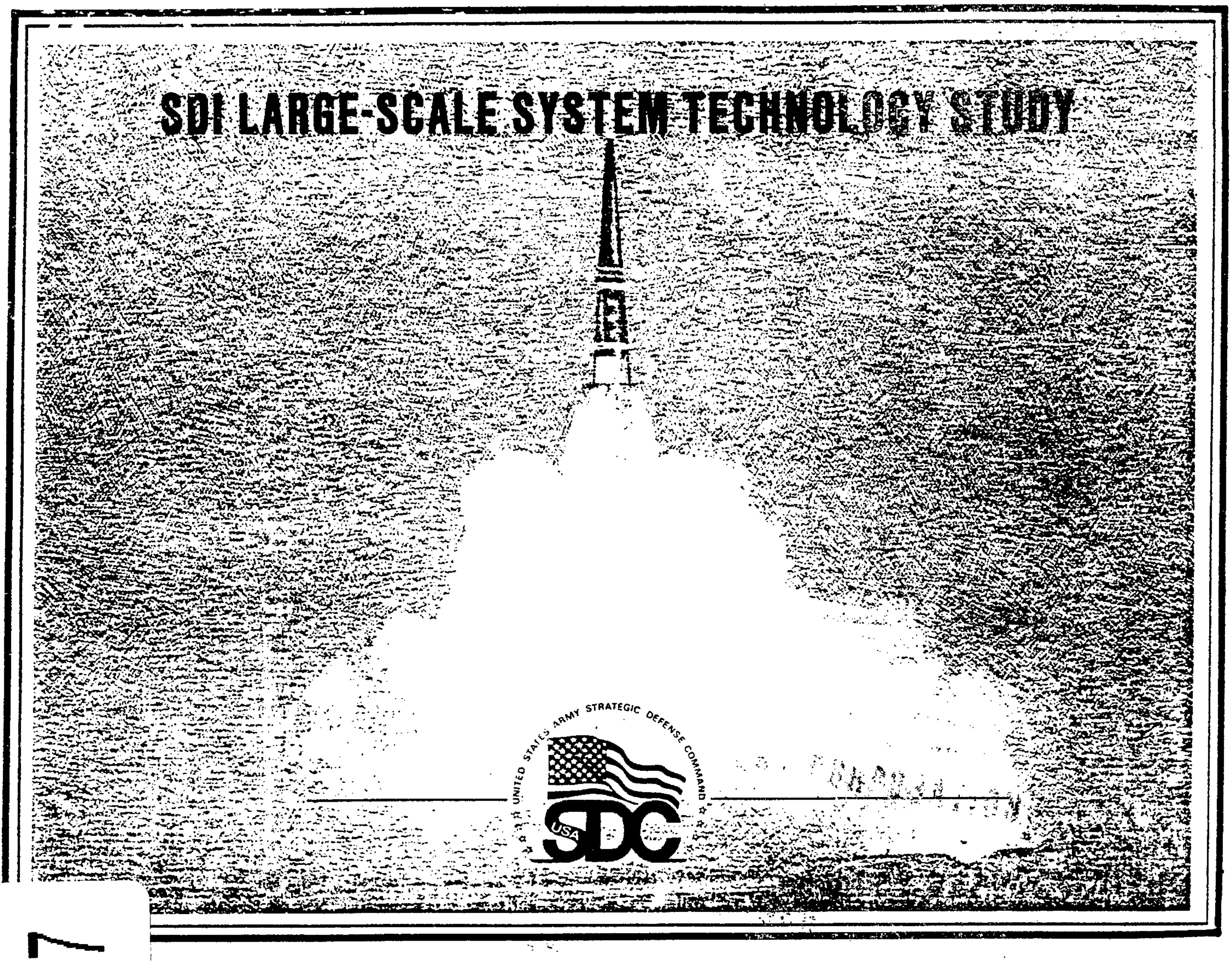
SDI LARGE-SCALE SYSTEM TECHNOLOGY STUDY



PREPARED FOR
U.S. ARMY STRATEGIC DEFENSE COMMAND

BY
SYSTEM DEVELOPMENT CORPORATION

18 APRIL 1986

Accession Number: 1708
Title: SDI Large-Scale System Technology
Study
Corporate Author or Publisher: System Development Corporation
Report Prepared For: U.S. Army Strategic Defense Command,
Huntsville, AL
Publication Date: Apr 18, 1986
Pages: 135
Descriptors, Keywords: Boost Phase Architecture Language
Sensor Midcourse Producibility
Algorithm Weapon Research Target BM
C3 Technology Model Simulation AI
Network

SDI LARGE-SCALE SYSTEM TECHNOLOGY STUDY

FINAL REPORT

18 April 1986

PREPARED FOR
U.S. ARMY STRATEGIC DEFENSE COMMAND

The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

FOREWARD

This report contains the results of a four month study conducted by the System Development Corporation on behalf of the Systems Analysis/Battle Management Directorate of the U.S. Army Strategic Defense Command (USASDC). The study addresses the global technical issues of the Strategic Defense Initiative (SDI) and attempts to derive viable approaches for their resolution. Because the recommendations contained herein represent, in the opinion of the authors, priority items for future study, it is expected that this report serve as a reference for planning future research and experiments.

The SDI Large Scale System Technology Study was conducted by a team of nationally recognized experts in various technical fields related closely to the SDI. The study team was thoroughly briefed on the SDI in general and specifically on the current planning for Battle Management/Command, Control, and Communication. Ample opportunity was provided for team members to interact with each other so as to encourage a healthy interchange of ideas, comments, and criticisms.

Appreciation is expressed to the SDI Large Scale System Technology team for their participation and support during the preparation of this report. Specifically, Dr. C.T. Leondes of the University of California - Los Angeles; Dr. Robert Bass of Inventek Enterprises; Dr. Carroll Johnson of the University of Alabama in Huntsville; Dr. Karen Gordon of the Mitre Corporation; Dr. Thomas Garvey of SRI International; Dr. Fredric Weigl of Rockwell International; Dr. Nils R. Sandell of AlphaTech; Dr. William J. Kenny of Control Data Corporation; and Mr. Malcolm Johnston of Charles Stark Draper Laboratory are thanked. Appreciation is also expressed to Dr. Daniel Siewiorek of Carnegie-Mellon University and to Dr. Herbert Hecht of Draper Labs for their special contributions.

The attention of a select review committee who attended the final briefing of the study team and who reviewed the draft Final Report is also greatly appreciated. The attention of the committee, headed by Dr. Carl Davis (USASDC) and Capt. William Hart (SDIO), did much to give focus to some of the more outstanding issues.

A core group was comprised of expertise within the Huntsville area to plan the project, assist in the organization and conduct of Panel sessions, and to review and critique the final report. Dr. Doyle Thomas was the USASDC sponsor of the project. He supported the project team with technical information, guidance and comment. Mr. Bill McDonald acted as chief moderator for the panel and was a valuable source of information about the technicalities of ballistic missile defense. Dr. Carroll Johnson assisted in project planning and review. Mr. Paul Thompson provided valuable portions of the introductory sections of the report. Mr. Mack Cooper was the project manager for the study and editor of the report. Special thanks goes to Ms. Brenda Tyson and her staff of graphic artists and word processing specialists. Without their long hours and timely support, the work of the Panel would have been much more difficult.

TABLE OF CONTENTS

SECTION PAGE

TABLE OF CONTENTS (Continued)

SECTION PAGE

TABLE OF CONTENTS (Continued)

SECTION PAGE

TABLE OF CONTENTS (Continued)

SECTION PAGE

TABLE OF CONTENTS (Continued)

SECTION PAGE

SECTION 1 - INTRODUCTION AND PURPOSE

Because the Strategic Defense Initiative presents one of the most challenging problems ever posed to science, the performance requirements for many components and subsystems exceed the current state-of-the art. Survivability, reliability, maintainability, security, and cost-effectiveness are critical issues. In addition to the problems of subsystem engineering, there is the need to coordinate the operation of large numbers of system components that are dynamically entering and leaving the battle space. This coordination is addressed by the Battle Management function. The algorithms and technologies required to support Battle Management are the subject of the SDC Large Scale Systems Technology Study.

Battle Management is defined on two levels:

1. As a collection of algorithms for gathering information of the state of the battle and for the allocation of system resources.
2. As a collection of technologies by which the algorithms are implemented and supported. These include the computer hardware/software, data base management, networks, and communications.

In general, the algorithms pose requirements for the technologies while the latter impose constraints on the former. Battle management approaches range from autonomous, with a minimum of coordination, to centralized with the best opportunities for optimal utilization of resources.

Work has already been directed toward the development of various aspects of ballistic missile defense, most of it focused on the problems of defending specific sites during the terminal phase of an attack. There are, still to be solved, many technical problems that will require significant skill and insight into a variety of technology areas. It is the purpose of the SDI Large Scale System Technology Study to bring to bear the specialized skills of a small panel of expert technologists to address some of the more difficult data handling and control issues of the SDI, particularly in the Battle Management Command, Control, and Communication (BM/C^3) function.

Initial planning to bring together such a panel began with a top-level functional analysis of the SDI as a whole. A functional decomposition of SDI revealed that three major functions exists: Information Acquisition; Battle Management/Command, Control, and Communication; and SDI Controllables.

From the functions thus identified, a panel structure was developed to provide not only the appropriate technical expertise but to be capable of addressing the spectrum of design from the philosophical to the practical. Qualified experts in each of these fields were then sought. The Panel was organized, as illustrated in Figure 1-1, to include the following:

1. Two overall system design approaches: Classical Systems Theory and Artificial Intelligence.
2. Battle management algorithms: Estimation, Decision and Control, the latter two being related hierarchically.
3. Implementation technologies: Communications, Networking, Data Management, and Computer Hardware/Software.
4. A system issue that spans all of the above, i.e., Fault Tolerance.

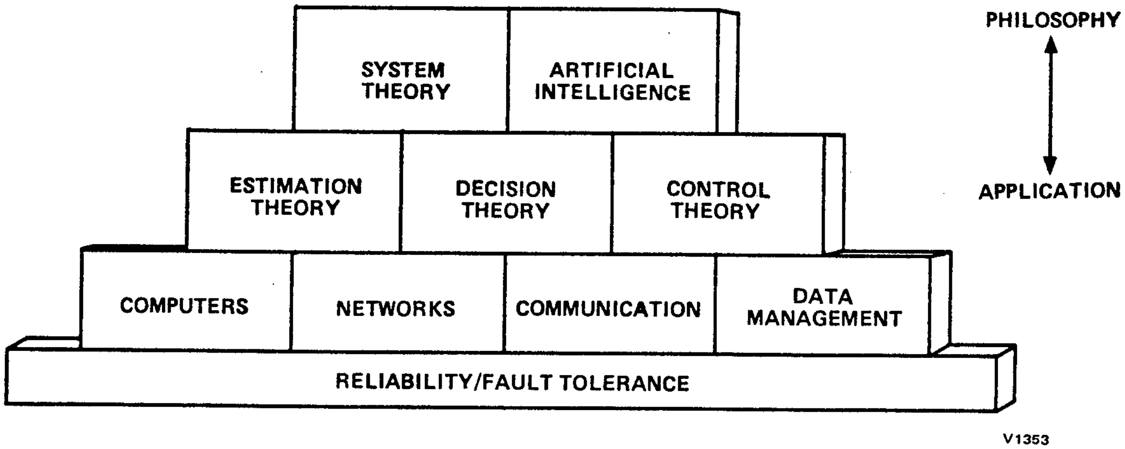


Figure 1-1. SDI Technology Panel Structure

Nationally recognized experts in each of these fields were sought and the panel was comprised as shown in Figure 1-2.

TECHNOLOGY REPRESENTED	TECHNOLOGY EXPERT	EMPLOYMENT AFFILIATION
SYSTEM THEORY	DR. CARROLL JOHNSON	UNIVERSITY OF ALABAMA IN HUNTSVILLE
ARTIFICIAL INTELLIGENCE	DR. THOMAS D. GARVEY	SRI INTERNATIONAL
CONTROL THEORY	DR. ROBERT BASS	INVENTEK ENTERPRISES
ESTIMATION/DECISION THEORY	DR. C. T. LEONDES	UNIVERSITY OF CALIFORNIA
DATA MANAGEMENT	DR. KAREN D. GORDON	MITRE CORPORATION
NETWORKS	DR. NILS R. SANDELL	ALPHATECH
COMMUNICATIONS	DR. FREDERIC WEIGL	ROCKWELL INTERNATIONAL
COMPUTER SYSTEMS	DR. WILLIAM J. KENNY	CONTROL DATA CORPORATION
RELIABILITY/FAULT TOLERANCE	MR. MALCOLM JOHNSTON	CHARLES STARK DRAPER LAB.

V1349

NOTE: DR. DANIEL P. SIEWIOREK OF CARNEGIE-MELLON UNIVERSITY WAS ALSO RETAINED AS A RESOURCE CONSULTANT TO THE PANEL.

Figure 1-2. SDI Large Scale System Technology Panel

Panel members were instructed to examine each of the subject technology areas for issues critical to the Strategic Defense Initiative. For each issue identified, candidate approaches and a recommended research plan were to be presented. It is important to note that these issues were considered independent of any specific system construct or threat scenario. As a result, the recommendations are necessarily general. More detailed study will require definition of baseline system configurations and loadings.

SECTION 2 - EXECUTIVE SUMMARY

A panel of qualified experts was assembled to study the more difficult technology issues of the Strategic Defense Initiative. The SDI Large Scale Systems Technology Panel was comprised of nationally recognized experts in the fields of system theory, data management, network theory, communications, computer systems, and fault tolerance. The Panel was asked to identify the more salient technical issues of the SDI in each of the technology areas, to identify alternatives to the resolution of the issues, and to make recommendations for research and/or experimentation. Some of the more outstanding recommendations of the Panel follow.

System Theory--Efforts should be directed to model the complex, multihybrid systems of SDI and to develop hierarchical measures of performance and effectiveness.

Artificial Intelligence--Techniques for reasoning and planning over time in the presence of uncertainty need to be developed. Artificial Intelligence may be applicable to numerous SDI issues including situation assessment, strategy/tactics planning, and diagnostics and repair. Research is needed to explore such applications and to obtain measures of effectiveness.

Control Theory--Scientific, distributed/decentralized control theory should be applied to the problems of system decoupling to reduce its dimensionality, improve reliability, and increase security through adaptive architecture and reconfigurable network topology.

Estimation Theory--Research into the application of estimation theory to a broad spectrum of SDI issues is recommended. Multisensor correlation by a distributed system using target attributes as well as position is needed. Techniques for precision pointing and tracking in boost, post boost, and midcourse may be addressed as well as the use of estimation theory to accomplish target threat discrimination from large, multicolor, focal plane arrays. Other recommendations focus on the need to provide executive command authority with reliable information to initiate, interrupt, or terminate the operation of the SDI system.

Data Management--In light of the highly dispersed nature of the SDI data systems, alternatives to absolute data consistency need study, perhaps by moving from local to more coordinated data bases. Methods for improving data base management performance in the context of BM/C^3 need investigation.

Network Theory--Network management algorithms and protocols for end-to-end error control, routing, flow control, and topology update need development. These algorithms and protocols should be distributed, adaptive, and as nearly optimal as possible.

Communications--Security requirements and policies based on defined threat scenarios need to be established to serve as a basis for designing network security systems. Waveforms and coding techniques to modulate expected signal types should be determined. Design hardening to mitigate nuclear effects must be investigated and understood before SDI BM/C^3 systems can be considered viable.

Computer Systems--High performance architectures for space applications through the study of parallel/distributed architectures, special function processors, data flow, and control flow must be developed.

Fault Tolerance--Present computer system fault detection, identification, and recovery techniques need simplification. Architectures combining parallelism and fault tolerance should be developed as well as distributed operating systems to operate with stringent real-time constraints. Metrics for software reliability measurement need formulating, and techniques for reliability prediction will require more work. Apply new approaches to critical BM/C^3 software under stressing fault scenarios and workloads and demonstrate with fault tolerant hardware.

One recommendation that was universally put forth by the Panel expressed the need for SDI Modeling and Simulation Facilities to support such diverse research as mentioned above. It was recommended that a preliminary study of the requirements for such a facility be initiated, and that existing, large-scale facilities be reviewed to determine their possible relevance to SDI needs.

It was intended that the performance of the SDI Large Scale Systems Technology Study would provide valuable insight into needed future research and/or experimentation. Probing into the key technology areas to identify the key technical issues was expected to reveal potential alternative solutions to key issues that must be solved. The potential alternatives, in turn, would suggest needed work in the form or study, research, and experimentation. In concluding the deliberations of the panel, members were asked to contribute to a tabulation of summary comments. Specifically, the following questions were asked:

1. Of all the issues/alternatives/recommendations identified for each technology area, what are the most important items to SDI?
2. What is the primary program interface for this item with the rest of SDI?
3. Is the item dependent on other SDI functions/issues/items?
4. Is the resolution of this issue critical to the success of the program?
5. Are there other U.S. projects that can provide valuable aid in resolving the issue?
6. What level of research (applied/basic) is appropriate?

The results of this work are contained in Table 2-1. While the table does not address all of the issues or recommendations of the panel, it does highlight some of the more important activities that should be addressed.

Table 2-1. Summary of Recommendations (Page 1 of 3)

ITEM	SDI PROGRAM INTERFACE	ARCHITECTURAL DEPENDENCY	PACING	PROGRAM LEVERAGING AVAILABILITY	RESEARCH LEVEL
1. System Theory					
a) Modeling - Complex, Multi-hybrid System	BM/C^3 only	No	No	ONR/NBS	Basic
b) Hierarchical MOE/MOP	BM/C^3 only	No	No	ONR	Basic
2. Artificial Intelligence					
a) Reasoning Over Time with Uncertainty	All	No	No	DARPA/RADC	Basic
b) Situation Assessment	All	Yes	No	NOSC/DARPA AFWAL	Applied
c) Planning Over Time with Uncertainty	All	No	No	DARPA/NASA	Basic
d) Strategy/Tactics Planning	All	Yes	No	DARPA/	Applied
e) Diagnosis and Repair	All	No	No	Air Force	Applied
3. Control Theory					
a) Scientific Distributed/ Decentralized Control Theory	All	No	No		Basic
b) Distributed Dynamic Weapon Target Allocation (Including Planning Algorithm)	DEW/KEW Fire Control	Yes	Yes	DARPA/ Aegis	Applied

Table 2-1. Summary of Recommendations (Page 2 of 3)

ITEM	SDI PROGRAM INTERFACE	ARCHITECTURAL DEPENDENCY	PACING	PROGRAM LEVERAGING AVAILABILITY	RESEARCH LEVEL
4. Estimation/Decision Theory Multisensor Correlation - Distributed - Attribute As Well As Position	SATKA Also	Yes	Yes	RADC/etc. DARPA/	Applied
5. Data Management Data Consistency/Performance Tradeoff	All	Yes	Yes	ARO/RADC	Applied
6. Networks Network Management Algorithms and Protocols	BM/C^3	Yes	Yes	DARPA/SURAN MSS/DARPA NRL-SPARWARS	Applied
7. Communications a) Communication Coding and Waveform Development	All	No	No	ISDN/DARPA	Applied
b) Security	All	Yes	Yes	NSA/BLACKER AUTODIN II	Applied
8. Computer Systems Spaceborne Computer Architecture - Parallel/Distributed	All	Yes	Yes	DARPA Strategic Computing Initiative NASA	Applied

Table 2-1. Summary of Recommendations (Page 3 of 3)

ITEM	SDI PROGRAM INTERFACE	ARCHITECTURAL DEPENDENCY	PACING	PROGRAM LEVERAGING AVAILABILITY	RESEARCH LEVEL
9. Fault Tolerance					
a) Coverage, Connectivity, and Control	All	Yes	No	NASA (AIPS), DARPA, Air Force, Navy	Basic Applied
b) Software Fault Tolerance	All	Yes	No	NASA DARPA, Air Force	Basic Applied
c) Damage Tolerance	All	Yes	Yes	NASA DARPA, Air Force, Navy	Basic Applied
d) Evaluation and Verification	All	No	No	NASA (AIPS) DARPA, AIR FORCE	Applied

A more detailed summary of the results of the panel's work is presented in tabular form in Appendix A. Issues, alternatives, and recommendations for each technology area are presented. The full text of the Panel members' final reports is contained in Sections 5 through 13.

SECTION 3 - OVERVIEW OF SDI SYSTEMS AND COMPONENTS

The scope, complexity, and size of the Strategic Defense Initiative makes it difficult to address the program using a single outline. Various papers and studies have approached the subject in different ways. One of the more common approaches is to break the SDI down into the various "stages" of the defensive battle. Another approach is based on the categories of weaponry and other equipment that may be employed. A third approach addresses the major functions and speaks in terms of the systems that will perform them.

