken out of context. Names of objects are therefore defined in a formal manner through the use of a naming tree. The naming tree provides the hierarchy of superior and subordinate C2ed objects that constitute the naming structure for the definition of an object. Names may be distinguished through the use of a sequence of the relative distinguished names of the C2ed object and each of its superior instances, but not necessarily all the way back to the root. Some names that may be perfectly understood in one context of one application may be highly controversial in the context of other applications. Clearly there are many more real-world objects than one can realistically model. Therefore abstract object models serve to reduce the number of objects to a manageable number. Naming of abstract objects should therefore not be confused with names of real-world objects but considered only for the purpose of identifying a group of features which make up an object. Concrete objects serve to mitigate ambiguity which may be inherent. The ultimate alignment between concrete objects and real-world objects depends upon the level of abstractions one is willing to accept. Therein lies the reason for the C2 Reference Model. Naming of abstract or concrete objects is a compromise intended to minimize ambiguity with real-world objects by minimizing duplications of features across classes of C2ed objects and maximizing their coverage of the C2 problem domain.

RELATIONSHIPS TO OTHER STANDARDS

OSI/NM FORUM, Object Specification Framework, Forum 003, Issue 1.0, September 1989.

ANNEX D: CONFLICT REGION OBJECT TAXONOMY

PURPOSE AND SCOPE

The purpose of this annex is to provide a generic taxonomy for use in generating more domain specific taxonomies or for comparing taxonomies of related domains. A generic taxonomy is useful to ensure complete, coherent and consistent coverage and treatment of requirements and capabilities within a more specific application. Many items are accounted for explicitly (e.g., unit objects at different echelons, natural environment), others are accounted for implicitly (e.g., headquarters facilities are C2 resources which are part of a distributed unit object; plans, orders, and report objects are C2 products which are part of a layer of a unit object). Of key importance to understanding both specific and generic taxonomies is the primary reason for existence or usage of an object. An environment object (e.g., river) is primarily defined to represent the mesofeatures of the environment (e.g., river depth and river width) independent of many other secondary usages such as for coordination (e.g., phase line, or line of contact) or as an obstacle (e.g., no-go / slow-go area). Due to multiple inheritance capabilities, classes/objects may be highly complex (i.e., objects may wear multiple "hats"). To prevent confusion and misunderstandings this basic reference taxonomy is therefore constrained to support the definition of classes/objects with their primary (salient) features. The generic taxonomy can therefore be used and reused as a basic building block for derived objects / classes, which in turn will be capable of representing significantly more complex hybrid objects.

ENVIRONMENT OBJECTS

Ground (aka land)	
macrofeature (topographic, geographic):	elevation, temperature, moisture, ...
mesofeature (terrain)	mountain, valley, wadi, gap, plain, plateaux, karst, canyon, vegetation, metropolitan, urban, rural , village, town, city, road (street, highway, expressway), field, railway, canal, swamp, marsh, island, atoll, reef, harbor, beach, shore, coast, ...
microfeature(structure)	rock, tree, building, bridge, cave, tunnel, pole, dam, doline, passage, shaft, fissure, debris, berm, ditch, animal, ...
nanofeature	window, branch, stalagmite, stalagtite, wire, particle, ...
Surface (aka sea)	
macrofeature (hydrographic, oceanographic)	depth, temperature, current, tide, ...
mesofeature (water)	wave, iceberg, lagoon, littoral sea, lake, channel, river, stream, aquaduct, rapids, falls...
microfeature	whirlpool, fish, ...
nanofeature	fin, particles
Atmosphere (aka air)	
macrofeature (climatographic, meteorographic)	altitude, temperature, moisture, wind, ionization, pressure, pollution, ...
mesofeature (weather, man-made)	cloud, storm, fog, smog, dust, fire, jet stream, smoke, aurora borealis, ...
microfeature	whirlwind, bird, ...
nanofeature	feather, particles
Space (aka sky)	
macrofeature (astronomic)	radiation, gravitation, magnetization
mesofeature (galactic)	galaxy, constellation, star, solar system, sun, planet, comet, moon, orbital ring, ...
microfeature	meteor, asteroid, ...
nanofeature	fragments, particles

UNIT/RESOURCE/ASSET/PACKAGE OBJECTS

Unit (echelon, organization [command resources, subordinate units])

Echelon	subordination of units
Army	corps, division, brigade/regiment, battalion/squadron, company/battery/troop, platoon, squad/crew, soldier
Navy	operational fleet, task force, task group/battle group, task unit, task element, crew, sailor
Air Force	theater air force, wing, squadron, crew, aviator
Marine Corps	amphibious ready group, expeditionary brigade/regiment, battalion/squadron, company/battery/troop, platoon/detachment, squad/crew, marine
Unit Role	ground, surface, subsurface, air, amphibious, space
Conflict	operations, e.g., Warfare, Combat
Conflict support	operations support, e.g., Warfare Support, Combat Support
Conflict Service Support	operations service support, e.g., Warfare Service Support, Combat Service Support
Mission Organization	aka task organization
Command Resources	Resources capable of generating missions, plans, and tasks, aka operational facilities, e.g., Headquarters, Command Centers, Command Posts, Tactical Operations Center, Fire Direction Center, Traffic Control Center, Control Station
Subordinate Units	Units with assets available for tasking by command resources

C2 LAYER TAXONOMY

Layer 7 — **C2 Conflict**, Policy (aka doctrine), Mission (aka command)
Peace, War, Campaign, Battle, Combat, Engagement, Armament
Authorize to use assets/packages

Layer 6 — **C2 Presentation**, Strategy, Plan
Plan to allocate, schedule, generic subordinates/assets/packages

Layer 5 — **C2 Operation**, Tactic, Task (aka order)
Task to synchronize specific subordinates/assets/packages

Layer 4 — **C2 Procedure**, Schema, Job
Train assets with schema

Layer 3 — **C2 Network**, Discipline, Assignment
Coordinate to ensure asset discipline

Layer 2 — **C2 Link**, Technique, Transaction
Prepare to instruct assets

Layer 1 — **C2 Asset**, Instruction, Package
Vehicle, Weapon, Sensor, Transceiver
Activate assets, e.g., base (w /people), tank, fighting vehicle, armored personnel carrier, troop, bomber, fighter, ship, carrier, cruiser, frigate, submarine, barge, lighthouse, beacon, battleship, destroyer, vessel, craft, tanker, boat, ...

PRODUCT TAXONOMY

Requirement — command, order, request, e.g., mission, plan, task, job, directive, assignment, warning, schedule, fragment / fact thereof, ...

Status — report, e.g., situation, estimate, spot, mission, operation, status, intelligence, detonation, casualty, transportation, damage assessment, fragment / fact thereof, ...

PORT LAYER TAXONOMY

Port Interaction	identification | infliction (impression) | transportation | communication
Port Package	image | ordnance | cargo | message
image	records of audio, video, signal, signature, data, photo, sighting facts
ordnance	destructive/explosive, preventive/barrier/obstacle impact
ammunition	missile, torpedo, shell, bullet, bomb, rocket, grenade, mine, ...
signal	interference
communication	jamming
identification	glare, smoke, chaff
cargo	supply, item, equipment (materiel or personnel), tools, object media
message	any productized fact, or set of facts, e.g, fire mission, mission plan, tasking order, track report, spot report, jam report, ...

Port Application Vehicle | Weapon | Sensor | Transceiver

Layer 7 **Application**

Authorize to use port/packages

Layer 6 **Presentation**

Plan to allocate, schedule, generic port/packages

Layer 5 **Session**

Task to synchronize specific port/packages

Mass / Fuse Packages

Layer 4 **Transport**

Internet / Bridge ports

Layer 3 **Network**

Interconnect Ports, **Route** packages

Layer 2 **Link**

Prepare package for interaction

Layer 1 **Physical**

Send / Receive package

COORDINATION OBJECTS

A coordination object is required to either enable or disallow any interaction in the conflict region. Coordination objects are context sensitive. The context is defined hierarchically through the C2 layers and Port layers requirement for coordination among C2 resources.