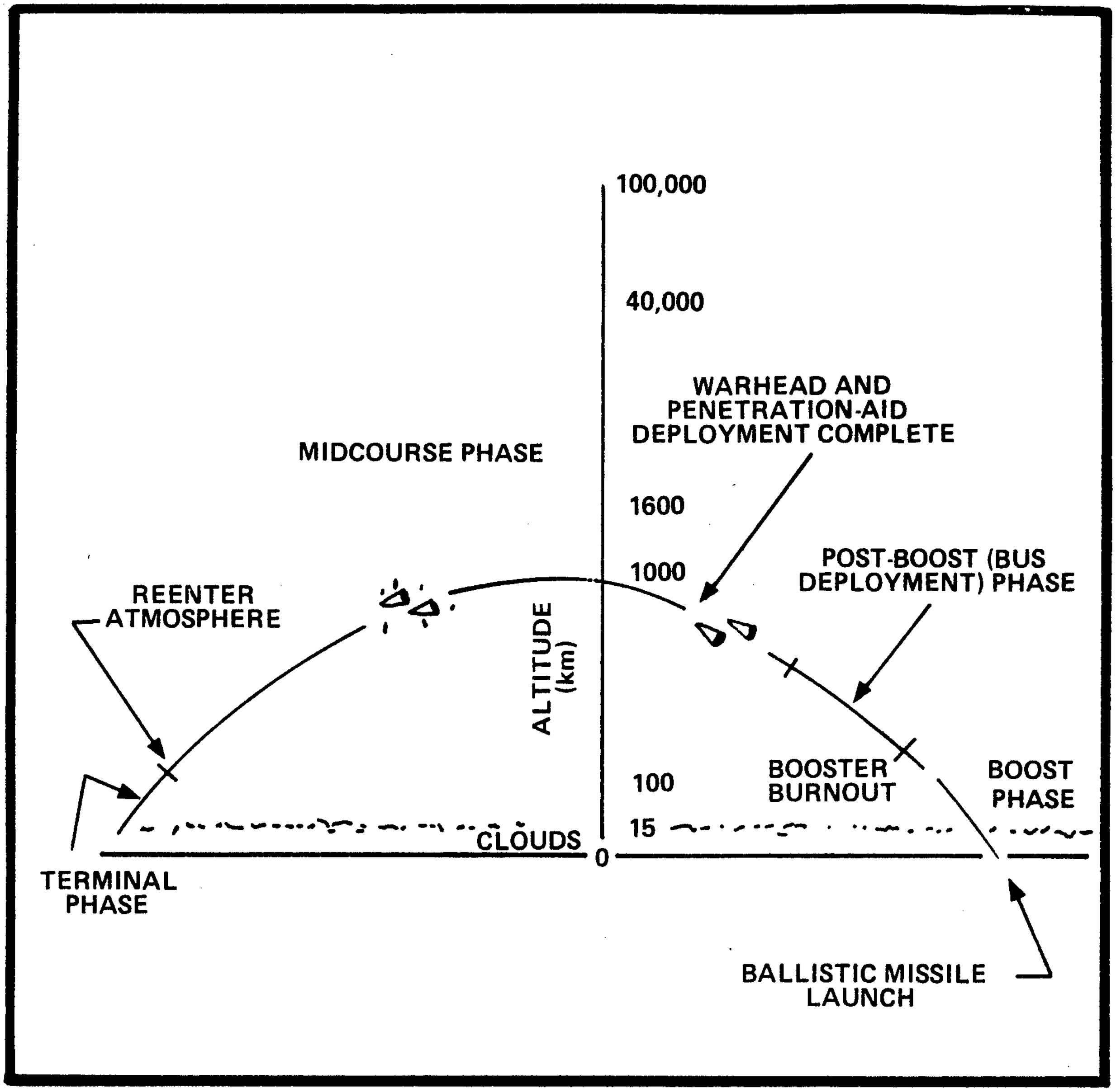
3.1 DEFENSIVE "TIERS"

A traditional approach to defensive systems has been to "layer" the defense using some appropriate scheme for addressing the physical domains or "tiers" of the battle space. Defensive systems are then developed that attempt to defend each tier and allow only a small "leakage" from one tier to the next. Thus, if three tiers are in the system and only a 10 percent leakage rate is allowed for each, the total system will allow only one attacker in a thousand to penetrate. This same philosophy is being applied to the strategic defense against nuclear missiles. Certain factors of atmospheric physics, related to defensive weapon performance, also contribute to this kind of thinking.

Discussions of the Strategic Defense Initiative often refer to the defense domains of "boost phase," "post boost," "midcourse," and "terminal" although other distinctions are made to serve specific needs such as "early midcourse," "high terminal/low terminal," etc. Figure 3-1 depicts the major phases of ballistic missile flight and gives some perspective of the time and space dimensions of each.

3.1.1 Boost Phase

This phase of the ballistic missile flight includes that period from booster ignition through burn-out of all propulsive stages to the separation of booster stages from the reentry vehicle "bus." This phase of the flight offers the most intense observables to defensive sensors. The boost phase also offers



SDC 2004

Figure 3-1. Typical BMD Trajectory

the defender the highest degree of "leverage" because any kills that can be accomplished during the boost phase will abort all of the multiple reentry warheads plus any penetration aids, decoys, etc., carried by that booster .

The advantages offered by defensive weapons used during the boost phase must more than compensate for the problems to be overcome in destroying a booster during its powered flight. The limited time duration of the boost phase, typically less than 200 s, is a severe constraint. During that brief time, sensors must detect the launch and associate the radiated signatures with a specific booster. The booster must then be tracked and defensive weapons directed at the vehicle. Verification of a successful kill must then be made or the associated tracking data is handed over to the next phase of defense.

Defensive weapons used during boost phase must overcome special problems brought about by the short time duration of the phase and the close proximity to hostile territory. Kill mechanisms that attack targets serially will have very limited time to move from one target to the next, acquire the target, fire, and move on. To be effective during such a time sequence, the energy directed against each target must be very intense. The development of sources of energy sufficent for this phase of defense is a significant challenge.

3.1.2 Post-Boost Phase

This phase is also referred to as the bus deployment phase because it is during this period that multiple reentry vehicles and penetration aids are separated from the booster and "bussed" to their independently assigned target trajectories by a post-boost vehicle or "bus." This phase begins at burn-out of the last booster stage and ends with respect to a given reentry vehicle or penetration aid, after that object has been placed in its intended trajectory by the bus. This phase typically lasts for about 5 min by which time the offensive weapons have reached an altitude of some 800 km.

Again, there are plusses and minuses offered by defending during this phase. While the offensive targets are much nearer the space environment of the space-based weaponry, with the duration of this phase being about twice that of the boost phase, new complications are encountered. Tracking the post-boost vehicle becomes much more difficult because its radiated signature is much less pronounced than that of the booster and burns are short duration. Tracking is also complicated by the trajectory changes introduced by the bus maneuvers. Time is again a critical factor because by the time this phase is completed, each booster payload that has not been destroyed will have yielded several independently targeted reentry vehicles plus numerous penetration aids and decoys.

3.1.3 Midcourse Phase

This phase begins once each reentry vehicle has been placed into its intended trajectory and ends when the effects of atmospheric drag begin to be encountered. During this phase, the reentry vehicle flies an unpowered ballistic path through an apogee of about 1200 km. This is the longest part of the flight and lasts about 22 min.

Two problems dominate the midcourse defense scenario. As the bussing phase is completed, each reentry vehicle may be accompanied by a number of decoys and penetration aids. Debris of all sorts may accompany the reentry vehicles to such a degree that the problems .of discriminating between bona fide reentry vehicle and bogus threat are greatly magnified. Then too, because the reentry vehicles are protected against the heat and dynamic forces of reentry, they are inherently protected against the mechanisms that might be used to destroy them.

The time duration of this phase offers the best advantage for this phase of defense. Significant surveillance functions can be carried out in the time available, and complex computations can significantly enhance the probabilities of an accurate discrimination of bona fide targets. The time duration also allows for multiple shots of the defensive weapons if needed.

3.1.4 Terminal Phase

The Terminal Phase begins at the outer reaches of the atmosphere (130 km) down to as low as 30 km. This phase takes less than 2 min to complete.

Although the duration of the terminal phase is very short, several factors act in favor of the defensive systems. By the time a reentry vehicle has reached the terminal phase, it will have been tracked successively through the three previous phases. Its key flight parameters will have been identified and made available to the terminal defense battle managers. As the reentry vehicle and its surrounding decoys and penetration aids encounter atmospheric drag, the difference in their reaction to the atmosphere begins to become manifest. This greatly simplifies the problem of discrimination. Also, because the terminal phase will take place over friendly territory, lines of communication are greatly reduced and weapons deployment enhanced.

The short time span of the terminal defense phase will greatly accelerate the pace of all phases of the battle. Surveillance, target acquisition, tracking, weapon guidance, intercept, and kill must all be accomplished within the 2-min window. Special requirements are placed on the defensive weapons for high acceleration, maneuverability, and speed.

3.2 SYSTEM COMPONENTS

Discussions of SDI system components have most often focused on the classes of weaponry that have been projected. Authors of summary articles have found this approach advantageous because it easily leads to discussion of the more intriguing parts of the program such as high energy lasers, particle beam accelerators, and rail guns.

3.2.1 Strategic Defense Weapons

By far, the most highly publicized portion of the complex Strategic Defense Initiative picture is that of its weaponry. Exotic concepts involving the use of many concepts from high energy physics have been proposed ranging from electrically driven, hypervelocity guns to nuclear powered x-ray emitting lasers. In general, however, the weapons proposed for use in SDI can be divided into two categories.

3.2.1.1 Kinetic Energy Weapons

This class of weapons includes the more common forms that rely on the mass and velocity of their "bullets" or explosives for their destructive force. Rocket propelled interceptors projected for the terminal defense phase are of this category. More exotic concepts might employ electromagnetic accelerators to direct small projectiles at hypervelocity to intercept a target. While these "rail guns" would be highly effective kill weapons, they suffer from a demand for large amounts of electric power and would have a relatively slow rate of fire.

3.2.1.2 Directed Energy Weapons

Weapons of this class include all of the futuristic concepts commonly associated with the SDI program. Lasers of several types would direct intense beams of radiation against their targets at the speed of light. Neutral particle beams and x-ray lasers would be highly effective outside the earth's atmosphere.

3.2.2 Sensors

Those system elements that are used to gather information about the threat environment are generally referred to as "sensors." A major part of the technical challenge of SDI lies in the development of tools and techniques for surveillance, acquisition, tracking, and kill assessment. Sensors may be located on the ground, in near earth orbit, in high earth orbit, or on airborne platforms.

The sensors envisioned for the SDI will serve a variety of purposes. They will observe potential hostile airspace to detect any evidence of an attack. Upon detection of a launch, sensors acquire and track ballistic missiles throughout the flight. Sensors provide the primary information used in defensive weapons control, and they give information vital to the assessment of whether or not a defensive strike has been successful.

Numerous technologies have been projected for use as sensors. Infrared sensors have been demonstrated to have the capability of accurately identifying a reentry vehicle against the cold background of space. Advances in solid state technology have made it possible to project designs for "staring arrays" of photodetectors that could greatly enhance the ability of sensors to observe large areas of space and to track objects with much greater accuracy than heretofore possible. Phased array radars have already proven themselves as excellent tools for acquisition and tracking and can be expected to continue to play an important role in the future SDI work.

3.3 SYSTEM FUNCTIONS

Viewing the Strategic Defense Initiative as an overall system of systems suggests that there are three major interrelated functional parts (Figure 3-2). The "Information Acquisition" function includes the gathering of all data about the environment, threat, battle situation, weapon status, etc., needed by the Battle Management, Command, Control, and Communication function to conduct the battle. From these data, the BM/C^3 function acts as the SDI decision element, exercising control through a broad array of "Controllables" to accomplish the defensive mission. Information gathered by the Information Acquisition function provides a feedback loop to allow the BM/C^3 function to make appropriate adjustments in the conduct of battle.

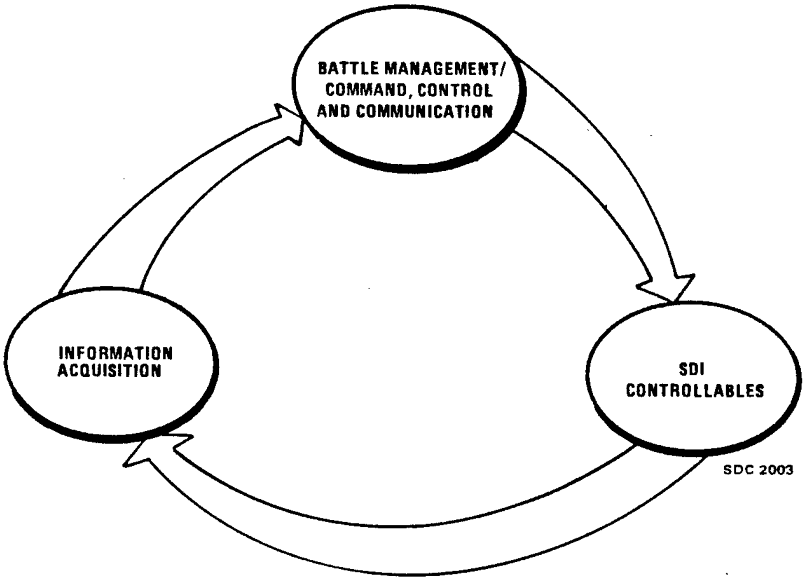


Figure 3-2. SDI Major Functions

In the functional decomposition of the SDI, the assignment of major functions and subfunctions to the various BM/C^3 tiers and nodes requires significant tradeoff analyses. For example, centralized information extraction and resource allocation may provide mathematically optimal results from an idealistic point of view but, in practice, will probably lead to prohibitive costs in processing, communication, and vulnerability. At the other extreme, autonomous systems must make decisions using limited information. In many cases, different logical functions will be co-resident at a node. For example, the Information Acquisition and BM/C^3 functions may be located on the same platforms and share onboard processing elements. The tradeoff studies that must be made to arrive at a functional configuration that has the highest cost-to-benefit relationship represent some of the most intensive future investigations.

The objectives of the SDI Large Scale System Technology Panel focus attention on the Battle Management/Command, Control, and Communication function. The technologies required to support BM/C^3 have been decomposed into two groups: the Information Acquisition and Control algorithms, and the technologies that implement the algorithms, i.e., Computers/Software, Data Management, and Communications/Networking (Figure 3-3).

3.3.1 Algorithms

BM/C^3 Estimation algorithms will use information provided by the Information Acquisition function to compile a Data Base that provides the inputs used by the Decision and Control subfunctions. Control actions are in turn implemented through the SDI Controllables. Taken together, the three classes of algorithms represent the logical definition of Battle Management. Given a system configuration and threat, the algorithms impose requirements on the implementation technologies while the state-of-the-art of the technologies imposes constraints on the algorithms.

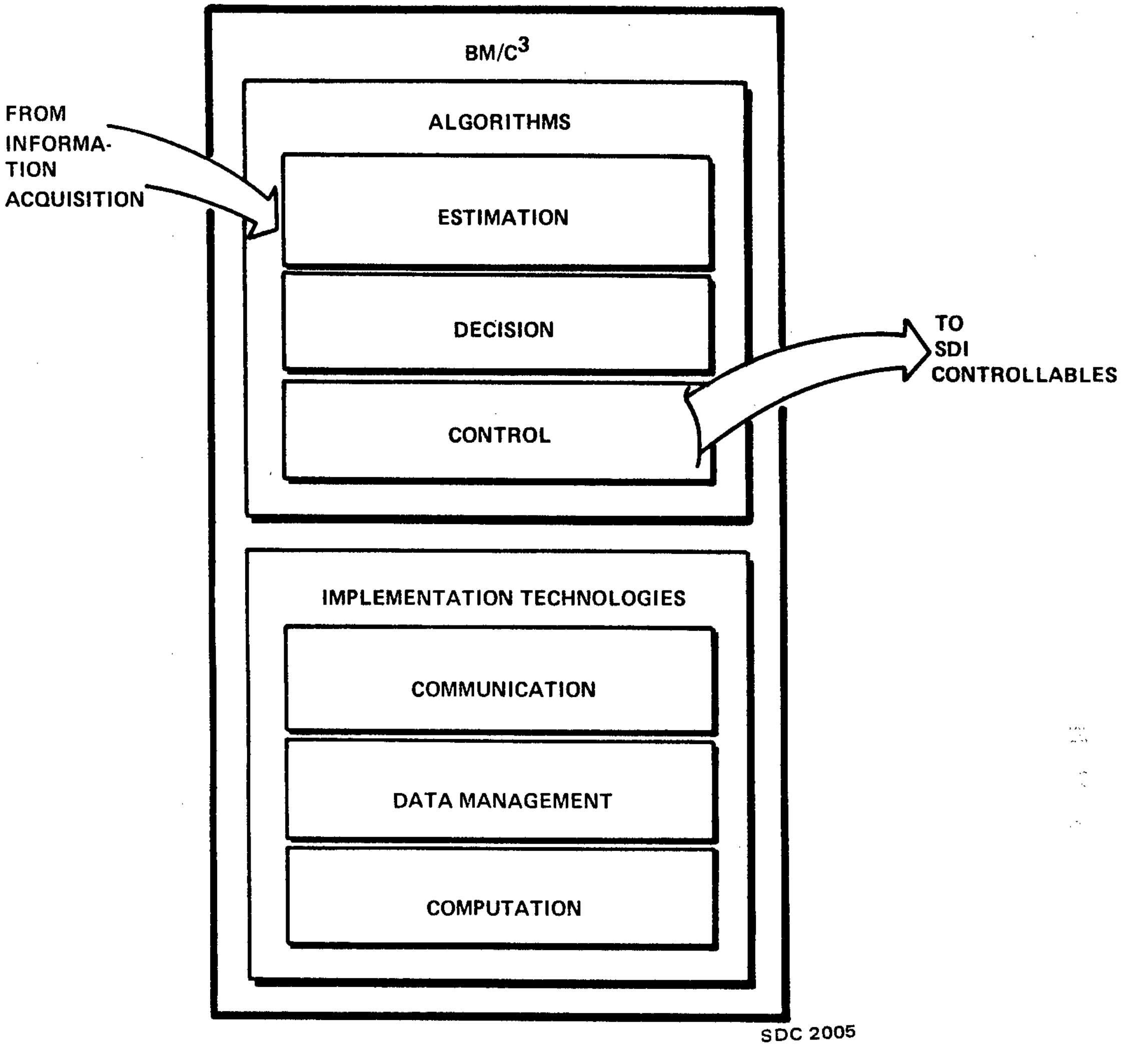


Figure 3-3. BM/C^3 Function Breakdown

3.3.1.1 Estimation

The Estimation subfunction of BM/C^3 assimilates all externally derived information into the Global System Data Base. This may include the correlation of data obtained from various non-SDI sources as well as sensor platforms, radars, aircraft, etc. This subfunction may be further decomposed: for example, the object specific subfunctions of surveillance, acquisition, tracking, correlation, identification, and kill assessment.

3.3.1.2 Decision

Decision-making is a significant part of the BM/C^3 function. It must select an optimum multitiered engagement logic for the dynamic battle scenario. This includes being responsive to priorities set by higher command authorities, coordination with interfacing systems, responding to system losses, planning for the future threat, and the assignment of weapons and sensors subsystems.

3.3.1.3 Control

The BM/C^3 Control function implements the policies made by the BM/C^3 Decision function and is a logical extension of that function. Values for both discrete and continuous control of Sensor and Weapons Systems must be determined in concert with the objectives and constraints established by the decision function. The control and decision functions can be distributed and hierarchical. The degree and extent to which each system element exercises autonomous control over their internal subfunctions is, again, a factor that must be determined by tradeoff study.

3.3.2 Implementation Technologies

The BM/C^3 algorithms are implemented with software/firmware installed on data processing hardware. Information and decisions are maintained in system data bases and are exchanged between functions and the external environment using communication and networking. These implementation technologies can be summed into the Communication, Data Management, and Computation subfunctions that distribute themselves throughout the entire SDI construct, both between and within nodes.

3.3.2.1 Communication

All information transfer needed for the assessment of data, decision-making for Battle Management, and implementation of those decisions becomes a significant BM/C^3 function. This includes communications with weapons centers, sensor platforms, command centers, national command authority, information/data stores, and computation centers. In addition, many networking problems are associated with nodes that are dynamically entering and exiting the battle space.

3.3.3.2 Data Management

Information sources will be distributed throughout the SDI construct. The BM/C^3 data management subfunction will maintain a global data base containing such information as target state vectors, state vector covariances, resource status, target designations, battle strategies, and control selections and use the SDI communication facility to make this information available to the appropriate nodes. Significant problems associated are with data latency, consistency, and security.