Volume	a scalable geometric object defined by a surface enclosing a conflict region, e.g., air corridor, airfield, airport, seaport, aerial avenue of approach, area of operation (air), radar sector, landing zone (airbase, air raid), drop zone, ...
Area	a scalable geometric object defined by a perimeter enclosing a conflict region, e.g., minefield, assembly area, built-up area, parking lot, anchorage, area of operation (ground, sea/ship, amphibious), zone of operation (offensive), sector of operation (defensive), beachhead, transit area, named area of interest, sector of attack, battle position, limit position, objective area, security zone, engagement area, area of approach, no fire area, depth curves, elevation curves...
Line	a scalable geometric object defined by a path traversing a conflict region, e.g., track, trajectory, phase line, front edge of the battle area (FEBA), front line of own troops (FLOT), line of departure, line of contact, line of fire, unit border, unit boundary (i.e., front, right flank, left flank, rear), fire support coordination line, line of communication, transportation lane, avenue of approach, axis of attack, security perimeter, route, line of position,...
Point	an unscalable geometric object defined by a highly localized subregion within a conflict region, e.g., target reference point, coordination point , observation point, start point, release point, way point, refueling point, control point, point of interest, buoy,...

ANNEX E: NOTATION, ACRONYMS, AND ABBREVIATIONS

- ∀ — asset
- ⊥ — port
- ! — response indicator
- ∪ — nesting indicator
- & — adjacent items are required jointly
- (...) — mandatory nesting of one or more enclosed items
- -- — LHS/RHS constitutes first/last or lowest/highest value in the range of values admissible for associated variable
- .L — layer/sublayer L
- .L — psd indicator
- / — adjacent items are alternatives which are admissible equivalents or aliases
- : — instantiation/specialization/is/alias indicator
- ::= — LHS is composed of/synthesized from RHS
- := — LHS is set /assigned/gets the value(s) provided by evaluating the RHS
- ; — item separator/terminator in a list of items
- <= — LHS is deaggregated from the RHS
- <...> — conditional nesting of enclosed items
- = — LHS is equivalent to RHS
- =: — LHS inherits from RHS
- =:: — LHS contains RHS
- => — LHS is an aggregation of the RHS
- ? — interrogation or request indicator
- [[i]] — the ith applicable document as per annex
- [i] — the ith reference as per appendix
- [...] — optional nesting|sequencing of enclosed item
- _ — underscore indicates pertinence to preceding text
- A — action
- a — Identification Indicator
- aka — also known as, taxonomically equivalent

Ap	application
ASCII	American Standard Code for Information Interchange
asn	assigned
asn	assignment
ast	asset layer
At	attribute
$\alpha\rightarrow$	α logical request
$\alpha\Rightarrow$	α physical action
b	transportation indicator
$\beta\leftarrow$	β logical response
c	communication indicator
C2	command and control aka C Square/C^2
C2ed	commanded and controlled aka C Squared/C^{2ed}
C3	command, control and communication, aka C2/C^3/C Cube
C3I	command, control, communication, and intelligence, aka C2
C4	command, control, communication, and computers, aka C2
C:	C class
Cl	class
cmp	composed
Cnf	conflict
Co	coordination
CoA	Course-of-Action
commo	communication
Cr	Conflict region
Δ	domain
D	decision
d	infliction indicator
Da	decision-aid
DSB	detailed storyboard
E	environment
EP	evaluated prototype
eqpt	equipment
eqv	equivalent
ETE	end-to-end
expr	experience
F	friendly
Fa	facility

FDRS	functional description and requirement specification
FM	field manual
G	foe/adversary
H	neutral
hasa	"has a," partonomically related to include or 'is composed of'
i	an index
ID	identification
IDD	interface design document
idntf	identification
IFF	identification friend or foe
inflc	infliction
info	information
inh	inherits
intr	interaction
IRS	interface requirement specification
isa	"is a," taxonomically related to classification, categorization or type
itd	implementation/technology domain
j	a positive integer
k	an integer
ka	knowledge acquisition (not to be confused with Knowledge)
knlg	knowledge
L	an ASCII capital letter
La	layered application
lnk	link
M	any one of the T base layers
m	any one of the services within M
MSG	message
msn	mission
N	any one of the Σ application layers
n	any one of the services within N
ntw	network
O	observation
Ob	object
OO	object-oriented
OOA	object-oriented analysis
OOD	object-oriented design
OOM	object-oriented methodology

OOP	object-oriented programming
Op	Ob_operation
opr	operations layer
OSI	Open System Interconnection
pck	package
phs	physical layer
pln	plan
prd	product
prc	procedure layer
prs	presentation layer
prt	interaction port
PSB	preliminary storyboard
psd	problem/solution domain
R	real resource
rqr	requirement
$\rho\Downarrow$	ρ requirement
Σ	problem/solution domain
SDD	system design document
Se	service
SOP	standing/standard operating procedure
SPEC	specification
sppl	supply
SPS	system product specification
SRS	system requirement specification
SSDD	system segment design document
ssn	session layer
SSS	system segment specification
ST	student text
STD	standards
stt	status
$\sigma\downarrow$	σ interaction source (outgoing/action port)
$\sigma\Uparrow$	σ status
T	implementation/technology domain
TC	training circular
trn	transport layer
trnsp	transportation
trs	transaction

tsk	task
τ↑	τ interaction target (incoming /event port)
U	universe
Un	Cr_unit
Ut	utility
V	virtual resource
VDD	version description document
VR	Virtual reality
X	resource
Y	interaction
Z	conflict
x	a variable defined by local context
{...}	optional nesting | sequencing of at least one or more of enclosed items
|	adjacent items are mutually exclusive alternatives which are admissible but not necessarily equivalent
↔π	logical peer-to-peer interaction π
...	a set of zero or more items
–>	implies
⇔γ	*physical action-event interaction* γ

APPENDICES

The following appendices do not form a part of of the C^2RM.

APPENDIX A: NOTES

ISSUES

The generic notions inherent in the C^2RM are relative in nature and as such will assume different realizations. Illustrative examples as well as practical applications may be developed by addressing the following key issues:

a) What is the mapping between a specific C^2 system and the C^2RM?
b) Which observables are generic and which are specific to a given function?
c) What are the observables of key importance?
d) Which observables must be explicitly communicated among resources?
e) Which observables must be explicitly communicated between adjacent layers?
f) What is the mapping between observables and internal variables?
g) Which layer is concerned with a given observable?
h) What measures-of-performance apply to a given layer?
i) What are the detailed isomorphic features among the four types of interactions?
j) What are the generically unique features of each of the four types of interactions?

CONCLUSIONS

1. The C^2RM provides a coherent and complete, layered reference model essential to integrate, leverage and exploit many techniques and capabilities which are evolving independently or competitively for intelligent C^2 systems.

2. The C^2RM provides a formal baseline for a commonly understood vocabulary and conceptual framework for analyzing, designing, or evaluating command and control systems.

3. The C^2RM provides many important ingredients which span the essential elements of C^2. Such a model can play a key role in leveraging application of emerging disciplines to C^2 problems in a coherent way.

4. The C^2RM provides the breadth and depth required of a formal framework to achieve an open system architecture for compatibility and interoperability among independently developed layered C^2 services. This conclusion is based on the premise that the C^2RM is analogous to the ISO OSI RM which is rapidly gaining acceptability as a vehicle for achieving compatibility and interoperability between independently developed layered communications services.

RECOMMENDATIONS

1. The C^2RM should be used in research of C^2 to provide a common framework for understanding and analyzing C^2 systems.

2. The C^2RM should be used in development of C^2 to provide a common framework for designing and synthesizing C^2 systems.

3. The C^2RM should be used to describe organizations of C^2 systems at any echelon, in evaluating their measures of performance and measures of effectiveness, and in improving their design.

4. The C^2RM should be used to develop interoperability standards. It is particularly suited to describe military systems which are layered hierarchically, their mission-oriented resource allocation structures, their interfaces, interactions and integrated operations.

5. The C^2RM should be used as a framework to develop large-scale, distributed C^2 models, simulations and war games.

6. Proponents for C^2 systems should collaborate to establish and agree upon a final version of the C^2RM to be used in improving training, analyses, tests, and evaluations and in guiding research and development of realistic applications of evolving C^2 concepts and requirements.