3.3.3.3 Computation

The BM/C^3 computation subfunction includes both hardware and software components of the system. The SDI application will stress the state-of-the-art in terms of both computer throughput and software reliability for space-borne assets. Such questions as the optimum distribution of computational elements cannot be resolved until the functional requirements for this sub-function are determined and tradeoffs performed for such factors as cost, risk, and performance, are made against those functional requirements.

3.4 TECHNOLOGY INTEGRATION

The selection of technology areas to address the SDI Large Scale System Issues was guided by the realization that the development of the major subsystems would call on a variety of technology areas. Figure 3-4 relates the SDI Technology Areas to the BM/C^3 functions discussed above. It can be seen that although a technology area may play a lead role in the development of a system

SDI FUNCTIONS SDI TECHNOLOGY SPECIALITIES	ESTIMATION	DECISION	CONTROL	COMMUNICATION	DATA MANAGEMENT	COMPUTATION
ESTIMATION/DECISION THEORY	1	1			2	2
SYSTEM THEORY		2	1			2
CONTROL THEORY		2	1			2
DATA MANAGEMENT	2			2	1	
ARTIFICIAL INTELLIGENCE	2	1			2	
COMMUNICATIONS			2	1	2	2
NETWORKS	2		2	1	2	
COMPUTERS	2		2	2	2	1
RELIABILITY/FAULT TOLERANCE	*	*	*	*	*	*

SDC 2006

1. PRIME RELATION
2. SECONDARY RELATION

* RELATES TO ALL

Figure 3-4. SDI Function/Technology Areas

function, the development of the total BM/C^3 function must draw on a wide spectrum of technologies. Throughout the study period, considerable attention was paid to the integration of technical specialties. Panel members were encouraged to communicate frequently with each other, and meetings were conducted in an atmosphere that was conducive to open debate and the free interchange of ideas and viewpoints.

This study has attempted to integrate the specialty areas of its participants with the objective of avoiding the development of premature conclusions about design configuration. By doing this, it was hoped that study results would identify alternative solutions to some of the more difficult issues facing the program without being constrained by preconceived details.

SECTION 4 - SDI TECHNOLOGY ISSUES

The sections that follow (Sections 5 through 13) contain material provided by the members of the SDI Large Scale System Technology Study Panel. Throughout the study, panel members were admonished to coordinate their work with each other, and during meetings, when all of the panel members were assembled, free and open interchange of ideas, criticisms, and supportive comments was encouraged. Nevertheless, because of the unavoidable overlap in the technology areas, there is some overlap of coverage by the contributing authors. This may be particularly evident in the treatment of such areas as security, reliability, etc. Rather than risk damaging the meaning or scope of a section by attempting to remove redundancies, the authors have chosen to leave each contributed section as unchanged as practical.

Because writing styles vary, and because adequate treatment of the subject technology areas place different demands on writing, the sections that follow vary somewhat in general structure. A common outline was used to guide the writing of all of the sections. It began with an introduction to the technology area, was followed by an identification and discussion of the issues, followed by a discussion of candidate approaches to each issue, and concluded with recommendations for additional research/experimentation.

SECTION 5 - SYSTEM-THEORY ISSUES IN SDI

Carroll Johnson
University of Alabama in Huntsville

The term "system" has been widely used, in diverse ways, to describe various aspects of SDI. In this section of the report we first briefly review the general scientific/engineering idea of a system and summarize the kinds of problems that the technology area known as "system-theory" attempts to address. Next, we examine the SDI concept from the system-theory point-of-view and identify and characterize the major system components that comprise a typical SDI construct.

As in any large-scale application of scientific ideas, the successful application of system-theory ideas to SDI will involve the resolution of a variety of design issues that naturally arise and are shaped by the nature of the application. The main body of this section of the report is devoted to a description of the major system-theory design issues that will arise in the SDI application and a listing of some candidate approaches that may be useful in resolving those issues.

5.1 THE IDEA OF A SYSTEM

The term system has become so pedestrian in recent years, and is so pervasive in discussions of SDI, it is perhaps worthwhile to begin with a brief review of the scientific notion of that term. In the technology area known as system-theory, the term system is used [1] to denote "a partially interconnected set of abstract objects..." or, as we prefer to say it in our own lectures, "an ordered sequence of events." This abstract idea of a system embraces a broad variety of natural and man-made processes that occur in engineering, economics, sociology, etc.

In general, a system possesses four primary attibutes: the system inputs, the system outputs, the system evolution law, and the system state. The inputs are those external actions that act upon the system to affect the system's sequence of events. The measurable behavioral features of that sequence are the system outputs. The evolution law is that rule or principle (perhaps unknown or poorly understood) that governs how the inputs and system initial conditions affect the sequence behavior--in a dynamic as well as static sense. This system evolution law is typically quantified and represented in the form of a mathematical model which may appear as a collection of charts, tables, graphs, mathematical equations, etc. At any given moment of time, the system internal status information needed to evaluate the impending next step in system evolution behavior is called the system state.

Each one of the system inputs naturally falls into one of two categories: control inputs or distubance inputs. If the input can be manipulated by the system analyst, it is defined as a control input (from the analysts' viewpoint); otherwise, it is a disturbance input. The skillful manipulation of control inputs to achieve a desired modification (improvement) of system behavior is called controlling the system. The collection of sensors, computers, actuators, etc., that perform those controlling actions is called the controller.

In practical applications, elemental systems are connected together in various ways to achieve a larger system having more complex and diversified sequences of events that accomplish specified goals. These larger systems can become so extensive and complex in their behavior that the conventional methods of system analysis and controller design become ineffective. Such systems are called large-scale systems and require special techniques for analysis and controller design. In the sequel, it will be shown that the typical SDI construct appears, overall, as a system of large-scale systems, i.e., a galactic system.

5.2 PROBLEMS ADDRESSED BY SYSTEM-THEORY

System-theory is concerned with three main problem areas that can be described, in the order in which they naturally arise, as follows:

1. The Architecture Problem - Determination of the best architecture (layout, organization, interconnection network) for the system pieces--when such architecture is not already fixed. At any given time, the best choice of architecture depends, of course, on the quantity and quality of the pieces and on the specific purpose (goal) of the system at that moment. In some applications, one can reconfigure certain aspects of the system architecture as the system is functioning. This is called real-time reconfiguration.

2. The Modeling Problem - Determination of an appropriate mathematical model to represent the system configuration and evolution law for a given architecture. This mathematical model is an essential tool for analyzing and predicting system behavior in the face of inputs, initial conditions and various uncertainties--when repeated trial and error testing of actual hardware is not feasible.

3. The Control Problem - Determination of the best plan or strategy for manipulating the system control inputs so as to achieve a specified form and/or quality of system behavior--in the face of disturbances, initial conditions, system component malfunctions, etc. This skillful manipulation of control inputs (controlling the system) enables one to effectively coordinate and enhance the performance of both individual systems and interconnected sets of systems.

5.3 SPECIAL TECHNIQUES FOR LARGE-SCALE SYSTEMS

The technological tools for solving architecture, modeling and control problems for elemental systems, and for relatively small numbers of simply-interconnected systems, are rather well-developed. However, as the number and complexity of interconnected systems increases, one usually reaches a point where practical constraints and/or mathematical limitations render those conventional tools, or their application procedure, ineffective. At that point, one is said to be dealing with a large-scale system and special tools must then be employed. For instance, practical limits on the extent of real-time information that can be provided for controlling a large set of interconnected systems may make it impossible to apply the conventional tools of centralized control theory in the usual manner.

The special tools and techniques developed for large-scale systems are based on the notions of aggregation, decomposition, and control decentralization. Aggregation involves the combining and merging (fusion, consolidation) of many effects and features into a few averaged effects and features. By this means, a large-scale system may be approximately represented by a relatively small number of elemental systems. Decomposition can be viewed as aggregation in reverse, where a single large entity is skillfully divided (partitioned) into a number of smaller entities. For example, a single global goal for a large-scale system may be decomposed into a number of local goals, one for each elemental system. Control decentralization is the process of replacing an all-knowing central control authority by a number of less-informed local control authorities each of which has access to only regional system information. By this means the unwieldy amount of information and extensive communication links required by a central control authority can be substantially reduced.

These large-scale system tools and techniques are studied and applied in such professional fields as Control Engineering, Systems Engineering, Operations Research, etc.

5.4 A SYSTEM-THEORY VIEW OF SDI

From the system-theory viewpoint, the generic SDI constructs outlined in Section 3 involve the interconnected and coordinated activities of six major functional components, each of which has the characteristics of a large-scale system. Those six major system components are:

1. The Sensor and Imformation System
2. The Computing and Data Management System
3. The Communication System
4. The Battle Management Information System

5. The Battle Management Decision System
6. The Weapon System.

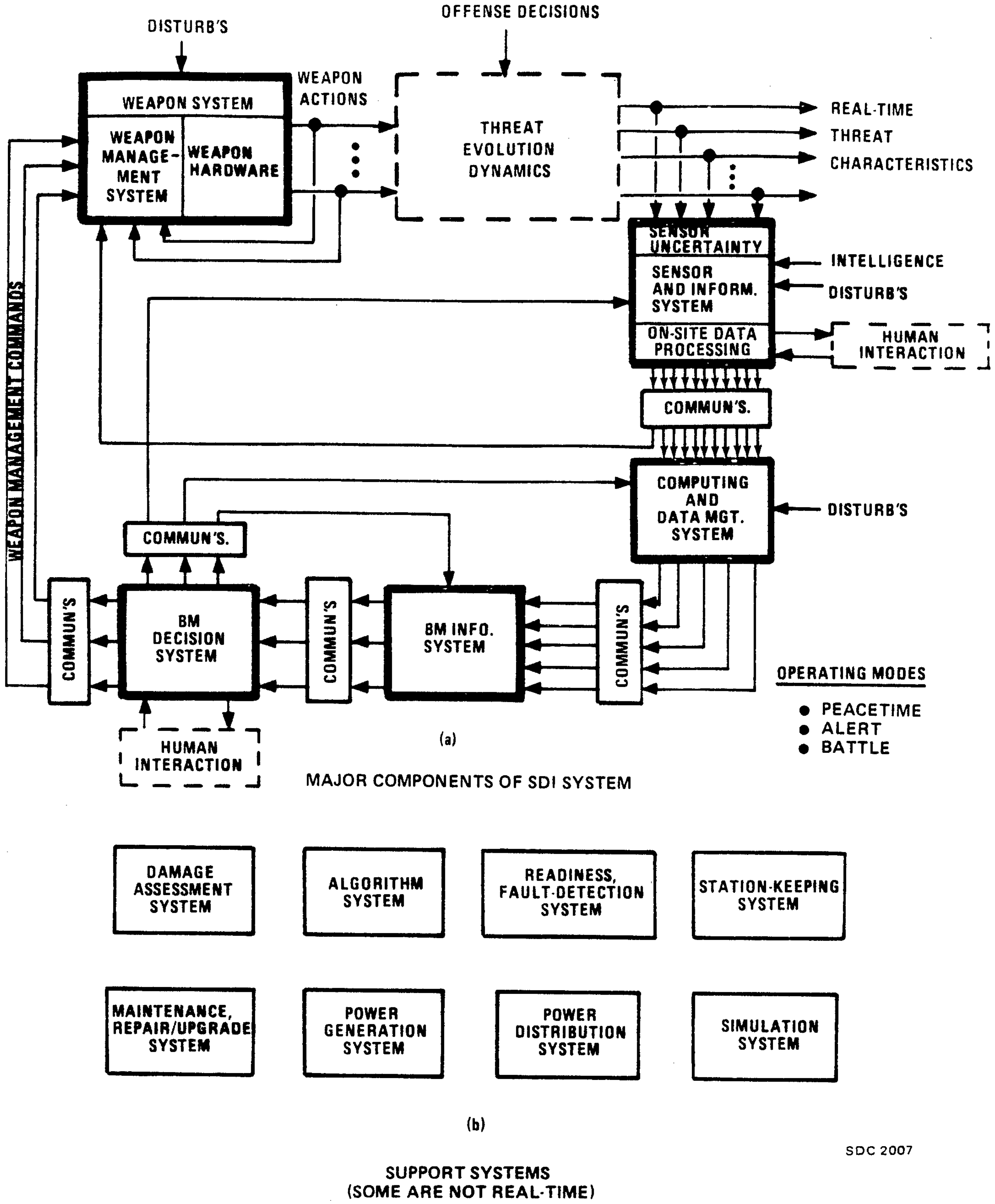
A preliminary system-theory block-diagram illustrating, in principle, how these six functional components are interconnected is shown in Figure 5-1. Each of the six major system components in Figure 5-1 is itself comprised of an interconnection of smaller (elemental) systems, such as individual sensors, weapons, computers, etc., that are spatially and/or functionally distributed and that must operate in a coordinated fashion to accomplish the overall function specified for that system component. Thus, SDI may be viewed as an interconnected array of six major large-scale systems. This array itself has the requisite features of a system so that one can conclude that the typical SDI construct is a system of large-scale systems; i.e., SDI is a galactic system. An abstract representation of such a system is shown in Figure 5-2.

5.5 UNIQUE PERFORMANCE REQUIREMENTS FOR SDI COMPONENTS AND ELEMENTS

The typical SDI construct leads to some challenging system-related design issues owing to the unique requirements on system components and elements. In particular, at the component level, the design must assure that the System Component Function:

- Is accomplished in the face of long idle/ready periods, and a broad range of intra-element faults, element-level failures, hostile battle-environment, etc.
- Degrades gracefully in the face of crippling damange to the system component (i.e., to the components' elements and/or their connecting links).
- Allows revision of function specifications while on station.

These component function requirements translate into hardware requirements on the array of elemental systems that comprise each component. In particular, the system component elements must be designed for long-life and with sufficient redundancy, excess capacity and operating flexibility to:



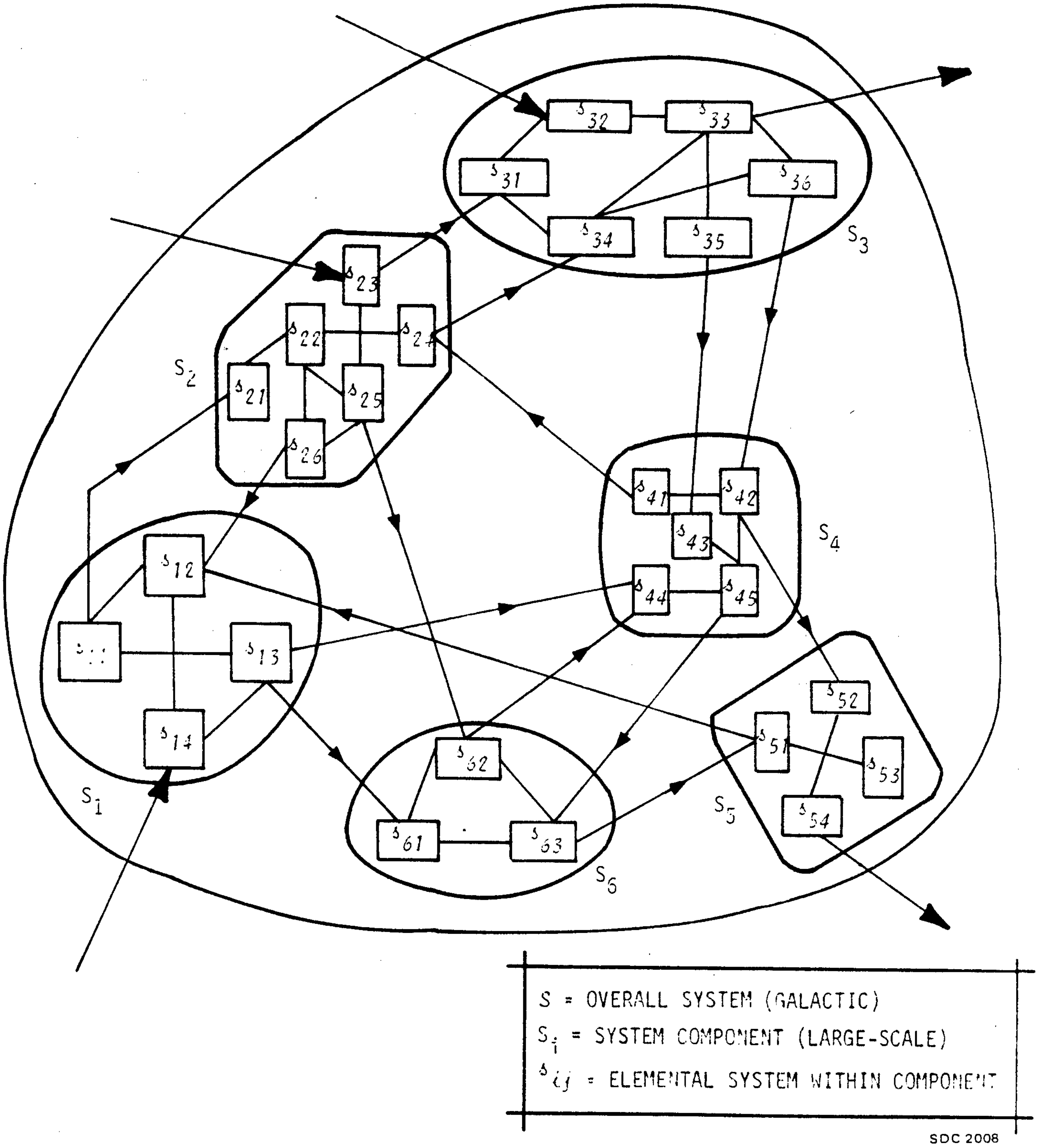
(a)

MAJOR COMPONENTS OF SDI SYSTEM

(b)

SUPPORT SYSTEMS
(SOME ARE NOT REAL-TIME)

Figure 5-1. (a) System-Theory Block Diagram Representation of the Six Major Components of SDI System; (b) Support Systems



S = OVERALL SYSTEM (GALACTIC)

S_i = SYSTEM COMPONENT (LARGE-SCALE)

s_{ij} = ELEMENTAL SYSTEM WITHIN COMPONENT

Figure 5-2. Representation of a Galactic System (=System of Large-Scale Systems)

- Enable intra-element fault tolerance
- Allow for system function healing (recovery) in the face of element disablements, via real-time reconfiguration of the element network
- Receive and implement revised operating instructions.

5.6 SYSTEM-RELATED DESIGN ISSUES FOR SDI

The requirements on system component functions and their associated elements, as outlined in the previous section, lead to system-related design issues involving each of the three major areas of system-theory (Section 5.2). Those design issues may be summarized as follows.

System Architecture Design Issues. The architectural issues in SDI arise at essentially two levels: the component level and the element level. At the component level, the issue is how should one interconnect the six major system components (sensor system, computing/data management system, the communication system, battle management information system, battle management decision system, and weapon system) so as to achieve maximum effectiveness of the overall SDI system. Some aspects of this component-level architecture issue have already been addressed in the SDI pilot-architecture studies conducted in 1985. However, the optimum configuration for networking all six components has apparently not yet been identified.

At the element level, the architecture issue is to find the best interconnection of elemental systems, within each component, so that the function provided by that component is reliably maintained in the face of a specified range of intra-element and element-level faults/failures, battle environment, etc. Morever, the quality of that function must degrade gracefully in the face of crippling faults, damage, etc. This level of performance robustness will clearly require the design of fault-tolerant properties within each elemental system and will also require the capability for real-time dynamic reconfiguration of the element-level architecture. In the latter regard, the best interconnection will consist of a repertoire of predetermined* optimum

*The short time available in an SDI engagement will probably not permit real-time "searching and learning" about optimum alternative interconnection patterns in the face of malfunctioning elements.

interconnection patterns, one for each major group of failure contingencies. The real-time detection and identification of such failures is an important sub-issue within the architecture category.

The element-level architecture issue also involves consideration of means for accomplishing commanded revisions of the component function specifications, while on station. This latter consideration may be another reason for requiring the capability for real-time reconfiguration of the element-level architecture. Further considerations of element-level architecture issues, especially as they relate to the communication system component, are presented in the Networks section of this report.

System Modeling Issues. The purpose of SDI system models (mathematical models) is to provide a means for assessing and predicting system or subsystem performance, under various contingencies, without building and/or testing the actual system. For simple systems, such performance information may be obtained from a purely mathematical anlaysis (solution) of the model equations. However, as in most industrial applications, the SDI system models will turn out to be so extensive in nature, and so mathematically intractable, that one must resort to a computer simulation of the models to "solve" the equations and obtain the desired performance information. It is sometimes found necessary to interface such computer simulations with selected pieces of actual system hardware, when the behavior of those pieces is too complicated to model and simulate. This is referred to as partial-system or hardware-in-the-loop simulation and is particularly demanding on computer throughput rates.

In addition to the highly detailed (high-fidelity) simulation-type models just described, some forms of simplified system models will also be required for the analytical design of controllers for controlling the SDI system. The latter family of models typically involves a high degree of aggregation and/or decomposition, compared to the simulation models, in order to obtain the necessary mathematical tractability for analytical design procedures. Therefore, the system modeling issues in SDI are twofold. First, one must develop a family of high-fidelity models to represent the important component-level and element-level features of the overall SDI system in computer simulation studies. These features will include some rapidly occurring dynamic characteristics which are ordinarily considered negligible in more common engineering applications but which are critically important in the time-stressed environment of the SDI application. In the course of developing these high-fidelity models, one must determine appropriate parameterized structures for the models and then identify the models' parameter values. The sheer size of the overall SDI system, and its multi-hybrid (analog/discrete/digital/logic) nature makes this a formidable task. Secondly, one must skillfully aggregate and/or decompose system features and behaviors, at the basic physics level or through the existing high-fidelity models, to obtain a family of simplified models that will allow effective application of controller analytical design procedures. As before, one must determine appropriate model structures and identify their parameter values.

Considerable innovation and effort will be required in developing effective simplified models because the science of system aggregation and decomposition is still in its infancy and general systematic procedures for handling multihybrid systems like SDI are not available. It is remarked that the procedures for designing decentralized controllers also utilize the tools of aggregation and decomposition to convert overall system multicomponent goals and specifications to simpler, localized goals and specifications.

The family of simplified SDI models will include also a collection of so-called performance models (functional models) that describe the approximate steady-state (i.e., nondynamic) input-output behavior of various subsets of the SDI system. These performance models, which typically appear as charts, tables, or graphs, are often employed in simplified simulation models used for qualitative and rough-cut studies and may be created by analytical, simulation, or hardware experimentation techniques. It is remarked that in some cases, performance models obtained from (perhaps costly) experimentation may be the only available models to represent certain complex phenomena associated with exotic systems.

The Need for an SDI Modeling and Control Simulation Facility. Our discussion of SDI modeling and control issues would not be complete without mention of the need for a thoroughly equipped Modeling and Control Simulation Facility at which the high-fidelity and simplified SDI system simulation models and control algorithms (possibly including some limited hardware-in-the-loop simulations) could be exercised, refined, and validated. The sheer size and complexity of high-fidelity SDI models and control algorithms will require extraordinary programming skills and efforts, and extensive state-of-the-art computer power, to develop and run SDI modeling and control simulation exercises--especially in the case of hardware-in-the-loop simulations where the computer must simulate in real-time. To assure uniformity and quality control in such large-scale programming and simulation efforts, it is imperative that one or more sites be designated as SDI Modeling and Control Simulation (MCS) Facilities and be dedicated exclusively to simulation studies related to SDI system models and control algorithms. It is envisioned that contractors would continue to use their own in-house modeling and control simulation facilities as development tools. However, the dedicated SDI MCS facilities would be the official simulation standard on which all SDI model simplifications and control algorithm design ideas are ultimately tested and proven. It is understood that the idea of an SDI Test Bed facility is already under consideration. In this regard, the Modeling and Control Simulation (MCS) Facility just described should be viewed as a separate entity whose mission is to develop effective system simulation models which will support the analytical design and simulation testing of control algorithms. As such, the MCS Facility will not involve the massive array of SDI hardware, and hardware testing equipment, that would be associated with the SDI Test Bed Facility.

System Control Issues. The term control, as applied to missiles, satellites, boosters, rockets, etc., usually refers to fin deflections, firings of reaction-jets, gimballing of engines, guidance commands, etc. Within SDI, there is certainly an important role, and an abundance of critical issues, relating to such matters. However, in the context of SDI Battle Management, the term control refers to a much broader notion. Namely, management controls consists of those real-time decisions, choices, enablements, authorizations, etc., associated with the real-time allocation of management-type resources. These resource allocations consist of such things as: access to communication links, computers, files, sensors, etc.; enablements of algorithms, data replication, data storage processes, communication alternatives, etc; authorizations of substitutions, modifications, etc.

Within each of the six major system components, (the sensor, computing/data management, communication, BM information, BM decision, and weapon systems) there are a variety of management-type controllables that are, or can be, manipulated in real-time to coordinate the components' elemental systems and enhance component performance.

Some representative examples of these real-time intracomponent controllables are as follows:

- Sensor Controllables - Kind and degree of sensor on-board data processing, sensor network configuration, teaming of sensors, correlation of sensor data
- Computing/Data Management Controllables - Task scheduling, choice of processors and algorithms, fault-tolerant computing measures, choice of when and where to store data, how to correlate and fuse data, when to replicate data
- Communication Controllables - Choice of communication media, coding encryption, channels routing paths, data rates, ECCM, network configuration.
- Battle Management Information Controllables - Kind and degree of data: updating, purging, fusing, substitutions, organization.
- Battle Management Decision Controllables - Choice of current system performance criterion and weighting factors, choice of decision processes and algorithms.
- Weapon System Controllables - kind of weapon, mode of application interceptor choice.

In addition to these intracomponent controllables, there are also intercomponent controllables associated with the actions of one system component on another. Manipulation of these intercomponent controllables is the (or part of the) main function of each system component. For instance, one important function of the BM Decision System component is to control/determine the commands, authorization, etc., sent to the Weapon System component. Likewise, the BM Information System controls the kind, degree, and formatting of information sent to the BM Decision System. In other words, some of the outputs of one system component act as control-inputs to other system components.

The important feature of management-type controls is that the system will usually function, although perhaps not optimally, without real-time manipulation of the controllables. Moreover, there is usually a management overhead cost (time, additional resources, redirection of energies, etc.) associated with the process of prudently manipulating the controllables in real-time. Thus to justify the controlling of controllables in management-type situations, one must show that the corresponding improvement in system performance outweighs the additional overhead cost.

The system control issues associated with SDI Battle Management concerns can be summarized as follows:

1. Identification of those things that are, or can be made to be, controllables at either the component, elemental, or intraelemental level.
2. Determination of system or subsystem performance criteria which can be used to judge the benefit of manipulating controllables.
3. Preliminary determination of which controllables are likely to be cost-effective to control. This is a rough estimation; final determination is made in #5 below.
4. Analytical design of control algorithms and information gathering/processing facilities needed to make and implement "optimal" real-time control decisions from the performance/reliability/robustness

point of view. This task will utilize the simplified system models discussed earlier.

5. Verification of control performance and cost effectiveness via simulation tests and hardware tests, as available.

These control issues are challenging because of: (1) the sheer size and interdisciplinary nature of the overall SDI system, (2) the number and variety of controllables that one can consider, (3) the multihybrid nature (analog/discrete/digital/logic) of the system components, and their controllables, and (4) the SDI system requirements for fault tolerance and graceful degradation in a hostile environment.

Control algorithms are, in a sense, custom tuned to each specific system being controlled. When system characteristics change due to aging, malfunctions, faults, etc., the control algorithm becomes mistuned and system performance degrades. However, in principle, it is possible to design smart control algorithms that can automatically adapt to such changing characteristics and maintain near-ideal performance. Such algorithms, called adaptive controllers, may prove useful in providing the kind of fault-tolerant performance required in SDI system elements.

The preceeding remarks are intended to be only an introduction to the control issues because control issues in SDI battle management and their candidate solution approaches were considered sufficiently broad and important to warrant a separate section in this report. Therefore, the reader is referred to the separate Section on Control Theory for further discussion of this topic.

5.7 CANDIDATE APPROACHES FOR THE SYSTEM-THEORY ISSUES IN SDI

As explained in the preceeding section, the system-theory issues in SDI can be grouped into three catagories: (1) architecture issues, (2) modeling and simulation issues, and (3) control issues. The purpose of this section is to call attention to some candidate approaches that may be useful in resolving those issues.

Approaches to the Architecture Issues. The SDI architecture issues related to system-theory are essentially networking and network reliability/reconfiguration issues. Viewed theoretically, the problem is to find the best way to connect N nodes by a network of links (nodes = the six major system components or, in the case of element-level architectures, the collection of elemental systems comprising one system component). Unlike the simple problems of transportation and telephone networking, however, the nodes in the SDI case must be linked and cross-linked to provide much more than just a means for visiting each node. In particular, each node in the SDI system is operating simultaneously, as an active dynamical process supplying a critical commodity to one or more other nodes. The network must be configured, and dynamically reconfigured as needed, so that this array of simultaneous commodity transfers occurs in a timely manner that optimizes an overall performance criterion (perhaps multi-objective) which includes a specified degree of fault-tolerance and graceful degradation. Networking problems of this sort are currently being addressed within the context of military BM/C^3, distributed computing, communication theory, and reliability/fault-tolerance theory. A detailed examination of those issues, and their candidate approaches, is contained in the four separate sections of this report entitled Communications, Networks, Computer Systems, and Reliability/Fault Tolerance. The reader is referred to the candidate approaches described therein for specific details and references.

Approaches to the Modeling and Simulation Issues. Solution procedures for modeling and simulation problems involving conventional, small-scale systems have been under development for many years and are now well established, [2]. In contrast, the modeling, simulation, and control of large-scale systems is a relatively new field of technology that is still in the early stages of development, [3]-[31]. In fact, much of the currently available large-scale modeling technology is tailored for the case of continuous-time, constant coefficient, linearized dynamical systems, and even in that restricted case, the appropriate way of looking at things is still not settled [22]. Nevertheless, it is possible to identify some broad conceptual ideas in large-scale modeling which, in principle, appear to be useful concepts for approaching the SDI modeling issues.

The first step in any effective modeling procedure is to decide on: (1) the purposes of the model and (2) the system features that are relevant to the model purposes. This step is necessary to avoid cluttering up the model with complexities that serve no useful purpose. The second step is to effectively capture the relevant system features in mathematical-model format. The articulation of system physical features as mathematical expressions will be a challenging task in the SDI case because of the multi-hybrid (analog/discrete/digital/logic) nature of the features that characterize the sensors, computers, communication processes, data management processes, information/decision processes and weapon processes. The result of this second step constitutes the raw model that is useful for highly detailed simulation studies but is typically too complicated for day-to-day simulations and too mathematically intractable for analytical design purposes.

Thus, the third step in modeling is to develop effective simplified models, including functional models, to use as surrogates for various aspects of the raw model. The creation of these simplified models involves all the classical ideas of approximation, perturbation, steady-state, and linearization theories as well as the newer ideas of what we will call spatial, functional, and temporal aggregation.

In the aggregation approach, the basic physical idea is to group together those system features having a common nature and represent their collective effects by a smaller or simpler set of averaged features (i.e., lumped features). Spatial aggregation refers to groupings based on commonality in geometric/geographic location, and its use is typified by the applications in power generation network modeling and finite-element structural modeling. The term functional aggregation is used here to denote those groupings based on commonality of system functional features, which may or may not be located spatially close. For example, a set of geographically distributed loads in a power distribution network may be simply represented by one aggregated load having averaged properties.

We use the broad term temporal aggregation to denote the collection of aggregation methods that are based on a commonality in the time-behavior characteristics of system features. In the literature, those methods go by such names as time-scale methods, singular-perturbation methods, stiff-system methods, mode-separation methods, coherency methods, multimodeling, etc. [15] - [21]. The time-behavior characteristics of interest in such groupings are: the speed of behavior evolution and the extent of dynamic excursion of the behavior variables. For instance, the modeler might group the total array of system dynamical features into two sets, those that evolve characteristically slow and those that evolve fast. On the other hand, if the physical coupling signals (flows) between two interconnected dynamical processes exhibit characteristically small dynamic excursions away from zero, one might approximate the aggregate effect of such coupling actions as zero, thereby obtaining a simplified model that is uncoupled.

If the original raw model is expressed in state-variable format, one can view the various physical aggregation approaches just outlined as attempts to mathematically simplify the model .by reducing the dimension of the original state space and/or reducing the mathematical interaction complexity of the right-hand side of the original state evolution equations. A word of caution, however, is in order here. Namely, if aggregation is approached from the purely mathematical point of view, without regard for the inherent physical structure of the system, one can come up with a mathematically simplified aggregate model that has phantom structural properties (e.g., coupling effects that physically do not exist). Some difficulties may arise if the subsequent controller analytical design is based on a model having such phantom properties.

There is another facet of functional aggregation that should be mentioned here, and that is the idea of grouping those system features that have a common degree of contribution to the system or subsystem performance criterion. Thus, if the modeler restricts the model to reflect only those system features having a major contribution to the performance criterion, a simplified model may result.

A shortcoming of all aggregation approaches to modeling is that the significance or insignificance of a given system feature may be strongly dependent upon the nature of the control (disturbance) actions that will ultimately act on the system. Moreover, the system initial conditions, sensor and actuator characteristics, noises, etc., may also be important determinants in that regard.

It should be remarked that in both the raw model and simplified model cases the system features one must deal with include the parameterized structure of the system and the associated parameter values. In this regard, it is usually preferrable to use structures that are naturally suggested by the physical aspects of the system rather than abstract mathematical structures that may offer little intuitive feel to the modeler in the course of parameter identification and sensitivity studies.

Another important concept for approaching the model-simplification issue is the concept of decomposition (partitioning, tearing, subdividing) which is essentially aggregation in reverse. This concept is useful in decomposing a given large model into a family of smaller, interconnected submodels that can then be dealt with individually [7]-[18]. The main concern in decomposition is the degree of importance of the neglected or approximated information/material flows that physically occur across submodel boundaries. The relative magnitude of those coupling effects is not necessarily the determining factor in deciding their importance since the overall system stability or instability may depend critically on certain "small" flows between submodels.

The decomposition approach is also applicable, in principle, to the task of replacing the overall system performance criterion, which itself may be multi-objective, by individualized local performance criteria for each submodel.

This approach is sometimes called decentralization of the performance criteria and is described in [6], [16], and [29].

So far, we have focused attention on modeling of system internal features. The overall system model, however, must also include dynamic models of system disturbance inputs and command-type inputs. For this purpose, one can adopt the classical random process (stochastic) approach and model the inputs in terms of their means, covariances, and higher moments, as appropriate [16]. Alternatively, one can consider the relatively new waveform-model approach [32] to modeling uncertain inputs that is based on analytical characterization of the individual modes of waveform behavior which occur in random-like combinations in the input. The latter approach is the central idea of Disturbance-Accommodating Control Theory [32], [33]. Both of these approaches are applicable also to the modeling of stochastic parameters, noise, and other uncertainties that occur in sensors, actuators, communication links, and target motions.

After a set of simplified system models has been developed, and the model parameters identified, the next step is to validate the models. That is, exercise the models on a computer simulation using a broad variety of realistic initial conditions, control inputs, disturbance inputs, parameter variations, etc., to demonstrate that the responses of the simplified models are indeed good and reliable approximations to those of the highly-detailed models. Data from actual hardware tests can also be useful in making this demonstration. This simulation exercise would be performed on the SDI Modeling and Control Simulation Facility described earlier in this section of the report.

After validation, the simplified models are ready for use as tools for analytical control design, day-to-day simulation studies, etc. As indicated earlier, the faithfulness of the simplified models (i.e., the importance of the terms neglected) can be influenced by the dynamic actions of the feedback control algorithm being considered. Thus, the faithfulness of the simplified models should be reaffirmed as different controller designs are considered.

Regardless of how well the designed controller appears to work on simulations using simplified models, final conclusions regarding controller effectiveness should be based on simulation exercises using the unsimplified, highly-detailed, raw models. There are some notorious examples in the aerospace industry that demonstrate what can go wrong when this latter step is not taken. This consideration also leads to the need for an SDI Modeling and Control Simulation Facility as described previously under system modeling issues.

This concludes our discussion of candidate approaches to the modeling and simulation issues in SDI.

Approaches to the Control Issues. The first two control issues, which are the determination of candidate controllables and the system/subsystem performance criteria for judging control effectiveness, should be approached from a close familarity with the hardware characteristics and the functions to be performed by the hardware. For preliminary studies, where hardware is yet to be chosen, these familiarities can be replaced by hypothesized generic features and functions based on projections of technology availabilities.

The third control issue, preliminary determination of which controllables are worth controlling, will probably require an approach that uses some form of simulation support based on simplified system models. The idea here is to roughly identify general trend directions in terms of system performance enhancement, so that the designer can avoid pursuing control designs for "low-payoff" controllables.

The fourth control issue is the analytical design of control algorithms and their associated information gathering/processing facilities. This issue must be approached using the family of simplified system models and will involve a variety of specialized control design concepts from large-scale system theory. Those concepts include [13]-[30]: system decomposition; decentralization of control information and decisions; hierarchical, multilevel, and multilayer control structures; coordination control; fault/failure detection; decentralized stabilization and state-estimation; multicriteria optimization; etc. Since the topic of Control Issues and Candidate Approaches is being addressed in a separate section of this report, we refer the reader to that section for more detailed information on candidate approaches to the control design issues.

The last control issue is verification of control performance and cost-effectiveness via detailed simulation and hardware tests. This is essentially a modeling and control simulation issue which has already been addressed in the previous section.

5.8 RECOMMENDATIONS

Large-scale system theory addresses problem areas that are highly relevant to SDI system issues. For the most part, however, the technological tools of large-scale system theory have not yet developed to the point where they can be directly applied, in an off-the-shelf manner, to the SDI system with its vast complexity, robust performance requirements, and multihybrid nature. However, the powerful conceptual ideas behind those tools all appear to be applicable, in principle, to the SDI system theory issues. Therefore, it is generally recommended that support be given to the continued development and generalization of large-scale system theory tools, toward the needs of SDI as described in this report.

In the case of SDI architecture issues related to system theory, it is recommended that efforts be made to more precisely identify and model the functional requirements of the six major system components so that their optimum networking can be studied. Moreover, within each major system component there exists a large-scale array of elemental systems whose interconnection structure is to be designed. The current research and technology in military BM/C^3, distributed computing, communication theory, and network reliability/fault tolerance theory should be be reviewed for its applicability to these intracomponent networking issues in SDI; see also the recommendations in Sections 10, 11, and 12 of this report.

In the case of SDI modeling and simulation issues, it is recommended that efforts be directed toward the system-theory/input-output modeling of the elemental systems that comprise the sensor, computing, data management, BM information, BM decision, and weapon components. These efforts should address the feasibility of developing state-type models and the modeling of discrete-time, digital, and logic-type inputs and outputs. Also, effort should be directed toward ways of simplifying these models, using the ideas of aggregation, decomposition, and decentralization, to arrive at models that can be used for analytical controller and network design. It is further recommended that the subject of SDI-type performance criteria, (measures of effectiveness/performance; MOE/MOP) and ways to decentralize such criteria for decentralized control design purposes, be investigated. The need for an SDI Modeling and Control Simulation Facility has been identified in this report. It is recommended that a preliminary study of requirements for such a facility be initiated, and that procedures used at existing large-scale multihybrid simulation facilities, such as NASA Houston's Space Shuttle Simulation (SAIL), be reviewed for possible relevance to an SDI Modeling and Control Simulation Facility.

In support of the control issues in SDI, it is recommended that the candidate battle-management controllables within each of the six major system components be identified and the associated overhead cost of controlling each controllable be assessed. It is further recommended that ways for modeling these controllables, in the format of control theory, be developed and that decentralized control design procedures be developed that will lead to fault-tolerant/adaptive performance. Further recommendations regarding control issues are contained in the separate section of this report entitled Control Theory.

5.9 REFERENCES

1. Zadeh, L.A., and C. Desoer, Linear System Theory; The State-Space Approach, McGraw-Hill Book Company, N.Y., 1963.

2. Close, C.M. and D.K. Frederick, Modeling and Analysis of Dynamic Systems, Houghton Mifflin Co., Boston, 1978.

3. Mesarovic, M.D., D. Macko, and Y. Takahara, Theory of Hierarchical Multilevel Systems, Academic Press, New York, 1970.

4. Lasdon, L.S., Optimization Theory for Large Systems, Macmillan, New York, 1970.

5. Kulikowski, R., Control in Large-Scale Systems, WNT, Warszawa, (in Polish) 1970.

6. Wisner, D.A. (Ed.), Optimization Methods for Large-Scale Systems with Applications, McGraw-Hill, New York, 1971.

7. Himmelblau, D.M. (Ed.), Decomposition of Large-Scale Problems, Elsevier, New York, 1973.

8. Proceedings 1976 IFAC Symposium on Large-Scale Systems Theory and Applications; Udine, Italy, 1976.

9. Sandell, N.R., P. Varaiya, and M. Athans, "A Survey of Decentralized Control Methods for Large-Scale Systems," Proceedings IFAC Symposium on Large-Scale Systems Theory and Applications, Udine (Italy), 1976.

10. Kokotovic, P.V., R.E. O'Malley, Jr., and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory--An Overview," Automatica 12, 1976, 123-132.