7. The C^2RM should be refined to include formal description techniques to provide a common tool for understanding of detailed analyses and design of specific applications.

8. The C^2RM should be reviewed throughout DoD, NATO and other allies to ensure that concepts, requirements, specifications and implementations are acquired in an organized and affordable fashion by protecting existing investments, minimizing duplication of effort, and guiding technology to solve applications more efficiently in a modularly open and interoperable fashion.

APPENDIX B: REFERENCES

PREVIOUS VERSIONS

1. The C³RM for Research and Development, Experimentation and Testing, Analysis and Design, Education and Training, Coordination Draft #1, 9 November 1987.

2. The C³RM for Research and Development, Experimentation and Testing, Analysis and Design, Education and Training, Coordination Draft #2, 26 January 1988.

3. The C³RM, Coordination Draft #3, 26 January 1988.

4. The C³RM, Coordination Draft #4, 29 August 1988.

5. The C²RM, Coordination Draft #5, 8 March 1990.

6. The C²RM, Coordination Draft #6, 11 May 1992.

BACKGROUND PAPERS AND DOCUMENTS

1. ISO 7498(Draft) Information Processing Systems - OSI Basic Reference Model, 1984. Also see Uyless, B, ISO OSI a Model for Communications, Prentice Hall, 1991, and Jain, B. N., and Agrawala, A. K., Open Systems Interconnection, Its Architecture and Protocols, McGraw-Hill, NY, ©1993.

2. Soley, R. M., (ed.), Object Management Group (OMG) Standards Manual, Draft 0.1, OMG TC Document 90.5.4, May 1990, OMG, Framingham, MA. Also see G. Booch, Object-Oriented Design with Applications, Benjamin/Cummings, Redwood City, CA, ©1991.

3. OSI/Network Management Forum, Release 1 Package,1990, (includes Protocol Specifications, Application Services, Object Specification Framework, Architecture, Glossary, Library of Managed Objects, Name Bindings and Attributes, Managed Objects Naming and Addressing, Conformance Requirements, and Shared Management Knowledge), Bernardsville NJ.

4. U.S., Department of Defense, Report of the Defense Science Board Task Force on Command and Control Systems Management, Office of the Under Secretary of Defense, Research and Engineering, Washington, July 1978, pp. 1, 7, 11.

5. Monteleon, V., "Distributed C³ Test Bed and User Interactions," SIGNAL, March 1987, pp. 85-91.

6. Snyder, F. M., Command and Control: Readings and Commentary, A Publication of the Program on Information Resources Policy, Harvard University and the Center for Information Policy Research, Cambridge, MA, April 1989, p. 124.

7. Booch, G.,"Object-Oriented Development," IEEE Transaction on Software Engineering SE-12(2), (1986), pp. 211-221.

8. The 1988 CASE Industry Directory, CASE Computing Consulting Group, Inc, March, 1988.

9. White, I., "Raising the Standards in C²," IEEE International Conference on Command Control and Communications, May 1989, Bournmouth England.© Controller HMSO, London 1988.

10. Layman, G. E., and Robins, J. R., USN CPT (Ret.), "Open Architecture for Modern Command and Control Systems," SIGNAL, August 1988, pp. 83-89.

11. Mayk, I., Rubin I., "Paradigms for Understanding C3 Anyone?" in Johnson, S.E., Levis, A.H., (eds.), Science of Command and Control: Coping with Uncertainty, National Defense University/AFCEA International Press, Washington, DC, 1988, pp.48-61.

12. Carnegie Group, Inc., Digital Equipment Corporation, Ford Motor Company, Texas Instruments Incorporated, and U S West, Inc., The Initiative for Managing Knowledge Assets (IMKA) Software Functional Specification, Phase 1 Knowledge representation, Draft #2.1, ©July 1990.

13. Holmes, J.E. , Morgan, P.D., "A Layered Approach to the Representation of Naval Command Systems," 2nd IEE Conference on Advances in C3I, England, April 1987.

14. Adapted from the works of WW I British MG J. F. C. Fuller. See Summers, H. G.Jr., On Strategy II. A Critical Analysis of the Gulf War, Bantam Doubleday Dell Publishing Group, Inc., NY, ©1992.

15. Boyes, J. L., Andriole, S. J., (eds.), Principles of Command and Control, AFCEA/SIGNAL MAGAZINE C3I Series; v.6, AFCEA International Press, Washington, DC, ©1987.

16. Leonhard, R.R., The Art of Maneuver: Maneuver-Warfare Theory and AirLand Battle, Presidio Press, Novato, CA, ©1991.

17. Antal, J.F., Armor Attacks: The Tank Platoon: An Interactive Exercise in Small-Unit Tactics and Leadership, Presidio Press, Novato, CA, ©1991.

18. Maggelet, T.F., Walker, R.P., Technical Standards for Command and Control Information Systems (CCISs), Edition 3, Institute for Defense Analyses (IDA) Paper P-2686, ATCCIS Working Paper 25, IDA Log No. HQ 92-40762, January 1992.

19. Bull, J. A., and Holmes, J. E., "An ODP Model for Military IDA Systems," Proceedings of the IDASCO'92, an IEE International Conference Publication No. 353, pp.133-137, April 1992.

20. Military Standard (Final Draft): Protocol Data Units (PDU) for Entity Information and Entity Interaction in a Distributed Interactive Simulation (DIS), University of Central Florida (UCF) Institute for Simulation & Training (IST) Report No. IST-PD-91-1, October 1991.

Please obtain references from the sources listed.

APPENDIX C: ACKNOWLEDGEMENT

Following disseminations to many potential contributors, written contributions, endorsements, concerns, and constructive comments were incorporated from the individuals acknowledged below. Many of these individuals took time out of their valuable schedule to share their perspectives on C^2 in general and on the C^2RM in particular. This list accumulated since the initial draft of the C^2RM. As such, the affiliation of each individual corresponds to the organization they served at the time in which their comments were received. It is hoped that this list will grow as this model is proliferated and the opportunity for its use avails itself with time. If any name is omitted it is unintentional. Please forward any correction as per form on last page of this document.

GOVERNMENT

Baldwin, USA LTC Russell,TRADOC CAC-CD FBL
Beane, Eli, CECOM C^2 and Systems Integration Directorate
Buddenberg, LCDR Rex A, Coast Guard
Chamberlain, Dr. Sam, ARL
Chappel, LTC Oscar A., TRADOC CAC-CD TPIO
Chausa, Bert, CECOM FBL Technology Assessment Center
Coe, COL Gary Q., USA (Ret.), JCS
Cohen, Herbert, AMSAA, BRG Associate Member and C^2 RM Subgroup Member
Conwell, Candace, NRaD
Cronin, Terry, CECOM SWD
Dasaro, Joseph, CECOM C^2 and Systems Integration Directorate
Dockery, Dr. John, JCS
Drillings, Dr. Michael , ARI
Effenberger, Frank, CECOM, C^3 Systems Directorate
Emmerman, Dr. Philip, ARL
Fallon, Dr. Robert J., JCS, C^3 Systems Directorate
Famolari, Eugene, CECOM, C^3 Systems Directorate
Fastlich, Alan, CECOM C^2 and Systems Integration Directorate
Fogg, David, AUSTRALIA DoD DSTO
Foster, John, CECOM MCS
Gaggin, David, CECOM C^2 and Systems Integration Directorate
Gandee, MAJ Patrick L, USAF, JCS, J6A
Gandy, MAJ Chris, U.S.Army Military Academy, West Point
Goldberg, Larry, C2RM Associate Editor, CECOM STINFO
Goodbody, Richard, NOSC, (GS Ret.)
Goodman, Dr. Irwin, NOSC, BRG Associate Member and C^2RM Subgroup Member
Greenlee, Patrick, NRaD
Hamilton, Mark, CECOM FBL Technology Assessment Center
Hartel, LTC R. J., USA, JCS J6I
Harville, MAJ James B, USMA
Hiniken, Dr. Paul, DISA
Hirschberg, Morton, LABCOM BRL
Hislop, Dr. David, US Army Research Office
Hitchcock, COL Dickey, USA (Ret.), CECOM, Acting Director, SEID

Hofer, Dr. Ronald C., PEO STRICOM
Holmes, John E., UK DRA
Ingber, Dr. Lester, NPS BRG investigator
Jackson, Vairzora, CECOM FBL Technology Assessment Center
Jones, Dr. Carl, NPS, BRG Member and C^2RM Subgroup Member
Kaste, Richard, LABCOM BRL
Kessler, Otto, NAWC AC Div
Klein, Dr. Theodore, CECOM, Center for C^3 Systems, (GS Ret.)
Knudson, MG Wayne C., USA, TRADOC CACDA
Lynch, Jim, NAVSWC
Meshal , Mohamed, CECOM C3SD
Mollenhauer, Clark, RADC, Air Force BRG Member
Mudurian, Mike, NOSC, Navy BRG Member and Chairman
Nolan, Tom, USAMSAA
Plant, Jack, CECOM, C^3 Systems Directorate
Preston, Vairzora, CECOM Technology Assessment Center
Pucilowski, Jr., Joseph J., CECOM, C^3 Systems Directorate
Rebbapragada, Venkata (Lakshmi), CECOM C^3 Systems Directorate
Reyenga, MAJ Robert, USA, TRADOC CAC-CD-FBL
Robson, MAJ Gordon W., SED, Tactical C2 and Training Systems
Ruptier, G.L., NRaD
Seltzer, David, CECOM RDEC
Sherman, LTC Edward, CECOM
Sherman, MAJ George, CECOM C^3 Systems Directorate
Sobolewski, Dr. Vic, AUSTRALIA DoD DSTO
Sovereign, Dr. Mike, NPS/SHAPE Technical Center
Tognola, Edmund, CECOM C^2 and Systems Integration Directorate
Turnage, James H.. DISA/JIEO
Vineberg, Dr. Maniel, NOSC
White, Dr. Ian, Directorate of Science, MoD, England
Williams, Eli, DISA
Williams, MAJ Ben, USA (Ret.), CECOM
Woffinden, LTC, Steve,USA, HQDA, ODISC4

INDUSTRY AND ACADEMIA

Antonisse, Jim, MITRE
Asok, Dr. Ray, Penn State University
Ayalon, BG Avraham, IDF (Ret.), Military Conflict Institute
Bell, LTC Ken, USA (Ret.), Theta Corp.
Billig, Lew, Rand Corporation
Bloom, Leon, Consultant/SAIC
Breinig, Robert, Melpar Div., E-Systems
Bull, John, ANSA Architecture Projects Management Ltd.
Chadwick, Dennis H., OR/C^3I Systems Engineering, General Dynamics
Chapel, LTC Oscar, USA (Ret.), Titan Corp.
Clare, Dr. Loren, Rockwell Science Center, Rockwell
Conroy, Larry, Melpar Div., E-Systems
Craighill, Dr. Earl, Information Sciences and Technology Center, SRI
Cross, Dr. Gene, MITRE
Dalton, John, Booz-Allen and Hamilton Corp.
Davis, David, George Mason University
Davis, Tom, Lockheed Austin Division
Dixon, Herb, BK Security Corp.

Donnell, Victor, Lockheed Austin Division
Dorn, Norman, AT&T, formerly with ND & Associates, Inc.
Dupuy, COL Trevor, USA (Ret.), The Dupuy Institute
Dworkin, Dr. Larry, Consultant
Edleson, Brenda, BK Security Corp.
Edwards, Dr. Jack L., AI Mgmt. & Dev. Corp
Feld, Philip, Director, C³I Programs, Defense Systems Inc. (DSI)
Fisher, Dr. Ben, C2 Laboratory, Hughes Aircraft Company
Frankel, Dr. Michael S., Information Sciences and Technology Center, SRI
Gatanas, Mark, Electronic Data Systems
Geiwitz, Dr. James, C2RM Chief Editor, Advanced Scientific Concepts Inc.
Ghosh, Dr. Sumit, Brown University
Hahn, Dick, Melpar Div., E-Systems
Hines, Jerry, Electronics Division, General Dynamics
Huynh, Dr. Tom, Systems Analysis and Simulation, R&D Div. Lockheed Palo Alto
Isaac, Jeane-Christophe, George Mason University
Johnson, Craig, Loral Command & Control Systems
Kelly, Kevin, MITRE
Kornell, James, Syukhtun Corporation
Kovacich, Dr. M., Systems Analysis and Simulation, R&D Div. Lockheed Palo Alto
Lavin, Dr. Milt, JPL
Layton, Horace G., CSC
Liebman, Elliot, Booz Allen & Hamilton, Inc.
Lightfoot, R. N. C., Institute for Defense Analyses
Lockwood, M. J., C³ Programs, Electronics Division, General Dynamics
Low, Larry, Information Sciences and Technology Center, SRI
Marshall, Dr. Donald S., The Military Conflict Institute, Chairman
McDonough, Martin F., Dynamic Research Corp.
McKay, Michael V., Rutgers CAIP
Might, Dr. Robert, George Mason University
Miller, Dr. Larry, ISI
Nerode, Dr. Anil, Cornell University
Niedenfuler, Dr. Frances, MITRE Washington C³I
Niznik, Dr. Carol, NW Systems
Novick, Yuki, ISR Corp.
O'Leary, Brian, Electronics Division, General Dynamics
Pei, Richard, Rand Corp, Wash., DC
Pfeiffer, Theodore, Consultant
Phoha, Dr. Shashi, Penn State University
Press, Barry, TRW
Press, Barry, TRW Systems Engineering and Development Division
Pyles, Dan, Melpar Div., E-Systems
Roland, Dr. Roland J., Roland & Associates Corp.
Rubin, Dr. Izhak, IRI/UCLA
Samet, Dr. Michael, ISR Corp.
Santoro, R.T., Oak Ridge National Laboratory/Martin Marietta Energy Systems Inc.
Schreier, Lou, Information Sciences and Technology Center, SRI
Schultz, Joann, Melpar Div., E-Systems
Schwalm, Dr. Norm, ISR Corp.
Shacham, Dr. Nachum, Information Sciences and Technology Center, SRI
Shaw, Dr. John J., Alphatech
Sheu, Dr. Philip C-Y, Rutgers University
Silver, Joel, ISR Corp.

Singer, Nick, Melpar Div., E-Systems
Sircar, Saurabh, Penn State University
Spiridon, Dr. Alex, SRI International
Stringfellow, George, Electronics Division, General Dynamics
Tenney, Dr. Robert, Alphatech
Terrel, John, GTE Advanced Machine Intelligence Technology
Walker, Dr. Bill, Electronics Division, General Dynamics
Walter, Mike, Electronics Division, General Dynamics
Wannamaker, Tom, Loral Command & Control Systems
Weber, Robert H., The Aerospace Corporation
White, Norman, Melpar Div., E-Systems
Williamson, Dr. Joseph, Melpar Div., E-Systems
Woodcock, Dr. A.E.R., Synetics Corp.
Zhou, Dr. Mengchu, NJIT

POC FOR COORDINATION AND COMMENTS

Commander
U.S. Army CECOM
AMSEL-RD-C2-TC (Dr. I. Mayk)
Fort Monmouth, N.J. 07703-5603
Tele: (908) 544-4996/4445, (DSN) 995-4996/4445
Fax: (908) 544-3687/3440, (DSN) 995-3687/3440/
DDN: AMSEL-RD-C2-CU@MONMOUTH-EMH3.ARMY.MIL

APPENDIX D: RELATED BACKGROUND

COMPARING WITH THE ISO OSI RM

From an ISO OSI RM perspective, a C^2 system is a network of systems, and the environment provides the media for communications. Thus any unit object which can communicate is potentially an ISO OSI RM system. From the C^2RM perspective, a C^2 system is a network of resources, and the environment provides the media for interactions. Thus any unit object which can interact is potentially a C^2RM resource. An ISO OSI RM system is a set of one or more computers, the associated software, peripherals, terminals, humans, and application processes which form an autonomous whole capable of communicating with other ISO OSI RM systems in the prescribed seven-layered fashion. The ISO OSI RM, however, is primarily focused upon the transmitter-receiver meta-object problem. It does not delve into the architecture of the application processes. The ISO OSI RM (Communications) application layer mediates between the lower six communications layers and any supported application process such as the C^2 process. Thus, from an ISO OSI RM perspective, the observation, decision, and action processes of the global C^2 process are supported by the layered communications process as shown in Figure D.1.