11. Seeks, Richard, Large-Scale Dynamical Systems, (book), Western Periodicals Co., 13,000 Raymer Street, N. Hollywood, CA, 1976.

12. Michel, A.N. and R.K. Miller, Qualitative Analysis of Large Scale Dynamical Systems, N.Y., Academic Press, 1977.

13. Singh, M.G., Dynamical Hierarchical Control, Amsterdam, North Holland, 1977.

14. Sage, A.P., Methodology for Large-Scale Systems, McGraw-Hill, New York, 1977.

15. "Special Issue on Large-Scale Systems," IEEE Trans. on Auto Control, Vol. AC-23, No. 2, April 1978.

16. Sandell N.P., P. Varaiya, M. Athans, and M.G. Safonov, "A Survey of Decentralized Control Methods for Large-Scale Systems," IEEE Trans. Autom. Control AC-23, 1978, 108-128.

17. Athans, M., "Advanced and Open Problems on the Control of Large-Scale Systems," Plenary Paper, Proceedings 1978 IFAC Congress, Helsinki, Finland, p. 2371, 1978.

18. Singh, M.G., and A. Titli, Systems: Decomposition, Optimization and Control, Pergamon Press, Oxford, 1978.

19. Singh, M.G., and A. Titli, (Eds.), Handbook of Large-Scale Systems Engineering Applications, North-Holland, Amsterdam, 1979.

20. Pervozvanskii, A.A. and V.G. Gaitsgori, Decomposition, Aggregation and Suboptimization (book), Nauka, Moscow, 1979.

21. Proceedings 1980 IFAC Symposium on Large-Scale Systems Theory and Applications, (Ed. by A. Titli and M. Singh), Toulouse, France, June 1980.

22. Kokotovic, P.V., "Subsystems, Time-Scales and Multimodeling," Automatica, Vol. 17, pp. 789-795, 1981.

23. Mahmoud, M.S., and M.G. Singh, Large-Scale Systems Modeling, Pergamon Press, Oxford, 1981.

24. Singh, M.G., Decentralized Control, North-Holland, Amsterdam, 1981.

25. Chow, J.H., et al. Time-Scale Modeling of Dynamic Networks, Lecture Notes in Control and Information Sciences, Vol. 47, Springer-Verlag, New York, 1982.

26. Jamshidi, M., Large-Scale Systems: Modeling and Control, North-Holland, New York, 1983.

27. Encyclopedia for Systems and Control, Pergamon Press, Oxford, England, 1984.

28. Kokotovic, P.V., Applications of Singular Perturbation Techniques to Control Problems, SIAM Review, Vol. 26, No. 4, October 1984.

29. Mahmoud, M.S., M.F. Hassan, and M.G. Darwish, Large-Scale Control Systems; Theories and Techniques, Marcel Dekker, Inc., New York, 1985.

30. Teneketzis, D., D. Castanon, and N. Sandell, "Distributed Estimation for Large Scale Event Driven Systems," Chapter in the book Control and Dynamic Systems; Advances in Theory and Aplications, Vol. 22, (ed. by C.T. Leondes), Academic Press, N.Y., 1985.

31. The new journal entitled, Large-Scale Systems, published by North-Holland/Elsevier Science, New York, N.Y.

32. Johnson, C.D., "Theory of Disturbance-Accommodating Controllers," Chapter in the book, Advances in Control and Dynamic Systems Vol. 12 (ed. by C.T. Leondes), Academic Press, N.Y., 1976.

33. Johnson, C.D., "Discrete-Time Disturbance-Accommodating Control Theory," Chapter in the book, Control and Dynamic Systems; Advances in Theory and Applications, Vol. 18, (ed. by C.T. Leondes), Academic Press, N. Y., 1982.

SECTION 6 - ARTIFICIAL INTELLIGENCE

Thomas D. Garvey
SRI International

6.1 INTRODUCTION

The Strategic Defense Initiative represents one of the most severe challenges ever proposed to the technical and engineering community. A deployed SDI system would likely be composed of a myriad of geographically distributed sensors and defensive weapon systems arrayed in a large dynamic communication network. The system would have the awesome responsibility of defending the U.S. and its allies from an all-out nuclear attack that might consist of thousands of thermonuclear warheads intermingled with hundreds of thousands of balloons, decoys and penetration aids. To present an effective defense, the SDI system must be capable of making rapid, autonomous decisions for target discrimination, resource allocation, targeting, and damage and kill assessment. It must be capable of anticipating and planning for future developments in order to marshall its resources against structured attacks. In addition, it must be reliable and robust in an extremely hostile environment, and therefore must be capable of various degrees of self-diagnosis and repair, both for individual elements of the system and for the networks linking those elements. It is natural to look to the field of artificial intelligence for insight and assistance in developing approaches to the task of creating such a highly autonomous system.

The potential application of AI to battle management and command, control and communication (BM/C^3) problems faced by SDI raises a number of significant technical issues. These issues surface at several levels which are defined by the extent to which AI is injected into the development, maintenance, and operation cycles of the SDI system, as well as by the architecture of the final system. While AI could well play a significant role in the development phase of the SDI system (e.g., by providing environments that facilitate the production of efficient, validated code), the issues addressed here are motivated by the potential application of AI technology to BM/C^3 operational, mission-specific functions. Those general battle-management functions likely to benefit most from the use of Al technology include:

- Situation assessment - including recognition of the presence, type, and level of an attack, kill assessment, decoy discrimination (or, more properly, Reentry Vehicle discrimination), and assessment of Blue force status, capabilities, and relative strengths and weaknesses.
- Situation monitoring - tracking "interesting" events and projecting likely situation developments in the absence of current information ("flywheeling").
- Strategy planning - including interpretation and implementation of preplanned Rules-of-Engagement and development of strategies for assigning RV/decoy tracks to defense tiers.
- Tactics planning - the detailed assignment of defensive resources to attackers, hand-offs to other tiers, contingency planning, reconfiguration of networks, allocation of sensor assets, allocation of "expendables" (e.g., probes, A/C-borne ASATs ...).
- Replanning - handling unanticipated emergencies, repairing failing plans, and improving and augmenting plans.
- Human/machine interfaces - facilitating interaction with the BM system at a "cognitive" level that avoids the time and bandwidth required for low-level interactions.
- Diagnosis, maintenance, and repair - of SDI system components and networks.

The AI technology areas of greatest apparent applicability to BM/C^3 include: reasoning and interpretation (including perception, vision and "signal understanding"), planning, distributed AI, and natural language and speech. Areas whose promise is offset by the current low level of the applicable state-of-the-art include learning,* program synthesis, and program verification.

6.2 A STRAWMAN SDI ARCHITECTURE

Currently, anyone aiming to characterize the technological issues prompted by SDI is enormously handicapped by the lack of an adequate system definition. To focus subsequent comments, I will attempt to characterize my view of an SDI system.

A number of competing technical and programmatic goals have been posed by the SDI community. These include very high measures of effectiveness (typically stated as the amount of leakage of RVs permissable through the overall defense system), rapid response, a high degree of autonomy in operation, robustness in the face of errors and system malfunctions, and reliability (meaning that the system will recognize and operate effectively against true threats, will not respond prematurely or without appropriate provocation, and will be resistant to attempts to interfere with, degrade, or subvert its operational capabilities).

The fact that each launch vehicle (in principle, at least) may be capable of spawning a multitude of RVs and decoys argues very strongly for a capability to intercept launchers in the boost or immediate post-boost phase. Intercepting a booster will reduce both the computational requirements and the ordnance requirements for subsequent stages by significant amounts. In fact, the current architecture suggested for SDI is a defense-in-depth, consisting of a number of such defensive tiers, each of which is designed to successively blunt an attack: the first tier is expected to reduce the strength of the attack by a

*Current views in learning research have resulted in a mulitude of theories addressing a variety of learning modes. "Concept Learning" involves the difficult processes of generalization and induction and is unlikely to figure prominently in SDI. "Parameter Learning" is oriented toward adapting given models to a specific situation and will likely be of considerable importance.

certain percentage (for example, 90 percent), the next tier would further reduce the remainder by a similar factor, and succeeding tiers would reduce the overall attack to a level acceptable to U.S. planners.

The requirements for effectiveness and coverage tend to require that at least a portion of the overall system be based in outer space (particularly to provide boost-phase coverage). To provide effective coverage along the entire flight path of an offensive weapon, the system must comprise a number of geographically distributed subsystems. To optimize the effectiveness of the components of the subsystems, it is likely that the functionality of the system will also be distributed (for example, requirements of sensor assets may demand different positioning criteria than required for defensive weapon assets). For reasons of survivability and security, various other functions may not only be spatially distributed but the distribution may be in constant flux. For example, the simplest architecture for a defensive tier is likely to be hierarchical, with a single "commander-in-chief" node and a static command structure. For survivability reasons, however, it will be attractive to have the locus of overall command be highly dynamic, be relocated on a continuous basis and (in particular) be reconstitutable should a critical node be eliminated. This approach raises significant issues with respect to the control of such a highly dynamic network, the local interactions and communications that must be supported, and the overall degree of autonomy of the individual elements of the network.

The concept of successive attrition of the offensive weapons during their passage through the tiers of the defense system raises questions about the amount of interaction among the tiers. Clearly, if the second tier is designed to handle the expected leakage from the first tier, it will be sized to respond to that leakage. It cannot be completely independent of the operation of the first tier, as it will be unlikely to serve its purpose well (if at all) should the first tier malfunction and allow a much larger leakage than specified. This argues for a system architecture where each tier not only depends on the preceding tier working correctly, but also for sharing of information between tiers. The tiers in such a system would not only not be independent structures, but would place a high degree of reliance upon information received from earlier tiers.*

A tiered defense, then, has certain drawbacks. One disadvantage is reduced reaction time for all elements--the nodes in the earlier tiers will have relatively little warning, the nodes in later tier will need to wait upon the results of earlier tiers to complete their own battle management plans. Another problem is the increased level of communication and coordination attendant upon managing a complex, fluid, distributed system, required both for understanding the current situation and for controlling defensive resources.

An alternative to a tightly-coupled architecture is the independent or quasi-independent architecture where a loosely coupled collection of entities accomplish system goals by making and carrying out local decisions. Such architectures would minimize the reliance upon communications in the operational phase of the SDI system, by either providing additional sensor/weapon resources to ensure effective coverage or by providing a distributed control scheme by which individual platforms would make local targeting decisions based upon their observations and knowledge of both target behaviors and the activities of other unfriendly elements. It remains to be seen whether adequate coverage, reliability, and adaptability are achievable with an independent architecture, but the problems associated with such approaches appear formidable at this time. Therefore, the discussion here is oriented toward the more tightly coupled architectures.

*This dependence invalidates the type of arithmetic that attempts to show that overall effectiveness of the system can be computed by multiplying the effectiveness of the individual tiers togeter--arithmetic that is valid only when the tiers operate independently.

The SDI system will have a presence in peacetime as well as war. The system will need to be able to alter its own alertness and defensive condition (with attendant costs) as conditions on earth warrant, and it must attempt to avoid "step-functions" in its operation. The system, therefore, must have a "standby" phase where it is continually monitoring pertinent information sources, an alert phase when global tensions increase, and an active phase where it is actively engaging penetrating weapons. The system must be maintainable, and modifiable in the preliminary stages, and able to switch to an autonomous mode of operation when hostilities commence.

The SDI system must also be "trustworthy." We need to be able to quantify our specifications for and our expectations of successful operation should the system ever be employed. At the same time, we need to be able to guarantee the fact that it will not activate in the absence of a true threat. This requirement for trustworthiness is likely to collide with the overall requirement for a high degree of automation and autonomy in the SDI system. In particular, to be effective, the system itself will be called upon to make critical decisions about weapons allocations, arming, and triggering, with little or no intervention from human operators. This will require an ability for the system to interpret and apply rules-of-engagement autonomously and securely.

To summarize then, my strawman SDI architecture is composed of several largely-independent defensive tiers, each of which has the responsibility both for reducing the size of an attack by a significant fraction, and for passing information about the attack to the next tier. Within each tier will be a collection of (likely) disparate sensors and defensive weapon systems. This collection will be managed with considerable autonomy within the tier, where the actual location of the manager will either be distributed throughout the system or changed frequently. Each tier will achieve its results by allocation resources both to keep track of the developing situation and to defeat the offensive weapons.

6.3 AI BACKGROUND

Just as there is no current consensus on SDI structure, architecture, and functionality, it is similarly difficult to find an agreed upon definition for the field of artificial intelligence. This difficulty worsens as AI technology finds its way into more traditional computer science applications. In order to provide a common ground for discussion, I will attempt to provide a working description of the field here.

The motivation for working on AI typically stems from one of two sources: the first is a desire to better understand the nature, functions, and mechanisms of natural intelligence; the second is a desire to provide more functionality and capability to computer systems--to make computers smarter. In the first instance (the scientific model), AI may be thought of as "experimental epistemology," where the computer provides a tool for performing experiments to verify models of human intellect. In the second instance (the engineering model), AI may be viewed as an extension of more traditional forms of automation. For application domains such as SDI, the engineering model is more appropriate, as it tends naturally toward an evolutionary approach to achieving functionality. However, the ultimate challenge presented by a full-fledged SDI system will probably require advances in the science of AI as well as the engineering of AI.

To discern what is meant by artificial intelligence, we must have recourse to a limited definition of natural intelligence. In particular, an intelligent entity is normally able to acquire information via its senses directly from the environment, to interpret that information in order to understand events and activities in the environment that are relevant to its own intentions and requirements, to determine courses of action that will accomplish its goals, to carry out the appropriate actions to achieve its intent, to close the loop by monitoring the progress towards its goals, and when discrepancies between expected and perceived events are recognized, to take remedial actions. Other attributes of intelligence include an ability to learn from past experience and an ability to communicate with other entities. The underlying capabilities required to accomplish these activities have become the focus for much of AI research.

The general areas of AI technology that aim to understand facets of intelligent behavior include: reasoning and interpretation, perception, planning, natural language, and learning. Since the accomplishment of complex tasks often requires capabilities beyond those possessed by any single individual, one of the topics addressed in AI research is "distributed AI," and is concerned with the organization and interaction of collections of distributed entities, many of which have specialized skills. In the following section I address in greater detail a few of the areas of AI likely to have the greatest impact on the development of an SDI system.

Certain characteristics are typical of "AI problems." While the majority of problems successfully solved using computers admit to closed, algorithmic solutions, the solution to an AI problem typically involves generating and searching very large "possible-solution-spaces" to select the correct solution. Large solution-spaces are generated fairly easily, for example, in the analysis of two-person games such as chess or checkers. In such games, the first player selects his move from a number of options; the second player then has a new set of options based on the move chosen by the first player from which to choose, and so on. The number of possible move sequences grows extremely rapid as the number of turns increases. AI appears to offer the greatest potential in those situations where it is difficult to predict a priori the actual environment where a solution will be required--and therefore the structure of the solution itself--and where encoded knowledge can reduce the effective size of the search space.

Much of AI work, then, is oriented toward the control of potentially huge search spaces. Early work in AI was devoted to uncovering various domain-independent heuristics that could be used to prune the search space (while guaranteeing not to remove the true solution), thereby hastening progress toward the discovery of the solution. These search strategies were focused on the search process itself, rather than the specific problem being addressed. More recent work in AI has emphasized the use of domain-specific heuristics, oriented toward the actual problem at hand in order to focus computational resources along fruitful lines.

The effective use of knowledge requires an appropriate means for representing that knowledge within the computer as well as procedures for manipulating knowledge. The selection of the best representation depends upon the operations that must be performed on that knowledge. Typically, simple representations lead to complex processing requirements and greater generality, while complex representations tend toward relatively simple processing and relatively narrow utility. The selection of an appropriate representation is one of the key, early decisions that must be made when attacking a new problem; many problems demand the use of multiple, non-monolithic representations.

As AI approaches to larger classes of problems prove successful, there is a tendancy to develop tools to facilitate the representation and solution of new, distinct problems of the same class. Expert-system research, for example, has yielded a number of commercially available tools [1,2,3], designed to permit the non-AI-expert to attempt to develop expert-systems solutions for his problem. This approach is necessary in order to enlarge the population capable of developing solutions to AI problems and to facilitate the transfer of technology from the laboratories into the field. However, the possession of an AI tool does not automatically create successful AI system developers out of ordinary programmers; it is by no means a panacea for the solution of Al problems.

It will be impossible to employ any present AI functionality and meet the stringent time constraints without the introduction of vastly different computing hardware. This will require true synergism between software approaches (such as problem decomposition) and supporting, multiprocessor hardware implementations.

The development of hardware architectures specialized for the types of parallel, asynchronous computation inherent in AI systems offers potential solutions to the real-time requirements. Of particular interest are processor architectures such as those being developed under the DARPA Strategic Computing Program (SCP) [4]. These are typically multiprocessor arrangements with flexible, high-bandwidth intercommunication among processors. Effective programming support and optimal (or even effective) utilization of processor resources to achieve solutions to complex problems is likely to be a research problem for some time to come.

As applications of AI are developed, it will become critical to be able to access and manipulate ever larger knowledge bases that are likely to consist of physically distributed components. It is ironic that the state of the art of AI today enables us to develop systems that are capable (to some degree) of emulating an expert's solution to a problem in his area of specialization. At the same time, the types of common sense reasoning activities carried on by people in everyday life are well beyond our capabilities. One reason for this is that it appears that people draw on a huge, diffuse collection of experiences and methods in order to interpret and act upon everyday stimuli. Until we are capable of creating and controlling similarly huge knowledge bases, we are likely to remain profoundly limited in our general AI capabilities. Furthermore, while automated learning techniques offer some hope of enabling an AI system to develop its own large knowledge base in a relatively autonomous fashion, those techniques in turn will require large knowledge bases of their own to relate new perceptions to past experiences. At this point, the amount of knowledge the BM/C^3 system will need to encompass is unclear; however, it is unlikely that learning technology will be of critical importance to SDI within the proposed development time frame.

Finally, AI researchers must guard against an excess of technical hubris induced by self-generated hype. AI researchers have identified a number of exceedingly difficult problems that form the basis of the field. In most cases, relatively minute inroads have been made in the solution of these problems. For example, AI systems can represent and draw conclusions in relatively simple situations and solve relatively simple problems. The promise offered by those inroads (in, say, expert systems) has been so great as to distort completely, in many cases, the perspective that ought to be maintained. As a result AI has taken on magical attributes, and the expectations of customers are exoatmospheric. It is critical to the orderly advancement of the field to maintain realistic expectations; that is, that the potential contribution of AI to solving difficult, complex problems is quite high, but significant effort remains before the potential will come to fruition.

6.4 KEY AI TECHNOLOGY AREAS

AI is concerned with the use, manipulation and understanding of knowledge. The areas of AI technology that would appear to offer the greatest potential for exploitation in SDI are those aimed at understanding the true situation based upon observable evidence, those aimed at effecting a controlled change to the current situation, and those aimed at facilitating intercommunication among intelligent entities. In addition, since these activities must take place utilizing elements distributed functionally and geographically, interpretation and planning methods must be designed with this distribution in mind. In this section we shall briefly describe each of these key areas, attempting to provide insight regarding the central themes and goals of the technologies, the current state-of-the-art, the criticality of the technology to SDI, technical hurdles that must be overcome, and a recommendation for future research to bring the state-of-the-art to a level consistent with the requirements imposed by SDI.

6.4.1 Reasoning and Interpretation

6.4.1.1 Background

Some of the earliest work in AI attempted to develop automated methods for proving theorems in logic. This work was motivated by the premise that logical reasoning might serve as a model for human reasoning: if a problem could be stated as a theorem to be proved, and applicable knowledge represented as axioms of a logical theory, then a proof of the theorem based on the relevant axioms would lead to a method for solving the problem. Effective proof procedures did emerge (most notably the resolution principle [5]) which offered a methodology for automatic deduction. Systems based on the resolution principle were used for early robot planning systems.

More recent work in the use of logic for problem-solving has concentrated on the development of logic programming languages, most notably PROLOG [6]. These languages enable a programmer to provide a set of facts and rules of inference about objects of interest and then ask questions about them. The system performs the deductions needed to answer the questions. While a key advantage to PROLOG is the ability for the programmer to specify "what" he wants done as opposed to "how" he wants it done, thereby freeing him from the details of the algorithms involved. At this time, however, this advantage is a mixed blessing in that it is relatively difficult to provide control information to the PROLOG system in order to capitalize on problem-solving knowledge.

A drawback of most logic-based systems has been the difficulty of introducing domain-specific information for use in controlling the process of achieving a proof. Since much human intelligence seems to involve knowledge about how to solve problems (as opposed to merely knowing facts), this was a serious deficiency, and led to research focused on the control of the proof process. A significant outcome of this work is the subfield of expert systems [7].