When the ISO OSI RM is viewed as a system, there are two groups of layers which sandwich the middle (fourth) layer. The lower three layers are largely independent of the specific user application and directly involve the resources used in carrying out the transmission of messages across direct links as well as indirectly through a system of networks. The upper three layers reflect the characteristics of the specific user communications requirements independent of the specific communication network system used. The middle layer, between these two groups, mediates between them, i.e., making the network transparent to the upper three layers.

A similar grouping organization is noted for the structure of the layers of C^2. For each interacting pair of resources in deployment, the lower three layers typically describe the processes independent of the commander and his staff. The friendly resources are linked and networked for the purpose of execution: interoperation with each other and interaction with the other combatants and the environment. But these networks must support various operational requirements which are generated by well thought out plans for the mission in effect. This is generally how the military user operates, i.e., a) a mission is articulated, b) plans are drawn and c) operations orders are issued for execution. These three complex processes define the three upper layers of the C^2RM. These three layers pertain predominantly to the commander and his staff. The middle layer is concerned with C^2 procedures which are formalized in many cases by Standing Operating Procedures (SOPs). C^2 procedures bridge between staff users and line users. The staff user performs scenario dependent actions as described by the upper three layers. The line user executes and provides services which are relatively scenario independent in accordance with the functions described by the lower three layers.

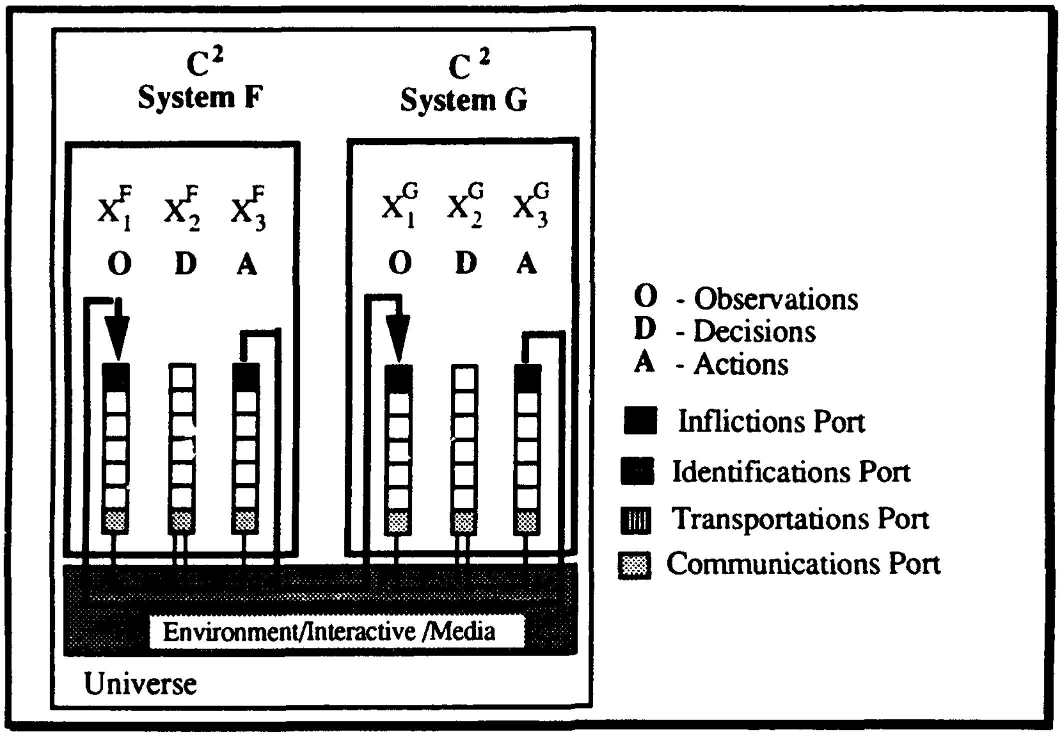

Figure D.1. The C^2 process through the ISO OSI RM

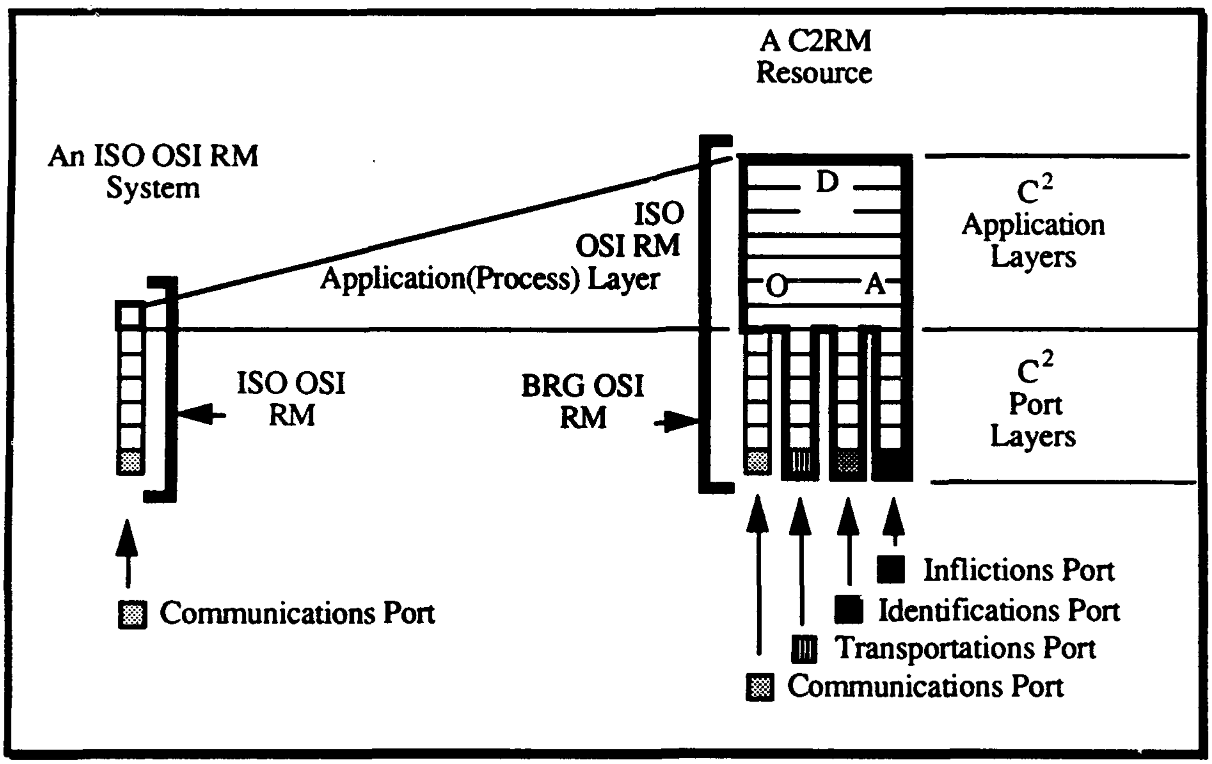

Figure D.2. Extending the ISO OSI RM to C^2

The C^2RM, however, is a more complete reference model from a C^2 perspective since it provides layers for the global C^2 process and for transportation, identification and infliction application processes which at their highest level of abstraction complement the communications process in support of the C^2 process. As shown in Figure D.2, an ISO OSI RM system is expanded at the ISO OSI RM application/application process layer to provide an isomorphically layered architecture for the C^2RM Application/Ports layer.

Similarly to the ISO OSI RM, this modular framework may not preclude integrated designs and implementations where appropriate. Thus, different algorithms or technologies may be used to implement any desired set of services within and across layers, provided that services of a given layer are relative to the requirements of the layer above. Much like the ISO OSI RM Security Architecture [[2]], security considerations must be applied at every layer of every resource and for every class of interaction. Ideally, any breach of security at a particular layer of a resource should be isolated by the (N-1)layer or the (N+1)layer and prevented from further propagation. Otherwise, the (N)layer of the other resources must be able to isolate the defective resource to prevent it from contributing to any further potential compromise.