An expert system is a computer system that attempts to mimic the way an expert in a specialized domain would approach the solution of a problem. The most common knowledge representation in current expert systems is the "production rule," an IF-THEN rule that describes how to infer the THEN part of the rule from knowledge of the IF part. Successful limited applications of expert systems have been demonstrated in a variety of intellectual domains, including medicine, geology, finance, and system design.

Expert systems have the advantage of a uniform, modular representation. The production-rule formalism can be used, not only for representing the basic facts of the application domain, but also for encoding control knowledge describing how and when to use those facts. Individual rules can be inserted, removed, and modified independently of one another (of course, the modification of a rule may lead to unexpected changes in the behavior of the overall system, but it will have no direct impact on any other rule). The production-rule representation lends itself well to the development of expert-system "shells." These are expert systems with the knowledge base stripped away, leaving just the inference mechanisms and supporting structures. Users may then develop their own knowledge bases within these shells (commonly called expert-system building "tools"), thereby creating new expert systems without having to replicate the base programs. These tools effectively provide new programming languages that, in many cases, facilitate the development of fairly complex systems.

Expert systems technology seems most appropriate for problems where the relevant knowledge is accessible (e.g., an expert exists who can introspect and explain his reasoning sufficiently well that the relevant knowledge can be discerned) and consists of large numbers of facts rather than a concise, unified theory, where the facts are relatively independent from each other, and where the knowledge is easily separated from the operations that may be performed on it. Most of the current expert systems may be characterized as diagnosis or "categorical reasoning" systems (a system that determines which of a--possibly large--set of categories best describes the situation at hand).

Two primary types of reasoning methodologies are used in expert systems extant today: those that perform logical (i.e., boolean) computations, and those that provide some form of uncertain reasoning mechanism. Most of the latter systems rely on some heuristic interpretation (or extension) of Bayes' Rule of conditioning. One important focus of research in expert systems work today is the development of effective alternative methods for uncertain reasoning [8,9].

A key role for reasoning and interpretation technology is to support perception--the process of relating sensed information to prior knowledge to understand the source of the sensed data. Perception can be viewed as attempting to find an optimal embedding of a perceptual model in a mass of sensor data [10]. For visual recognition of objects, the "situation" model could be an actual, geometric (iconic) template; for more abstract perception (i.e., situation assessment), the models could represent abstract, probabilistic relationships and behaviors among entities. As sensor systems typically provide only indirect evidence for the presence of the actual situational elements of interest, an inferencing capability is key for perception. In particular, sensed data can be expected to be imprecise, frequently inaccurate (sometimes just wrong), incomplete, asynchronous, and of varying degrees of "granularity." To overcome these defects and to compose an integrated, coherent picture of a situation, a reasoning capability is required. The better the individual sensors are, that is, the more precise, accurate, and complete their coverage, the less requirement there is for sophisticated reasoning techniques.

6.4.1.2 The Role of Reasoning in SDI BM/C^3

Situation assessment is a key, generic function required at all levels of an SDI system. The assessment of any situation implies the understanding of current data in the context of prestored situational knowledge and the current model of the evolving situation. Situation assessment functions that must be supported in SDI include:

- Monitoring platform, network, and system health and diagnosis (and prediction) of malfunctions
- Interpretation of sensor data
- Discrimination of RVs from decoys
- Recognition of attack and determination of attack structure.

Included in situation assessment will be the recognition of attacks on the system itself (for self-defense countermeasures), and the IFFN (interrogation: friend, foe, or neutral) function.

An important aspect of true situation understanding is an ability to make accurate, defensible estimates regarding future developments. In particular, understanding a situation implies understanding the activities and processes that actors in the situation are involved in. This means that situation assessment goes well beyond simply compiling an enemy order-of-battle (possibly) valid at a single instant in time. Rather, it includes determining the roles and intents of the elements of the order-of-battle, and being able to use this knowledge to predict likely developments.

At a basic control-theory level, selecting and monitoring the appropriate statistical model for an evolving stochastic process would be useful for a large variety of tasks, such as tracking targets. A role that an AI system might play would be to continually evaluate the quality of the model's predictions by comparison with observed data. Should the system's estimates and observations begin diverging, the system could hypothesize and evaluate alternate models or higher-order parameterizations to select a more faithful model, and to continue developing predictions.

This process-oriented view of situation assessment is mandated by the need to estimate likely future situations for planning purposes. However, an important side benefit of this view is that.the ability to estimate likely changes to the situation enables the system to maintain a plausible model of the situation, even in the absence of up-to-date information. This will help alleviate the difficult problem of database synchronization by providing a means to update situation models in a predictable fashion, thereby enabling the situation assessment modules to "freewheel" in the absence of current data.

6.4.1.3 Criticality of Reasoning and Interpretation to SDI BM/C^3

The SDI system will be capable of bringing a great deal of firepower to bear on targets which it must select autonomously. The ability for the system to ascertain the characteristics of the situation to determine an optimizing (or, at least, a satisfying) solution to problem of neutralizing a structured attack is of the highest criticality.

In effect, the degree to which high-performance inferencing capabilities are required in SDI is a function of the quality of the system's sensors and weapons, and of the system designer's ability to predict an enemy's capabilities and probable actions and counter-actions. If the sensors are completely reliable and effective, the inferencing system's requirements decrease, as it will not be required to construct and verify an overall assessment of the situation to clarify or verify sensor readings.

The relative inaccessability of many of the elements of an SDI system demands that the individual components be capable of self-diagnosis and repair on a routine basis, independent of operational employment of the system. To maintain effectiveness under attack, the SDI system must be able to perform emergency diagnosis, repair, and reconfiguration.

6.4.1.4 The Current State-of-the-Art

Expert systems have been constructed (and are in use) for performing diagnosis of problems or faults in relatively narrow application domains and for aiding in the configuration of complex computer systems. While capable of impressive results in their domains, these systems are not time-stressed, do not reason over time, and do not deal with very large numbers of objects. They do not have to worry about a hostile opponent attempting to thwart their plans. At this time, it is fair to say that most AI systems do not handle the noise, inaccuracies, and other complications of real-world applications. To approach a large-scale problem, two alternatives arise: either scale (i.e., simplify) the problem to where it fits the available technology, or enhance the state-of-the-art to handle the critical ramifications of the problem. It is unlikely that SDI problems will scale well and still yield useful results; therefore, we will need to enhance the state-of-the-art.

Limitations of current research in reasoning and represenation have to do primarily with the lack of effective, proven representations for real-world knowledge, our inexperience with very large knowledge bases, and an inability to perform inductive operations such as generalization in a sound, effective manner.

Most AI systems represent knowledge as a set of discrete statements about the world. Most objects in the real world exhibit continuous properties; most processes represent continuous changes over time. We need to develop computational representations for time and events that take place over time and space, for uncertainty, circumscription and ignorance that allow for developing models of dynamic, partially understood situations, and for continuous states and processes.

Most AI work to date has addressed problems that require relatively small amounts of knowledge. Providing effective support for SDI will most likely require that we develop and use very large KBs (with which we have relatively little experience). Development of specialized computer architectures will be necessary to permit the effective use of such exensive KBs.

6.4.1.5 Recommendations for Research in Reasoning and Interpretation

Extensive work is required to facilitate:

- The development of effective methodologies for reasoning over time
- The development of effective representations for uncertainty and belief
- The creation of useful approaches for controlling the activities of an interpretation system
- The development of new, specialized hardware architectures "tuned" to reasoning tasks.

The proper way to address these issues (so that their solution supports SDI) is to select research problems that are microcosmos of the SDI problem and whose solution requires effective approaches to the issues. The research problems must stress the integration of both software and hardware (both for sensors and for computation), have realistic time pressures, be sensitive to external environmental influences, and produce results that are scalable to the SDI problem.

6.4.2 Planning

6.4.2.1 Background

Planning is the construction of a detailed method that, when put into action, will accomplish certain desired goals. Planning activities span a spectrum from designing high-level strategies, whose implementation will require years, to the simplest actions such as choosing the next place to step when walking. Planning is one of the most pervasive of human intellectual activities.

Effective planning requires knowledge of the environment in which the plan is to be executed (the target world) and an ability to reason about (or simulate) the effects of actions, both those of the plan executor and those of other agents in the target world. Plans are typically sequences of primitive actions, each of which will incrementally modify the environment until the desired situation is attained. As sequences of actions become longer and longer, the planner's ability to estimate the likely state of the world diminishes. For this reason, most plans will specify activities only to a limited extent, allowing the executor to accommodate to the actual environment when the activity is imminent or occurring.

Because of its importance and difficulty, automatic planning has long been an important area of AI research. However, no automated systems currently exist for planning activities involving complex, dynamic ("real-world") situations at this time. There are several reasons for this shortcoming:

- Artificial-intelligence-based planning systems have largely concentrated on formal approaches to planning in detail, rather than on structured (e.g., hierarchical), performance-oriented approaches.
- Most planning schemes assume a static world
- Effective schemes for representing dynamic, uncertain situations have not been available
- Very little work (outside of the somewhat stylized world of game playing) has been performed on the problem of planning activities against an intelligent adversary.

SDI operations, in particular, will include activities that take place over extreme range of time scales. To initiate combat, plans must be set in motion far in advance of the anticipated activities to ensure the coordinated application of force. Once hostilities have commenced, however, any actual battles could be as short as a few minutes, and individual operations may require reaction times of less than a second.

6.4.2.2 Role of Planning in SDI

Many of the functions in SDI will require an efficient utilization of resources to achieve the necessary levels of effectiveness; this is likely to be most critical for weapons allocation, but possibly of equal importance for situation assessment. Configuring and reconfiguring communication networks, implementing self-defense measures, and repair and maintenance of equipment all require planning functions. Planning is a pervasive and critical function for SDI.

6.4.2.3 Requirements for Planning in SDI BM/C^3

Three primary aspects of SDI (and, in general, real-world planning) tend to set it apart from problems addressed by more traditional AI planning work [11]. These are:

- Varying degrees of uncertainty associated with all phases of planning and execution
- The dynamics associated with typical situations
- The difficulty of specifying detailed goals crisply.

Each aspect of the problem contributes to the difficulty in automating the planning process.

Uncertainty about situations and potential activities derives from a number of sources including:

- Current uncertainty and ignorance about the identities, dispositions, capabilities, missions, and goals of adversaries

- Uncertainties regarding the status and current disposition of the planner's own forces
- The "decay" of information content of a knowledge base over time
- The nondeterministic nature of operations
- Cover and deception operations by an opponent.

The effects of uncertainties upon the planning process are manifold. The degree to which a plan may be formulated in detail is strongly determined by current uncertainties and by the degree to which estimates of future situations can be made. In particular, as uncertainties are compounded, a point called the planning horizon is reached beyond which further planning is likely to be ineffectual.

Uncertainties accumulate when operations with uncertain outcomes are chained together. Whenever an operation could have a multiplicity of possible results, each possible result must be considered in determining the potential state of the world. Each outcome may lead to a different possible world situation. The rapid expansion of the number of possible situations results in the planning horizon.

As mentioned, one effect of situation dynamics from the planner's point of view is to increase the overall uncertainty about the target situation. Another important effect is to increase greatly the complexity of creating a plan. This requires robust planning techniques capable of organizing and managing a large search space. Additionally, since planning may take significant amounts of time, the cost of creating a plan must be factored into the overall cost of carrying out an operation.

Most planning work in AI has dealt with goals specified as crisp, logical statements such as (GRASP ROBOT <object>). A successful plan is one that results in making such statements TRUE. Most real-world planning situations, however, require that a (possibly large) collection of conditions be met. In many cases it will be practically (if not theoretically) impossible to meet all conditions simultaneously, and therefore another method of specifying goals will be required. In particular, the planner must decide on the degree to which each condition must be met and what combinations of (partially met) requirements will be acceptable.

This suggests an approach similar to that used in optimization, where an objective function is created that summarizes the goals of the system. By attempting to maximize the objective function, a planning system will work towards its goals.

6.4.2.4 Recommendations

Planning work, to be useful in real-world applications, needs to address aspects of problems that have been traditionally simplified away or not explicitly recognized. In particular, planning research must address the following issues: temporal reasoning, uncertainty, nondeterministic operators, continuous states, and noncrisp goals, as detailed below.

Time is of the essence--a real-world planner is primarily concerned with changes over time; changes brought on by the actions of an opponent, by the world (random, entropic changes, and projections of fully and partially specified trajectories), by one's own forces, and by neutrals. For the planner: What is HAPPENING is more important than what IS. Effective process models are required to represent explicitly allowable (and/or probable) changes over time.

Uncertainty is a fact of life for the planner. One important implication of this is that there will normally be a finite planning horizon, beyond which it is impossible to foresee events precisely enough to enable any sort of effective planning. This horizon will be a characteristic function of the processes and our current knowledge of appropriate state variables; it should be computable. One of the key resources that must be considered is the time (and cost) required for planning itself. This may induce an earlier planning horizon than might otherwise be obtained.

Operators for causing a change from the current state to a preferred state are rarely deterministic. Rather, they induce their own uncertainty whenever activated. This means that any time an operator is required in a plan, the resulting (planning) world must be split into a number of possible worlds, with the likelihood of each based on the likelihood of the associated operator outcome. This also means that the plan must include <monitoring> and information-gathering operations, directly.

Most real-world processes are continuous (but not necessarily linear) and the planner must also deal with continuous processes. It is computationally advantageous if the process can be partitioned into linear (or linearized) subprocesses, with nonlinearities expressed at the boundaries of the subprocesses.

States represent position and motion parameters (among other things) for activities in the real world and, therefore, are rarely crisp. Many state descriptions are the result of perceptions, where any implied crispness may be an artifact. This will require a constructive means for determining appropriate state boundaries. The resolution implicit in the choice of boundaries will be a function of the level of the plan.

Since descriptions of states may be arbitrarily detailed, it is critical that the detail be managed by appropriate abstractions of process and operations. A typical approach to rationalize the planning process is to plan hierarchically, solving coarse, high-level abstractions of the problem before proceeding to ever greater detail. Other approaches might be resource-oriented, where constraints on key resources are taken into consideration first, and then efforts are made to utilize less important resources.

Resources will be what most real-world planning is concerned with: allocation, evaluation of their effectiveness in the current context (or environment), estimation of their cost, and anticipation of their need. If the search space can be effectively constrained, then available allocation (optimization) schemes might be applied effectively.

Goals will often be best expressed as constraints, rather than as crisp state descriptions. Tradeoffs between goal components will have to be specified, and an objective function created. This objective function must be used by the planner to guide its processing. The notion of an objective function implies an optimization approach; this is appropriate in real-world planning, since most time will be spent trying to evaluate tradeoffs while attempting to accomplish our goals.

6.4.3 Distributed AI

6.4.3.1 Background

Human problem-solving processes frequently require cooperation among a number of individuals with specialized capabilities [12]. While this approach may be enormously more complicated than methods for individual problem-solving, it is necessary in many cases due to the normal distribution of functions and capabilities among individuals. Distributed AI work represents a shift in paradigm from a focus upon the individual problem-solver with full control over its local information and knowledge, as well as sole responsibility for the solution to the problem, to a view of a distributed collection of assets that cooperatively accomplish a given function [13]. Such a paradigm shift often leads to simpler solutions to the problems that motivated the shift, while initially resulting in problems of significantly greater apparent complexity. Such is the case in distributed AI: the approach appears to require much more complex approaches to problem solutions than would be required by a centralized problem solution; however, the ultimate solutions may well be significantly simpler when developed in a distributed architecture. For example, a distributed AI system would not require strict concurrency among knowledge bases. Cooperative problem-solving requires an effective communication capability, which must include protocols to manage the communications, methods to maintain distributed knowledge bases to an appropriate level of consistency, and a means to estimate both what another entity knows (believes) and what his capabilities are.

6.4.3.2 Role of Distributed AI in SDI BM/C^3

Since SDI will consist of an array of distributed weapons and sensors with functionality distinct capabilities, the management of interactions to achieve shared goals will be of critical importance. Data, knowledge, and capabilities will be distributed geographically around the BM/C^3 system; potentially scarce (and possibly ephemeral) communication resources will be required to interconnect the distributed components. Managing such a distributed system and taking advantage of specialized properties of nodes in the system in a cooperative, problem-solving manner is a challenge to AI researchers (to say the least).

6.4.3.3 The State-of-the-Art of Distributed AI

Most work in cooperative problem-solving has focused either on extremely basic issues in representation of knowledge and beliefs or on networks of similar devices such as sensors or switches. No useful applications have been developed.

6.4.3.4 Recommendations

My only recommendation here is that the distributed nature of the problem needs to be emphasized from the beginning of any research program addressing SDI related issues. Any implementations should enforce a distributed architecture for situation assessment and planning.

6.4.4 Natural Language and Speech

6.4.4.1 Background

Natural language research is similar to AI generally in that there is both a scientific underpinning oriented toward understanding human language skills, and an engineering one aimed at the development of useful language understanding systems. Speech research might be thought of as the real-world adjunct of natural language, in that it aims to understand utterances in noisy environments, spoken by unknown speakers (possibly subject to varying levels of stress), that are frequently ungrammatical. Most researchers in these fields are aiming at the goal of enabling a computer to understand written and spoken "natural" English [14].

Natural-language understanding includes the ability to cope with intrinsically ambiguous grammatical statements and pronoun references as well as to handle nongrammatical constructions, misspellings and mispronunciations, slang, and many other idiosyncrasies and pecularities. The reason that people are generally insensitive to the numerous possibilities for misinterpreting most natural language statements or to the inability to assign any interpretation at all has several bases:

- People communicating generally share a great deal of common knowledge--communication, then, consists of information exchanged to bring the knowledge bases (KBs) of the communicators into some general agreement ("registering" the KBs), and information designed to effect an incremental change to the other person's KB, or information (queries) designed to elicit information about the other person's KB.
- There is a tremendous amount of redundancy inherent in any utterance (written or oral) in the context of the communicators' KBs. This redundancy enables the reduction or elimination of the "intrinsic ambiguity" in a statement.
- People have the ability to request confirmation of an understanding of another person's statements. That is, most communication is a dialog, where questions may be asked and clarification received. While this enables relatively low instantaneous-bandwidth communication, it increases the total bandwidth required and extends the time over which communications must be maintained.

6.4.4.2 Role for Natural Language and Speech Processing in SDI BM/C^3

The apparent utility of natural language for interfacing humans to such things as an BM/C^3 system is typically centered on two points: first that the users can communicate with the system in natural language, which is what they are used to, and that, second, the required bandwidth for communication can be reduced by taking advantage of the receiver's ability to disambiguate and understand the utterance. The second advantage really relies on the implicit assumption of large, comprehensive, shared KBs (which would be advantageous for other reasons), and is independent of natural language per se. With respect to the first point, historically when it has been critical that communications be brief, secure, and well understood by the intended recipient (e.g., battlefield communications between a soldier and his commander--two true natural language processors!), they have opted for a reduced, tightly constrained, highly formatted, unnatural language, where ambiguity is limited, and very few (if any) incorrect interpretations are derivable from an utterance. I think this is the path that SDI should follow. This "unnatural language" approach will reduce the impact that natural language work will have on BM/C^3, but will have less effect on the work on speech as researchers will still have to deal with problems of noisy environments, speaker independence, etc.

The human/machine interfaces will always be more comfortable and perhaps reliable with communication at a cognitive level but in a very tight real-time environment it appears difficult to avoid the low level interactions where the common knowledge is so strong that only very little needs to be said. In fact, the whole role of man in the execution phase of this system raises serious questions. Paradoxically, there is not time enough for him to interact effectively, yet it cannot (should not) be executed without him. Natural language work appears useful only in the development stage or in the long term monitoring and situation assessment (standby) stage.

6.4.4.3 Criticality to SDI

While natural language interaction may well be important during development, test, and maintenance of the SDI system, I do not think that either natural langauge or speech understanding will be critical to the operational battle-management functions of SDI.

6.4.4.4 Recommendations

As it is likely that neither natural language understanding or speech processing per se will play a significant role in an SDI engagement, I am not recommending that they be supported under the SDI BM/C3 program. However, the relevance of these areas to the development, maintenance, upgrade, and general day-to-day operations for the battle management system is clearer, and if these functions should be included within the program, then support for natural language should be reevaluated. Furthermore, the inferential and planning techniques studied as a component of natural language research are germane to SDI BM/C^3, and are addressed elsewhere in this report.

6.5 SUMMARY AND RECOMMENDATIONS

Generally speaking, the utility of applying AI to BM/C^3 is a function of the quality of both the system's sensor resources and its defensive weapons as well as the system's architecture. To the extent that sensors are able to accurately detect targets and discriminate those RVs, and that the weapons are capable of reliably destroying those RVs, the battle-managment problem could revert to the more traditional operations research task of determining an optimal (or acceptable) allocation of weapons to RVs. This picture is complicated by the fact that any attack would likely be structured and, therefore, any allocation scheme would require the ability to phase the use of its resources, and therefore some ability to anticipate future needs. Furthermore, it is highly unlikely that SDI sensor and weapon resources would be effective and dependable enough to base weapon release decisions on the raw sensor data. This, then, implies a requirement to draw conclusions and inferences from a collection of multisource data and prior knowledge. In addition, the need to provide a flexible response to situations that cannot be well characterized in advance implies a need for a planning/replanning capability.

Similarly, the choice of system architecture will have a major impact on the requirements for reasoning, planning, and communication, and the appropriate distribution of functionality. The more controlled and hierarchical the SDI battle management function, the greater the stress on our abilities to plan and reason. The more distributed functions are, the greater the requirement for a completely new approach.

To the extent that these capabilities are needed, the only technology that addresses the development of techniques for inferencing and planning is AI. Unfortunately, AI is not in very good shape for application to SDI at this time; it is still fairly immature as a discipline, with relatively few applications outside the laboratory. Worse, very little work has been accomplished to provide effective support for the types of inferencing and planning that must be accomplished to solve SDI problems. This situation is exacerbated by the relatively small population of competent AI researchers.

My recommendations are fairly general. I think to support an application of the magnitude of SDI, a population of "AI engineers" will have to be created. As opposed to "knowledge engineers," whose role is to transfer knowledge from the brain of the knowledgeable to the computer, the role of the AI engineer will be to transfer technology from the basic research laboratories into real applications. They will need to have strong theoretical backgrounds in AI and related fields, but they will be required to develop applications that are able to deal with the "inelegancies" of the real world, such as errors, noise, malfunctions, time, and beliefs.

A way to begin creating this population would be to develop a set of scaled-down SDI-like problems, perhaps based on networks of heterogeneous sensors and effectors, whose solutions would stress critical capabilities. These problems must require performance-oriented approaches and solutions, but should not be so focused that shortcut solutions that avoid the difficult issues could suffice--the goal of the work must be technology development. These problems should require the use of real sensors and advanced computational hardware and architectures. They should be time-stressed, and the elements of the problem should emphasize the distributed aspects of the overall system. An ideal set of problems would present a progression in solution requirements beginning with the current state-of-the-art, and ending with technology that supports the SDI requirements.

The approach to advancing the state-of-the-art in ways likely to produce technology applicable to SDI problems advocated here is patterned after the DARPA Strategic Computing Program. In particular, a set of candidate, scaled-down problems with the key characteristics of the SDI BM/C^3 tasks should be defined. The set should consist of problems of graduated levels of difficulty, starting with tasks just beyond the current state-of-the-art, and gradually increasing in difficulty until problems of the same order-of-magnitude as the actual SDI problem can be handled. These problems should stress researchers in a number of realistic dimensions. The problems should require the development of planning and reasoning systems that are distributed over some type of communication network, and that are able to access real sensors and effectors connected to the network. The problems should include aspects of resource limitations, distributed resources, time-criticality for operations, hostile interactions, and uncertainty at a variety of levels. The key aspect of the problem set should be that it enforces realism in the solution; it should not be possible to find a special-purpose solution by simplifying the problem.

An example of such a problem set might focus on the task of monitoring the flow of vehicles through a transportation grid composed of internetted sensors of various types [13]. Parameters that could be varied to provide a set of problems include the number, spacing, and topology of the network, the types and quality of the sensors, the bandwidth and reliability of the communication network, and the distribution of speeds of the vehicles. The problem could be extended in other dimensions to drive different areas of research. For example, the monitoring system could be charged with regulating and controlling the flow through the network (perhaps by using local sensors to issue instructions to individual vehicles, which may then be monitored as they pass through the rest of the network), in an effort to achieve minimal throughputs, even in situations of high loads.

Initial work on a system of this sort should involve simulation, with the simulated component gradually being replaced by actual hardware. Such a system would be relatively capital-intensive, and is, therefore, a logical candidate for inclusion in one or more testbeds that would be made available to the research and development community.

In addition, basic research programs in AI must be strengthened. A common phenomenon when a basic research area begins to mature and show signs of applicability is that a significant segment of the capable basic researchers respond to the financial lure of application development and are lost to the research community. Furthermore, funding support for basic research tends to decrease in favor of support for applied efforts. If AI is to have any hope of utility for SDI, it is critical that the basic research programs advance the state-of-the-art well beyond its current level. This means that basic research programs must be encouraged and strengthened.

Both basic and advanced research programs must stress the following real-world issues:

- The development of effective representations for processes, time, and belief
- Development of methodologies for problem decomposition
- The use of large knowledge bases
- Distributed problem-solving
- The use of specialized computational hardware architectures.

It is difficult to prioritize these issues, but the need for specialized computer hardware is a common requirement. I would recommend accelerating the DARPA SCP computer hardware program.

Many problems posed by current SDI thinking appear to require the types of techniques being developed by AI researchers. Unfortunately, it will require significant money and time to advance the state-of-the-art in hardware and software sufficiently to capitalize on this technology in the area of BM/C^3.

We have outlined certain of the difficulties as well as recommendations for change in this section.

6.6 ACKNOWLEDGEMENTS

The Strategic Defense Initiative truly represents one of the most severe challenges ever proposed to the technical community. In addition, SDI has serious political and economic implications that are understood poorly (at best) at this time. While these latter issues are admittedly extremely important, in this section I have attempted to focus tightly on just the technical issues involved with the possible application of AI to SDI battle management. Many colleagues here at SRI helped keep me focused on the critical issues, and I would like to thank them for their careful reading of the section. Special thanks are due to Don Nielson, Len Wesley and Jane Robinson for their advice, suggestions, and constructive criticism; I have tried to incorporate their suggetions where possible, and I think the section has benefited. Equal thanks must go to Joani Ichiki who assisted in the editing and presentation, and in transferring changes between manuscripts prepared on two different word-processing systems.

In addition, working with the other members of this panel has been stimulating, educational, and enriching, and I thank them for the experience. I would especially like to thank Doyle Thomas of the US Army Strategic Defense Command and Bill McDonald and Mack Cooper of SDC Huntsville for organizing and (to the extent possible) moderating the interactions of a collection of diverse viewpoints.

6.7 REFERENCES

1. J.C. Kunz, T.P. Kehler, and M.D. Williams, "Applications Development Using a Hybrid AI Development System," AI Magazine, 5(3) (1984).

2. C. Eorgy and J. McDermott, "OPS: A Domain-Independent Production System Language," Proceedings of the Fifth International Joint Conference on AI, pp. 933-939, 1977.

3. F. Hayes-Roth, D.B. Lenat, and D.A. Waterman, eds., Building Expert Systems, Addision-Wesley Publishing Company, Reading, MA, 1983.

4. DARPA, "Strategic Computing: New Generation Computing Technology: A Strategic Plan for its Development and Application to Critical Problems in Defense," 1983.

5. J.A. Allen, Logic: Form and Function, Edinburgh University Press, Edinburgh, 1979.

6. W.F. Clocksin and C.S. Mellish, Programming in Prolog, Springer-Verlag, New York, NY, 1981.

7. P. Harmon and D. King, Expert Systems, Wiley Press, New York, NY, 1985.

8. P. Bonnisone and R.M. Tong, "Editorial: Reasoning with Uncertainty in Expert Systems," International Journal of Man-Machine Studies, 22(3), 1985.

9. J.D. Lowrance and T.D. Garvey, "Evidential Reasoning: A Developing Concept," Proceedings of the International Conference on Cybernetics and Society, Seattle, WA, 1982.

10. J.M. Tenenbaum and H.G. Barrow, "Experiments in Interpretation-Guided Segmentation," Artificial Intelligence, V.B., pp. 241-274, 1976.

11. T.D. Garvey, "Issues and Approaches to Planning Under Uncertainty," Final Report for Office of Naval Research, Contract No. N00014-81-C-0115, SRI International, Menlo Park, CA, 1984.

12. R.G. Smith and R. Davis, "Frameworks for Cooperation in Distributed Problem Solving," IEEE Trans. on Systems, Man, and Cybernetics, SMC-11(1), pp.61-70 (1981).

13. V.R. Lesser and D.D. Corkill, "The Distributed Vehicle Monitoring Testbed," AI Magazine, 4(3), 1983.

14. C.R. Perrault and B.J. Grosz, "Natural-Language Interfaces," SRI Tech Note, Artificial Intelligence Center, SRI International, Menlo Park, CA, 1985.

SECTION 7 - CONTROL THEORY

Robert Bass
Inventek Enterprises, Inc.

Control theory is a special case of System Theory in which the basic system modeling is assumed and in which the problem definition concerns the selection (including optimization and validation) of the control policy, i.e., the selection of the control vector u. If u = u(t) is specified as a function of time t one has an open loop control law. If u = u(t,x) is specified as a function of time t and of the state vector x of the process, then one has a feedback control law. The objective of selection of the control law is to affect the future evolution in time of the state of the process, such as to drive the state toward a predefined set point or terminal state.

The above definition refers to a fixed structure, centralized feedback control system. For the state of the art, refer to Figure 7-1. The problem has essentially been solved for arbitrary nonlinear processes, nonlinear sensor kinematics and nonlinear actuator kinematics, and stochastic process disturbances and sensor noises, by the Mortensen Separation Principle that rigorously decouples the problem into two separate problems: state estimation, and control law optimization. The former is solved by the DMZ equation (Duncan-Mortensen-Zakai equation) while the latter is solved by Dynamic Programming in function space by means of the MBHJ equation (Mortensen-Bellman-Hamilton-Jacobi equation). (Literature references may be found in [Bass, 85] which surveys and summarizes some 84 technical papers on centralized control theory.)

Within the conventional framework of centralized control, certain possibilities of generalization are apparent and desirable. These include:

- Hybrid State Spaces
- Hybrid Time Domains

P = PROCESS DYNAMICS

S = SENSOR KINEMATICS

A = ACTUATOR KINEMATICS

C = COMPUTER

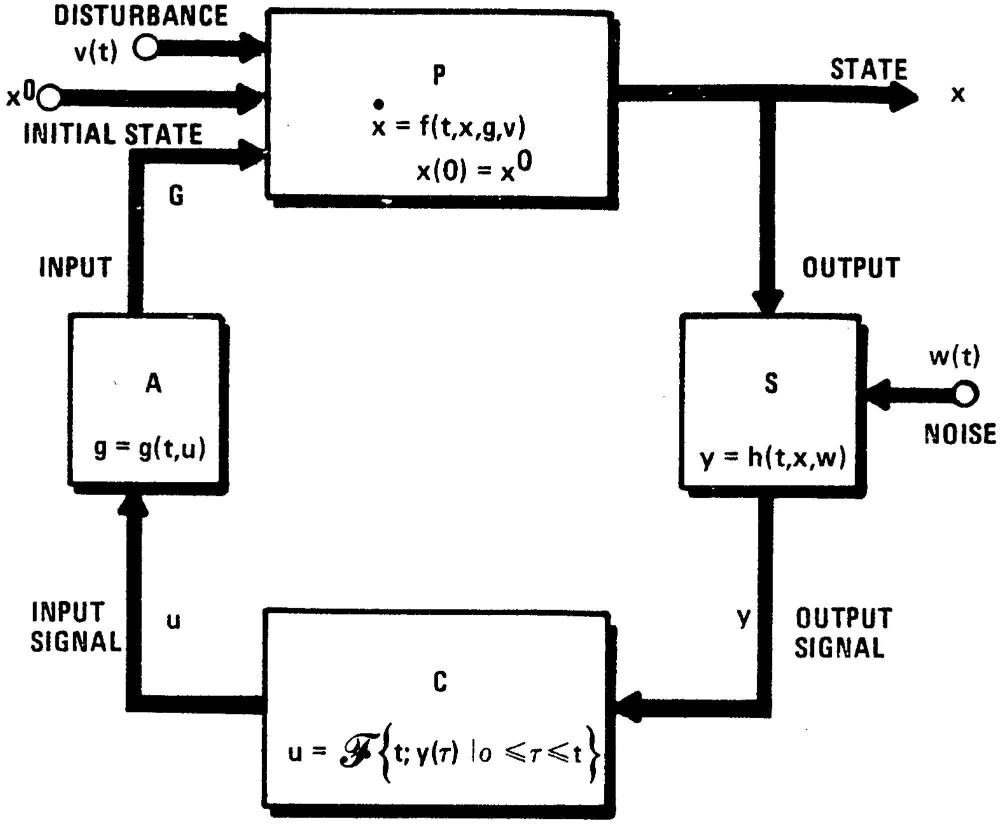


Figure 7-1. Fixed-Structure Centralized Control

- Hybrid Control Spaces and Policies
- Very High-Dimensional State Spaces and Multimodeling
- Control Algorithm Types: Parameter-Adaptivity Explicit or Implicit
- Control Processor Architectures
- Enhanced System Qualities:
  - Stability
  - Robustness
  - Producibility/Expandability
  - Reliability/Maintainability.

Current research trends in centralized control will continue to provide more and more generalized frameworks for treatment of these issues and so they are not of primary importance for the present study.

What is different about the BM/C^3 problem in a radical way is the necessity (for many overlapping reasons) to allow for the utilization of distributed-processing architectures and the over-riding importance of fault tolerance, graceful degradation in the presence of individual element failure or loss, and total security of certain aspects of information flow.

Another difference with established control theory is that the state variables, observed variables, and control variables are not necessarily physical quantities, but are what may be called management variables. In fact, for present purposes we may define management controls as consisting of those real-time decisions, choices, enablements and authorizations associated with real-time allocation of management-type resources. These include:

- Access to communication links, computers, files sensors, etc.
- Enablements of algorithms, data replication, data storage processes, communication alternatives, etc.
- Authorizations of substitutions, modifications, etc.

Even this augmentation of physical-type variables by econometric-type variables is not the most difficult generalization required for progress in the present context, however.

A candidate architecture for an adaptive-structure, distributed-element, decentralized control formulation of the problem is presented in Figure 7-2. The key issues are now:

- Producibility/Expandability
- Reliability/Maintainability
- Security.

7.1 ISSUES

7.1.1 Producibility/Expandability

State of the art control technology can handle at most a few hundred state variables. Certain studies at IBM of high-dimensional matrix operations suggest that for the foreseeable future the inherent limitations of Numerical Analysis problems such as Round-Off Error and Truncation Error and Convergence Time will limit control theory to at most a thousand state variables.

However, if the SDI problem is formulated as a completely coupled dynamical process, the numbers of state variables will be in the tens of thousands or more. While abstract studies independent of dimensionality doubtless yield valuable insights into system architecture, the implementation of an actual system is clearly infeasible unless the dimensionality of the problem is somehow reduced. Therefore producibility requires some sort of decoupling. Similarly, decoupling the problem would enhance expandability.

7.1.2 Reliability/Maintainability

Systems including as elements hundreds or thousands of different sensor suites, actuator suites, and computer networks cannot be designed on the hypothesis that all elements will be "up" at all times. Moreover, in the SDI context, it must be assumed that elements will not only fail spontaneously but that some elements will be suddenly lost as a result of enemy action. Therefore, it is absolutely essential that the system be able to reconfigure itself in

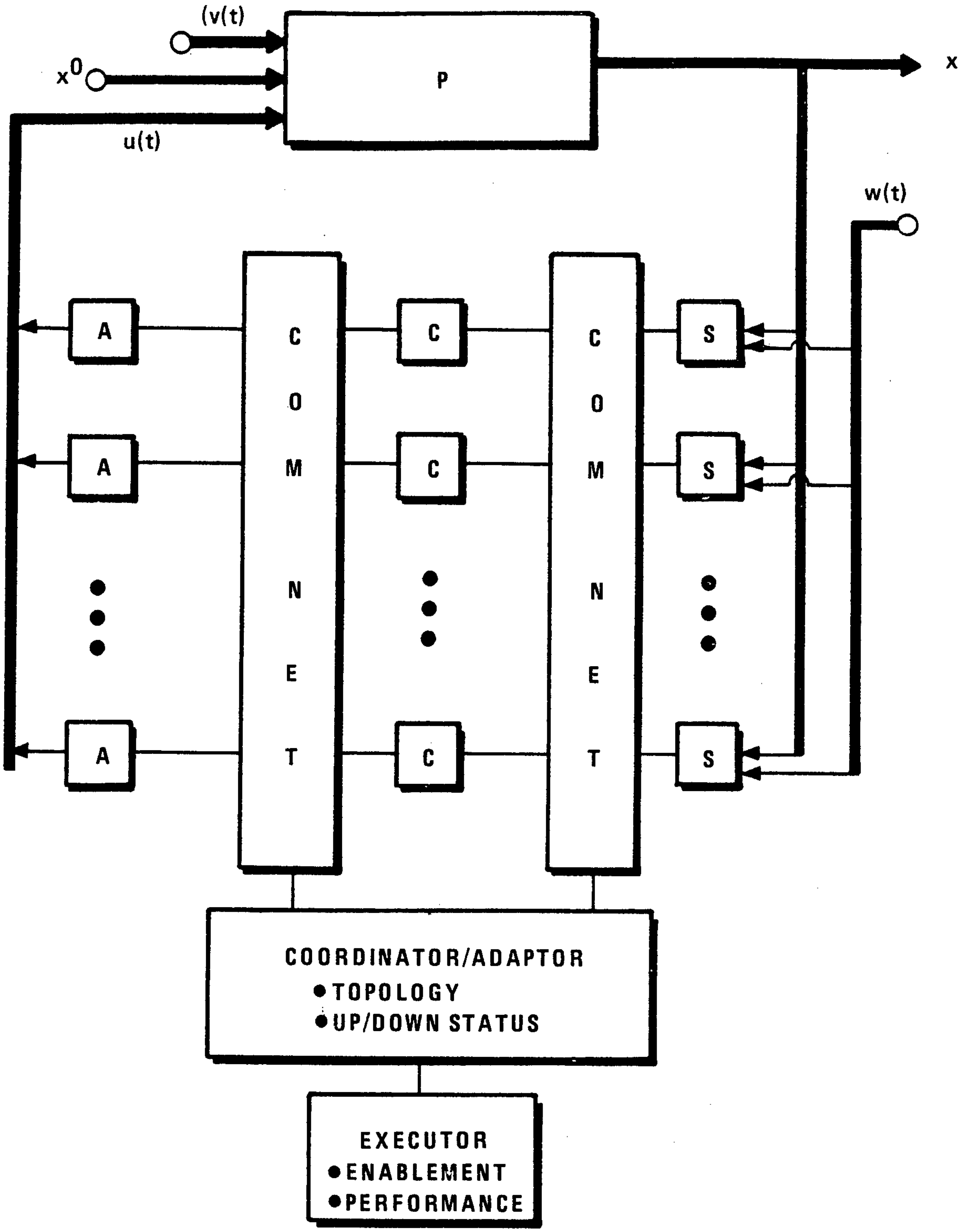


Figure 7-2. Candidate Adaptive-Structure Decentralized Control

real time, and that the performance of the reconfigured system degrade only gradually as more and more elements are lost. This can be achieved in a brute-force manner by hardware redundancy, but considerations of cost-effectiveness indicate that a preferable approach would be some kind of analytic redundancy involving adaptive architecture and including reconfigurable network topology. Also maintainability would be enhanced if the temporary absence of one or a few elements would only gracefully degrade overall performance.

7.1.3 Security

In early December 1985, the Presidential Science Advisor disclosed publicly that he had discussed within the White House the possibility of seeking cooperation of the USSR in mutually scheduled deployment of SDI systems by an offer from the USA to share with the USSR some key and absolutely vital part of the system, such as the software and C^3 aspect of system enablement. This highlights the extreme importance of security of the executive function of the total system. The record of amateur hackers (much less professional computer-criminals and saboteurs) of breaking into banks and defense networks points out the unresolved nature of this problem.

7.2 CANDIDATE APPROACHES

7.2.1 Ad Hoc Engineering

This is the present state of the art. No systematic methodology for handling the preceding three issues has been uncovered during the present study.

As already mentioned, many features of the BM/C^3 problem require generalizations of standard centralized control theory which are already being researched outside the SDI context.

7.2.1.1 Hybrid State Spaces

In addition to the Euclidean spaces typical of electromechanical state variables, one wishes to include discrete state spaces consisting of a finite or countable number of parameter-states, such as whether a given component is functional (up) or dysfunctional (down). The final state space would then consist of a Cartesian product of both continual and discrete sets.

7.2.1.2 Hybrid Time Domains

Typically electromechanical kinematics and dynamics is modeled as occurring in continuous time, whereas measurements or observations may be made in either continuous time (by analog instruments) or at discrete times (as in sampled data systems), but are usually converted to discrete times (by A/D converters) for digital processing.