Comparing with the ODP/MCI RM

The Basic Reference Model of Open Distributed Processing (ODP-RM) [[5]] and associated Methods for C3I Interoperability-Reference Model (MCI-RM) [19] provide important insight of key aspects which are incorporated in the C^2RM. The ODP-RM includes five Viewpoints which are all relevant to C^2: Enterprise, Information, Computational, Engineering and Technology. The first two clearly span the Problem/Solution Domain and the latter two clearly span the Implementation/Technology Domain. The third Viewpoint spans both Domains. The MCI RM, however, is an enterprise model and as such it does not model interactions with the environment and with adversarial forces.

The MCI RM provides a general model for open systems to cooperate and interoperate by sharing their services and stressing the quality of their services. An MCI Service is an encapsulation of both function and data. Thus, the MCI RM suggests that the application layer/application process of an ISO OSI system interoperates through layered services. The C^2RM, therefore, provides the layered structures for such services. Knowing the Grade of Service is a key feature of the model.

APPLICABILITY TO DIS

The Distributed Interactive Simulation (DIS) [20] is a system of dissimilar, heterogeneous, interoperable simulators residing at multiple locations which span a large geographic area and are networked to provide a synthetic, virtual representation of warfare environment. Component simulators exchange data via standard Protocol Data Units (PDU). As such, DIS is an extension of the Simulation Networking (SIMNET) Program developed by the Defense Advanced Research Projects Agency (DARPA).

The basic DIS concepts are:

a. No central computer for event scheduling or conflict resolution.
b. Autonomous simulation nodes responsible for maintaining the state of one or more simulation entities.
c. There is a standard protocol for communicating "ground truth" data.
d. Receiving nodes are responsible for determining what is perceived.
e. Simulation nodes communicate only through changes in their state.
f. Dead reckoning is used to reduce communications processing.

The key DIS design features incorporate:

a. Object-Oriented Entities with private and public components
b. Entity Sphere of Interaction
c. Gaming Area
d. Analytic as well as Monte-Carlo Models
e. Aggregation and Level of Resolution
f. System Management
g. Communications Services in accordance with ISO OSI Reference Model

The C^2RM is compatible with DIS in the sense that it provides a standard open system structure to simulated entities, and their sphere of interaction.. Aggregation is most natural at the layered boundaries. The Level of Resolution may conveniently correspond to the Asset layer of Resources. Dissimilar simulators may be more productively related to each other on the basis of which services they provide. As future simulators evolve into more open systems in concert with layered decision-aid applications, PDUs may be used to support decision-aids as well as simulations.

APPENDIX E: A GENERIC STRUCTURE EXAMPLE

As shown in Figure E.1, individual resources, depicted by the seven layers of C^2 applications, may roam about the environment independently carrying out autonomous missions, or they may be grouped to complement and supplement each other's effort in an aggregated manner as shown in Figure E.2.

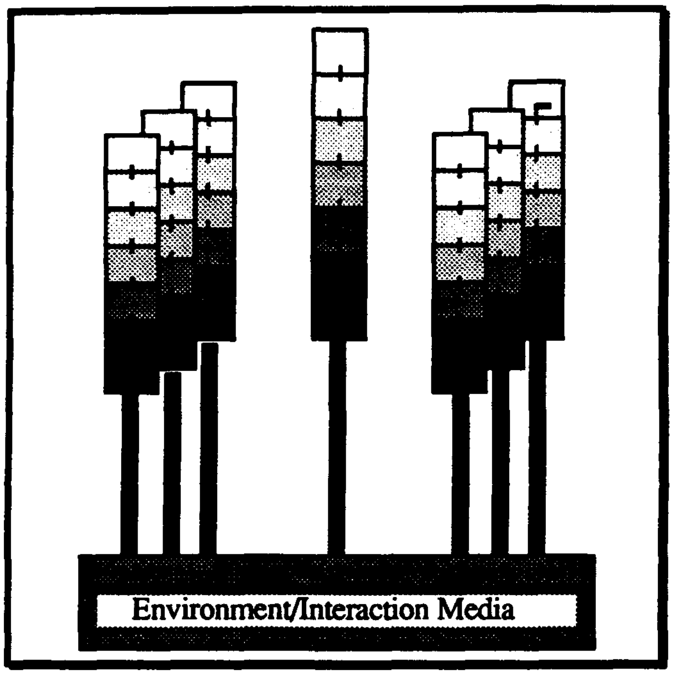

Figure E.1. Loose coupling of independently layered, autonomous resources

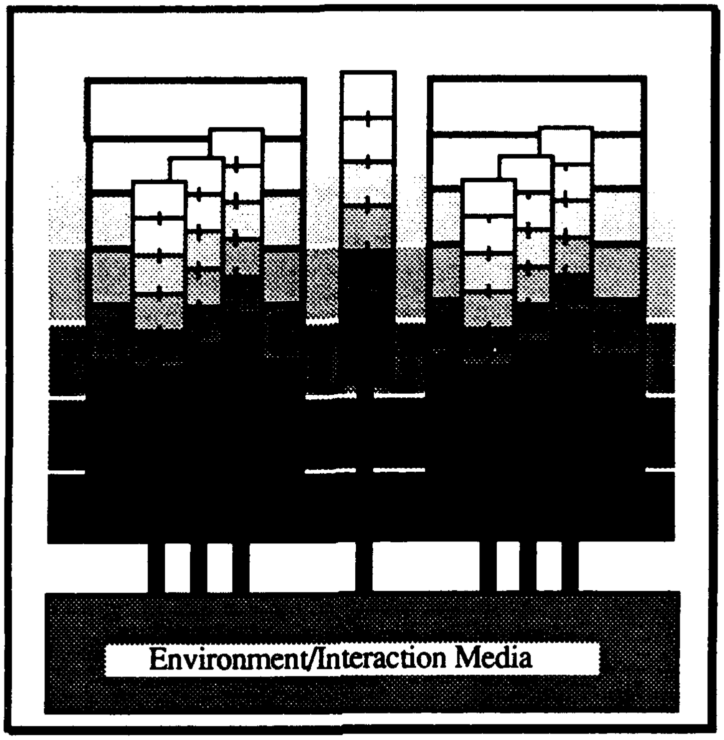

Figure E.2. Tight coupling of hierarchically layered, aggregated and distributed resources

In Figure E.3, the potential interactions among resources are explicitly distributed to depict a more complex three tier C^2 system structure. At the highest tier, denoted by **a**, one command resource is responsible for establishing the mission, generating, updating and selecting plans and monitoring the resulting operations as generated by the subordinate resources, **aa** and **ab.** At the middle tier, control resources, **aa** and **ab**, are responsible for establishing the operations, generating, updating and selecting procedures and monitoring the resulting network interactions as activated by their respective line resources, **aaa** and **aba**. The resources, **aaa** and **aba**, at the lowest tier, are responsible for establishing a cohesive network of multilateral interactions which use and expend the assets at their disposal in support of the system level mission as communicated through channels from the command resource, **a**. The C^2 system shown in Figure E.3 may represent a wide range of C^2 paradigms. In particular, a comparison with Figure 7 may be used to map several possibilities. For example, resources grouped according to {(**aaa**) (**aba**) and (**a, aa, ab**)} or {(**aaa, aa**) (**aba**, **ab**) and (**a**)} may constitute the observation, action and decision subsystems, respectively. Following the formal description provided thus far and as typified in Figure E.2, a more specific example is illustrated next.

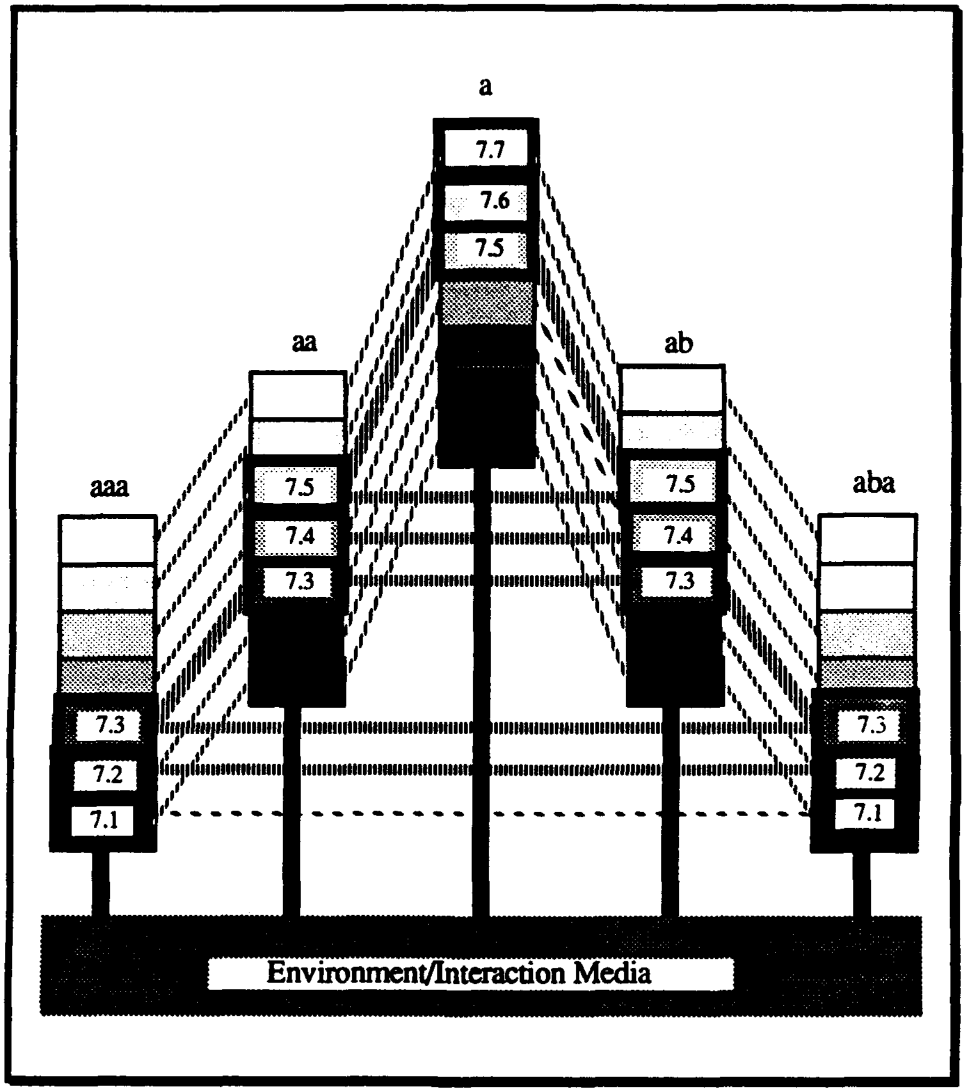

Figure E.3. An abstract C^2 system structure

APPENDIX F: A GENERIC SCENARIO EXAMPLE

Consider the topological positioning of resources and their mutual physical interactions for a hypothetical but illustrative example as shown in Figure F.1. Resource C is the commander of system F. Resource S is a sensor of system F which includes its own controller. Resource W is a weapon of system F which also includes its own controller. Resource G belongs to system G. Thus, system F is a simplified version of the C^2 system shown in Figure E.3 where resource **a** corresponds to resource C, resource **aa** and **aaa** have been integrated or aggregated to represent resource S and resource **ab** and **aba** have been integrated or aggregated to represent resource W. Systems F and G are involved in a mutual conflict. More specifically, the conflict has deteriorated down to the level of engagement (Conflict layer 7.7.2). Note that resources C, S, and G are fixed sites whereas resource W is mobile as indicated by the textured arrow for transportation. As shown by the other textured arrows, resource S is within identification range of resource G as a potential target. Resources C and S, as well as resources C and W are within communications range. Finally, resource W is within range to inflict damage upon resource G. Having identified the types of interactions which are key to a given resource in a given scenario, each resource may be expanded in terms of the layers which correspond to each interaction class and the application sublayers which couple among the interactions across resources. In the following scenario the proposed layers involved for each proposed service are identified in parentheses.

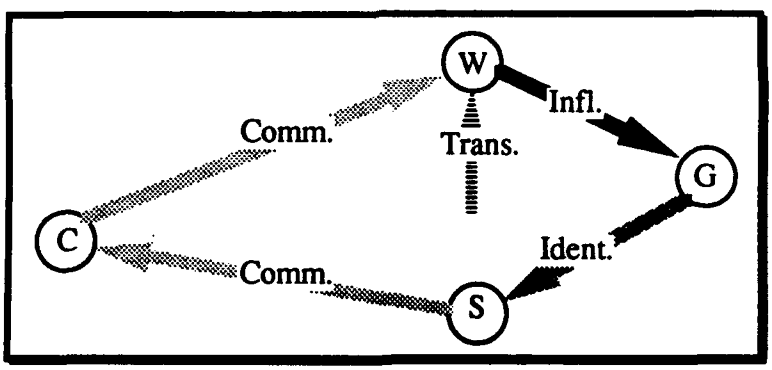

Figure F.1. Illustration of topological positioning and physical interactions

Consider a single thread analysis, as shown in Figure F.2, for the context of one hypothetical sequence of events chosen to illustrate what could occur from the time of detection by resource S to the time of firing a round of ammunition by resource W upon target resource G. Using binoculars as its physical asset for the identification port, resource S detects (Identification layer 1/S) an activity on top of an adjacent hill. Resource S tracks (Identification layer 2/S) the activity for a short time period (Identification layer 3/S-5/S) and notices a large antenna. Resource S performs additional correlation (Identification layers 3/S-6/S) to determine that resource G at location-time (x,y,z,t) is an observation post. At layer 7.1/S, the target information is stored and displayed as a **package** data object. At layer 7.2/S, the target information is **transacted** either for immediate transfer to the commander, resource C, using its available communications port (Communications layers 6/S-1/S), or for local assessment of its relevancy with respect to any one of the several available **assignments** generated at layer 7.3/S. As an unexpected target of opportunity, an assignment cannot be made by resource S and layer 7.4/S must decide if any procedures have been authorized for handling the target as a part of any one of several ongoing **jobs**. Assuming that **tasks** activated at layer 7.5/S did not anticipate this class of target at the given location, the **plan** generated at layer 7.6/S must be reviewed to see if, indeed, the recognized target is of interest to the **mission** derived at layer 7.7/S for any resource of System F. Assuming that the mission is sufficiently broad, resource S decides to report target resource G as a target of potential interest to resource C. Resource S reinterprets, revises or regenerates a plan (layer 7.6/S), task (layer 7.5/S), job (layer 7.4/S), assignment (layer 7.3/S) and transaction (layer 7.2/S) in that order to enable a communications package (layer 7.1/S) to be motivated and formatted down through the communications port (Communications layers 6/S-1/S). The resulting message is transmitted through the environment to resource C. Resource C demodulates and decodes the message (Communications layers 1/C-6/C) for presentation to layer 7.1/C.

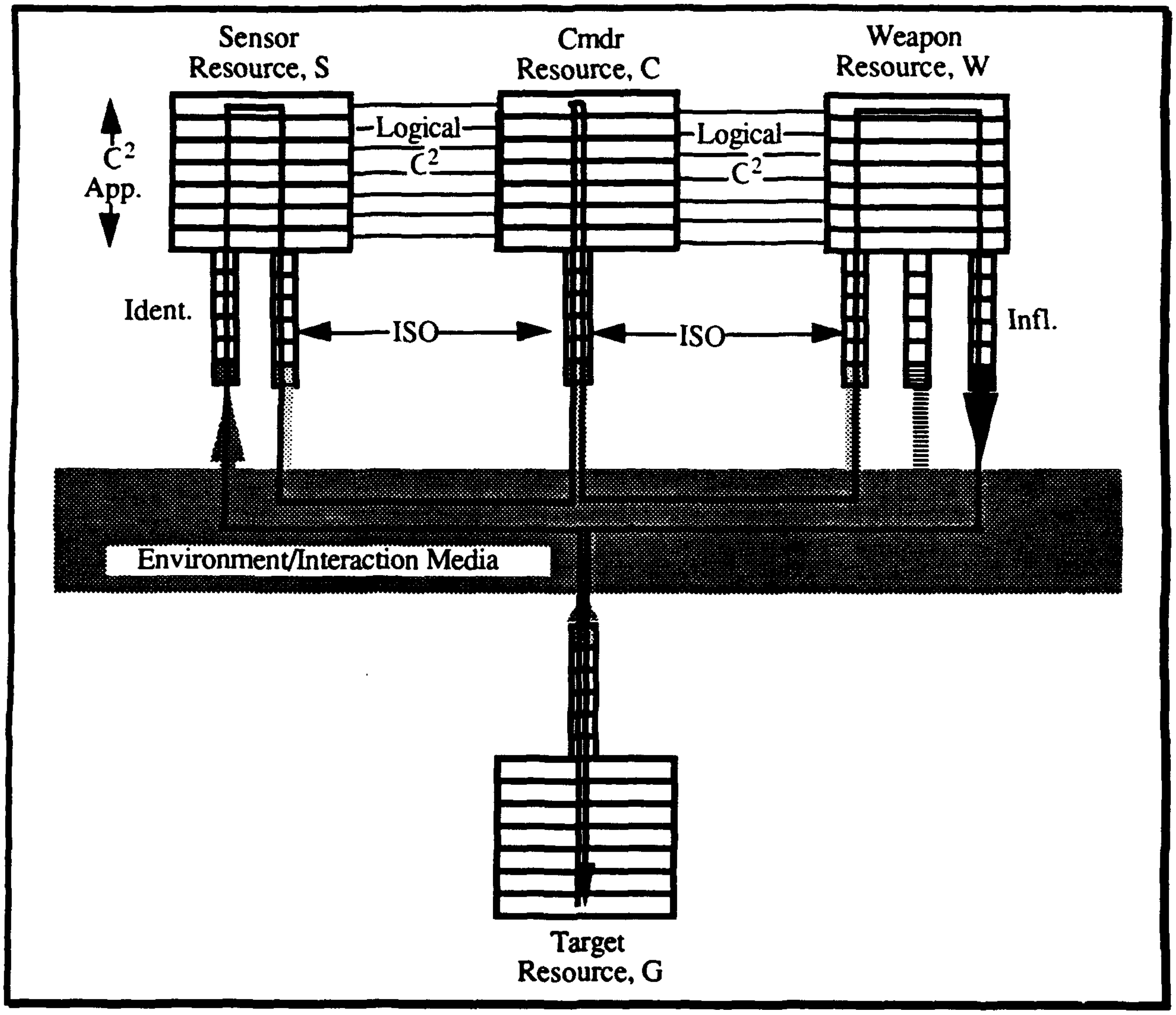

Figure F.2. Illustration of a single thread analysis of interactions

Stored and displayed by layer 7.1/C, the package is made available to layer 7.2 /C for re-transacting. Re-transacting, even if trivial, is essential for cross-coupling packages between any two resources. At this point, resource C may ask for more information. In this example, however, time is of the essence and the commander must decide immediately whether the target is a threat to his immediate mission or some future mission. Being caught by surprise, the package which is automatically re-transacted for potential infliction is preempted by a review process for consistency and practicality with respect to currently active assignments (layer 7.3/C), jobs (layer 7.4/C), tasks (layer 7.5/C), plans (layer 7.6/C) and missions (layer 7.7/C) as understood by resource C. Resource C establishes that the target is a threat to current and future plans and operations. Resource C decides to select (layer 7.7/C) Resource W to engage the target G. Resource C generates an engagement mission (layer 7.7.2/C-7.7.1/C), redraws or revises the plan (layer 7.6/C), issues a new task (layer 7.5/C) to initiate the job to fire on the given target (layer 7.4/C), and remakes an assignment (layer 7.3/C) to repackage (layer 7.2/C) the target information in a manner suitable for a transaction (layer 7.1/C) to resource W. The repackaged information about target resource G is processed by layer 7.1/C to create a communications transaction which follows the ISO OSI RM services down through the communications port of resource C (Communications layers 6/C-1/C), through the environment and up through the communications port of resource W (Communications layers 1/W-6/W).

Resource W (layer 7.1/W) stores and displays the package and makes it available to layer 7.2/W for potential re-transaction. A transaction (layer 7.2/W) allows for cross-coupling of communication, transportation and infliction packages. At this point, resource W must decide (layer 7.3/W) whether it is close enough to cause the required damage or whether it should move closer to improve the probability of impact. Once an assignment is contemplated for infliction (layer 7.4/W) it is presented higher up to be scheduled as part of an ongoing job. The candidate job is then subject to review for priority, consistency, and urgency with respect to existing tasks (layer 7.5/W), plans (layer 7.6/W) and missions (layer 7.7/W). Resource W decides to respond with a hasty fire mission (layer 7.7/W). The overall deployment plan is checked (layer 7.6/W) to assure that Resource S is a safe distance away (layer 7.5/W). The job is approved (layer 7.4/W), and an assignment is made for infliction (layer 7.3/W). The target information available from storage (layer 7.1/W) is repackaged with supplemental timing constraints (layer 7.2/W). It is then reprocessed to create an inflictions package (layer 7.1/W) which is serviced by the inflictions port. Infliction port services may range from a survey of the weapon position to the loading armaments and pulling the trigger (Inflictions layers 6/W-1/W). The armament is launched through the environment and may penetrate up through any port of target resource G which may become damaged. In this example, for hypothetical completeness, Resource W is able to destroy the antenna of target resource G.

This example may be easily modified and expanded to illustrate many other examples of interest to specific R&D, operations and tests, education and training organizations. The user is challenged to come up with his own example to be considered for future reference and as an aid for understanding command and control and this C²RM. A good start is to use any one or more of the following references F.1- F.7:

APPENDIX F REFERENCES

F.1. Daniel P. Bolger, Dragons at War, Land Battle in the Desert, Ivy Books/Ballantine Books, NY, © 1986.

F.2. John F. Antal, Armor Attacks, Presidio Press, CA, ©1991.

F.3. R. Leonard, The Art of Maneuver; Presidio Press, CA, ©1991.

F.4. Robert L. Shaw, Fighter Combat, Tactics and Maneuvering, Naval Institute Press, Annapolis, Maryland ©1985.

F.5. Wayne P. Hughes, Jr., Fleet Tactics, Theory and Practice, Naval Institute Press, Annapolis, Maryland ©1985.

F.6. J. Richard Hill, Anti-Submarine Warfare, Second edition, Naval Institute Press, Annapolis, Maryland ©1984.

F.7. James F. Dunnigan, How to Make War, A Comprehensive Guide to Modern Warfare, All the World's Weapons, Armed Forces & Tactics, Quill William Morrow, New York,©1988.

DO YOU HAVE ANY COMMENTS OR SUGGESTIONS?

Name, Title, Position ______________________________
Organization ______________________________
Street Address/Mail Stop/Office Symbol ______________________________
City, State, Zip Code ______________________________
Telephone (Include area code) ______________ Facsimile ______________
Email ______________________________

For comments on the document you may add pages as needed.
Please send your comments to the following address:

Commander
U.S. Army CECOM, C^2 Systems Integration Directorate
AMSEL-RD-C2-TC (Dr. I. Mayk)
Fort Monmouth, N.J. 07703-5603
DDN E-mail : AMSEL-RD-C2-CU@MONMOUTH-EMH3.ARMY.MIL
(If sending E-mail, please start the Subject field with the word "C2RM")

All types of comments are encouraged. Comments may be constructive or destructive, positive or negative, general or specific, outlined or detailed, technical or editorial, terse or verbose, direct or indirect, with or without rationale, with or without recommendations. A comment may be a mere endorsement, objection, disputation or a request for clarification. All comments will be considered and maintained for the record in support of updating the C^2RM. All comments will be acknowledged if you so desire (Check below). Each comment will be reviewed for a potential contribution which hopefully will lead to a more robust open system architecture framework for the C^2 community-at-large.

Please check appropriate choice:

______ You may include my name in the C^2RM Acknowledgement Appendix C.
______ I would like to receive future updates.
______ I would like to be invited to future meetings of the BRG C^2RM Subgroup.
______ I would like to be a member of the C^2RM user group.
(My application is to ______________________________)

______________________ ______________
Signature Date