7.2.1.3 Hybrid Control Spaces and Policies

As already mentioned, there is a distinction between control laws of the type called open (e.g., fire and forget) and closed loop (e.g., shoot--look--shoot). There is also a distinction between continuous control laws (which can be approximated discretely by pulse-width-modulation schemes) and inherently discrete control laws (such as switching controls, as in on-off and bang-bang controls). A more subtle kind of discrete control signal consists of enablement or authorization controls.

7.2.1.4 High-Dimensional State Spaces and Multimodeling

The very high-dimensionality of state spaces already alluded to can be approached in various ways. In multimodeling one introduces a set of overlapping problems of tractable dimension, in each of which certain peripheral aspects have been simplified or assumed already solved. Alternatively, given an exact, but high-dimensional formulation, one can seek to decompose it into a convergent sequence of tractable problems (as in [Varaiyun 69]), or one can use a method such as Kron's method of tearing to decompose the problem into a collection of tractable problems. (For a survey of 55 papers related to tearing, see [Harrison 69].) Still another alternative is to go completely to the limit of infinite dimensionality, in which at least structural results that illuminate generic possibilities can be derived analytically. Two leading approaches in the infinite-dimensional case proceed via functional analysis (as in [Balakrishnan 81]) and via algebraic or topological semi-group theory (as in [Curtain 78]).

7.2.1.5 Control Algorithms: Adaptivity Explicit or Implicit

As already emphasized, the biggest new problem is that caused by the necessity of considering decentralized control as opposed to classical centralized control. For a survey of 156 papers on large-scale or decentralized control, see [Sandell 78]; for a survey of 25 key such papers, see [Athans 78A]; and for a summarizing editorial see [Athans 78B]. An additional complication concerns multi-tier systems, as in the SDI multi-layer proposals, which is called hierarchical control. Also the necessity for fault-tolerance or self-healing characteristics leads one to consider the field of adaptive control. As mentioned further below, what is needed is robustness, or insensitivity to unknown or uncontrollable parameter variations. Feedback control is preferable to open-loop control as it introduces at least some degree of passive robustness. A systematic method of maximizing intrinsic or passive robustness within the widely-accepted LQG framework has been presented in [Bass 83]. Furthermore, by raising the transfer-function order of the controller, one may introduce a passive hyper-robustness as in [Johnson 86]. One may also consider explicitly-adaptive control, in which the parameter variations are somehow identified (either explicitly or implicitly), and the resulting information used to adjust control gains or their equivalent; in this case, one cannot avoid dealing with problems which are both time-varying and nonlinear. Also, learning and discrimination algorithms, as in AI., may be considered.

7.2.1.6 Control Processor Architectures

By now the suggestion that computer architectures may be made more effective by designing them to be algorithm-dependent has been widely accepted. For example, Integrated Systems, Inc., of Palo Alto, has published proposals to develop multi-processor, parallel-logic architectures to fit certain problems whose optimal solution consists of a parallel bank of Kalman filters with adaptively-weighted output fusion.

7.2.1.7 Enhanced System Qualities

A large fraction of conventional centralized control research concerns the enhancement of such system qualities as stability, robustness, producibility/

expandability, and reliability/maintainability, all of which are of course also needed in the SDI context. In the present instance, it may be well to recall the distinction between electromechanical structural stability or [ordinary] robustness, and information-flow robustness. The latter would involve both communication robustness and computational robustness (including numerical stability and graceful degradation as arithmetic precision is reduced).

Additional details regarding approaches to decomposition by aggregation (both spatial and temporal, as well as functional) may be found in the section on general System Theory above.

Also, the present study should be regarded as a continuation of [BMD 78], which included references to 77 papers pertaining to the problem at hand, including applications of both ordinary and singular perturbation theory and multi-time scale techniques. These valuable prior efforts will be incorporated herein by reference rather than by being listed again.

In illustration of the fact that only ad hoc applicability of the many known techniques is available, consider an important example, namely the weapon-target allocation problem. Reasons will be presented for regarding this as an example of a stochastic, dynamic, distributed, hybrid-state, optimal control problem.

The problem is stochastic because it involves surveillance system residual uncertainty (including problems of both target location and target identity [discrimination between decoys and boosters or RVs], as well as prediction uncertainty (including the problem of target impact zone prediction and target trajectory characteristics identification), and engagement uncertainty (including interceptor reliability and probability of kill).

The problem is dynamic because it involves target/weapon-platform motion, as well as launch of new targets, MIRVing, and use-destruction of certain weapons.

The problem is *distributed* because it involves BM survivability (which implies the necessity of multiple platforms) as well as BM/weapon-system functionality and security.

Finally, the problem involves *hybrid states* because it involves both discrete decisions (such as weapon-target pairing) and continuous decisions (such as time of fire).

An approach to solution has been suggested which postulates the applicability of stochastic optimal control theory, hierarchical control theory, and team theory.

The use of *stochastic optimal control* theory is postulated to include the development of a centralized solution but with the calculations distributed among the different battle managers; this would require the development of computational approaches and protocols/algorithms to ensure that the calculations can be carried out in an environment of failing communications.

The use of *hierarchical control* theory is postulated to involved coordination mechanisms to satisfy prespecified constraints, and individual objective functions to be optimized.

The use of *team theory* is postulated to include both decentralized solutions (by the many techniques referenced above) and multimodeling (which, as already suggested, involves simplified representations of problems other than the one being focused on, and therefore implies the necessity of the coordination already alluded to).

The foregoing outline of a possible applied research program aimed at the weapon-target allocation problem shows that much new theoretical study and many *ad hoc* engineering choices would have to be made in order to treat this example by known approaches.

In this connection, it is relevant to quote the conclusion of [Sandell 78]: "Our most fundamental conclusion, after surveying a vast amount of literature, is that although considerable progress has been made in many directions, the questions of what structures are desirable for control of large scale systems has not been addressed in a truly scientific fashion. In our opinion, we do not believe that the existing mathematical tools ... are powerful enough to define a preferable structure for decentralized and/or hierarchical control. Likewise, [Athans 78A] concluded: "Thus, we desperately need innovative approaches on how to select a good structure ... . future relevant theoretical directions for research ... must contain novel and nontraditional philosophical approaches."

7.2.2 Conjectured Applicability of FSD (Functionally Structured Distribution)

This is an intuitively discovered but as yet unproved method that claims to address all three of the preceding issues simultaneously. (Refer to appendix at the end of this chapter for a preliminary heuristic "proof" of the optimality of FSD.)

An initial perusal of the published article [Billings 82] on FSD leaves the false impression that what is claimed is an entity. Actually, what is conjectured is a universal methodology for approaching any distributed processing problem. The approach includes two fundamental architectural elements or building blocks, together with four rules governing allowable interconnections between the blocks.

The methodology is reminiscent of two well-know fundamental structural results. In the theory of Structured Programming, it is a proved theorem that all possible computer programs can be composed of sequences of just three basic structures: branching, looping, and sequentially continuing. In the theory of Digital Electronic Circuits, it is a proved theorem that all possible circuits can be composed of just three fundamental elements: and-gates, or-gates, and not-gates. (Both of the preceding results are corollaries of known results in Boolean Algebra.)

What the discoverer of FSD is claiming (without a completely rigorous proof as yet), or, from the scientific point of view, is conjecturing, is that his set of architectural building blocks and interconnection rules is sufficient for the architecture of any possible distributed-processing system and in all cases analyzed to date provides optimal results regarding the key issues mentioned in the preceding section. More investigation is needed in order to refute or confirm this conjecture. In particular, alternative concepts, suitable for comparative purposes, need to be identified, and quantitative comparative criteria, as called for by Athans, need to be proposed.

As an example of the possible applicability of the FSD concept, a preliminary attempt to apply the principles of FSD to a BM/C^3 problem is illustrated in Figures 7-3 through 7-7.

The problem chosen for this example is the post-interceptor-launch midcourse-guidance problem, which is selected because it has a natural decoupling which facilitates the application of the FSD concept.

The FSD concept assumes that the problem will be solved by the use of only two generic elements, called C elements and D/C elements, together with four rigid rules of interconnection of the elements.

The two basic architectural elements or building blocks are as follows. The C elements are programmable computers. The D/C elements are non-programmable data centers manufactured with a firmware ACP (Access Control Program).

The four rules of interconnection are as follows:

1. Any C may contact any D/C and exchange information if security restrictions are met.
2. No C may ever contact any other C directly.

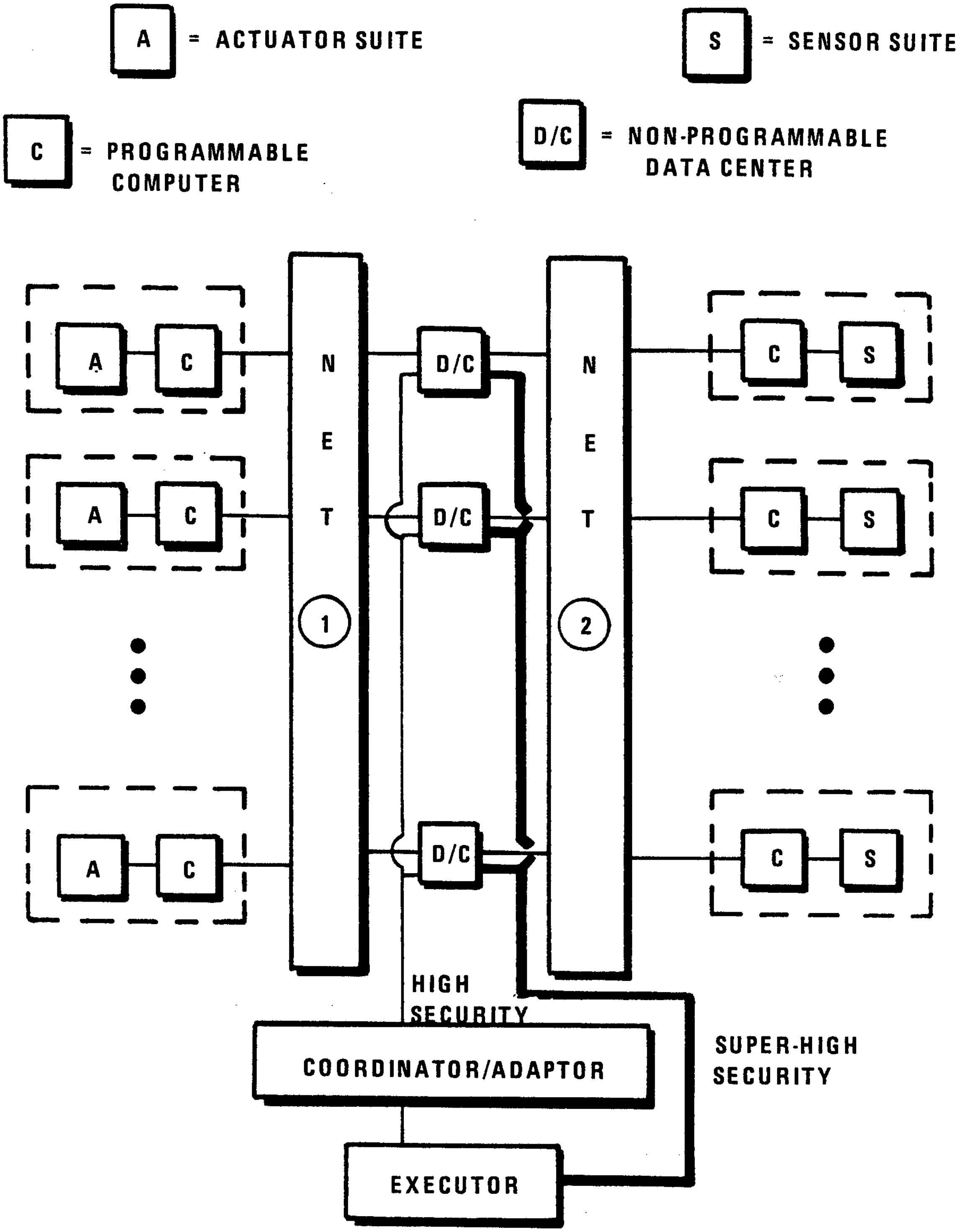


Figure 7-3. Fault-Tolerant/Secure FSD Control

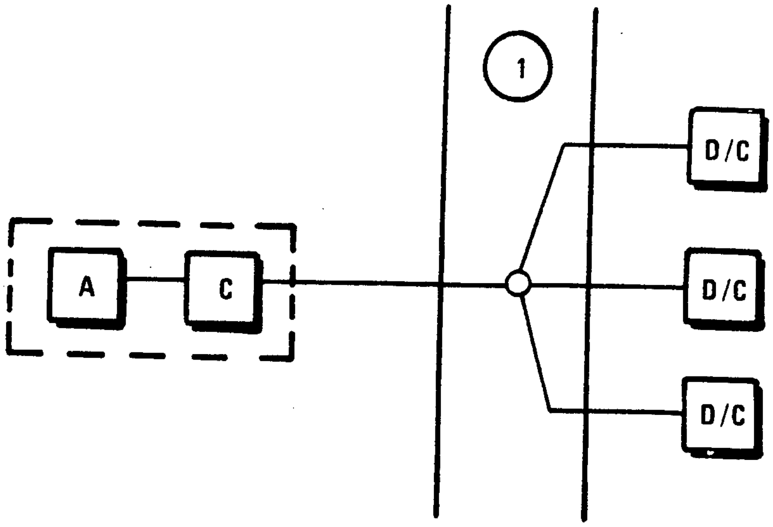


Figure 7-4. First Net Topology Configuration

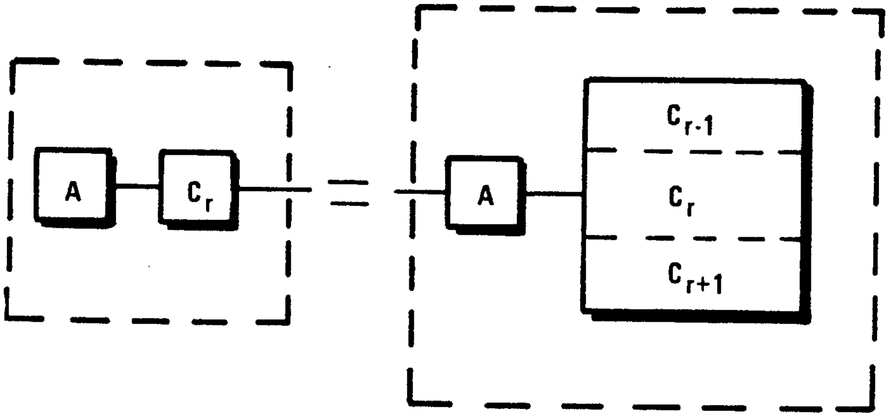


Figure 7-5. Adjacent Level Algorithm Redundancy

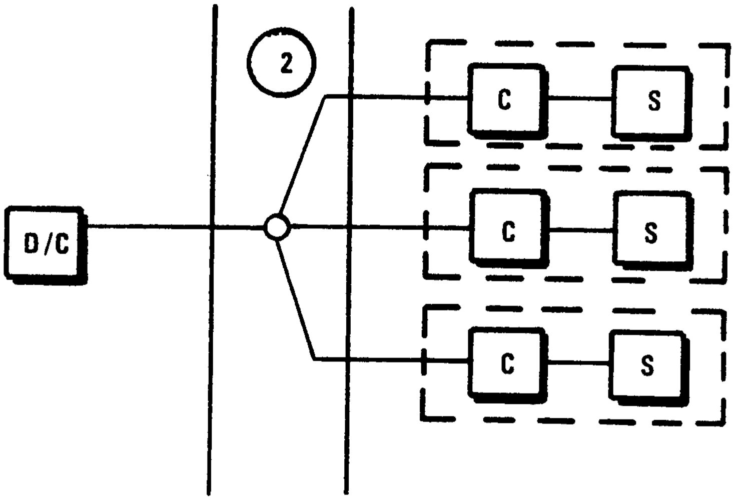
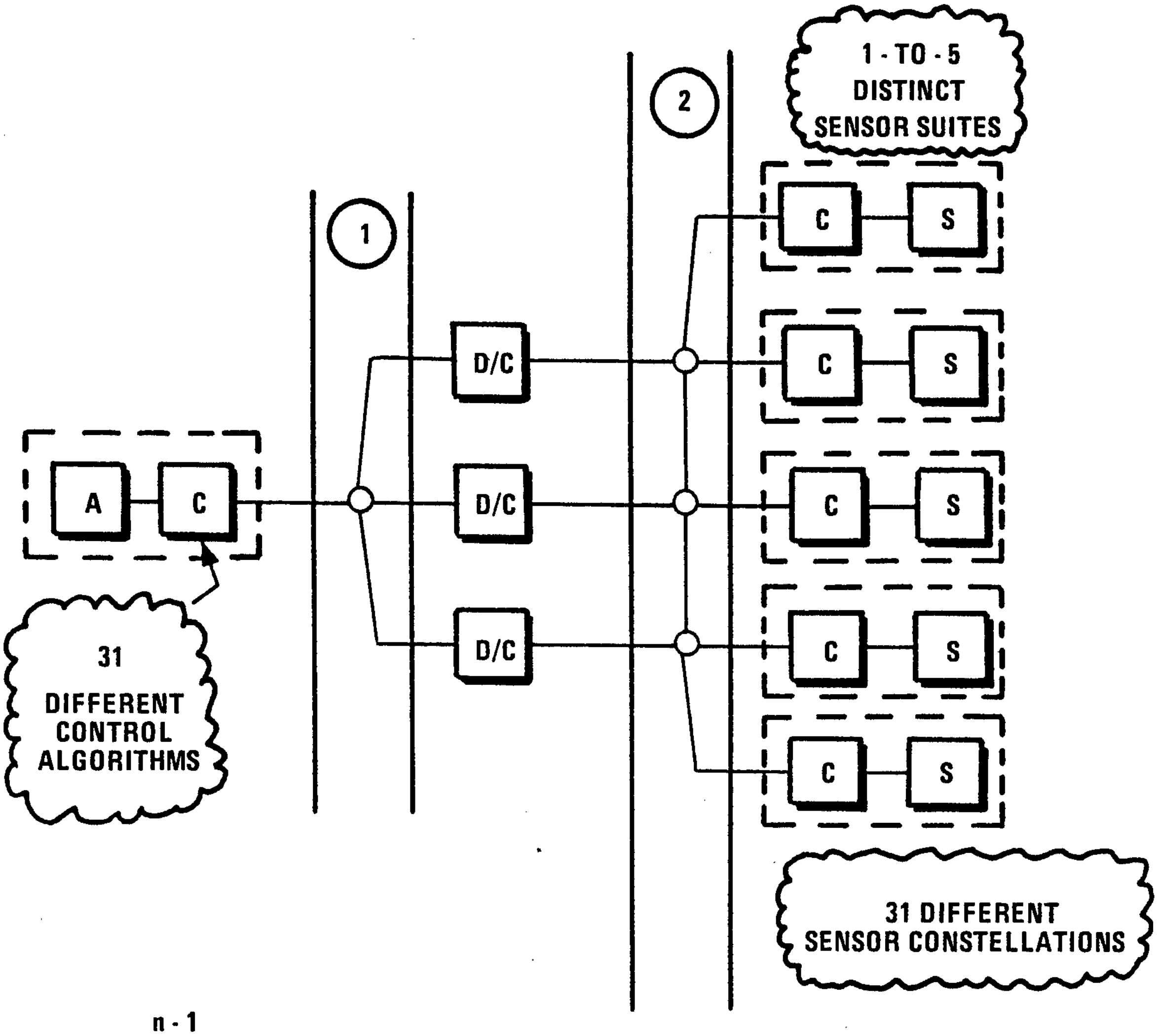


Figure 7-6. Second Net Topology Configuration

1 - TO - N SENSOR SUITES DOWN

⇒

2^n - 1 DIFFERENT SENSOR CONSTELLATIONS



$$\sum_{k=0}^{n-1} \binom{n}{k} = 2^n - 1$$

$n = 5 \Rightarrow$

$2^n - 1 = 2^5 - 1 = 31$

Figure 7-7. Control Algorithm Composed of 93 Tractable Solutions in Parallel

3. No D/C may ever transfer information directly with any other D/C.
4. No D/C may ever initiate a contact with a C.

Clearly, the FSD concept can be demonstrated in such commercially available environments as the DEC Cluster, but it is a more specific architecture than any uncovered by the present survey.

Recall now Figure 7-1 and its replacement by Figure 7-2. The next step is to apply the FSD concept and replace Figure 7-2 by Figure 7-3. The further details provided in Figures 7-3 through 7-7 illustrate how the system is to be made reliable and fault-tolerant.

The system naturally decouples at the A elements. In Figure 7-4 it is shown that the topology of communication network 1 may be so designed that the A element will receive the required information even though any one or any two of the three most adjacent D/C elements fails. (Below it will show why these three elements each contain adequate information.)

In Figure 7-5 it is shown how the element C associated with the A element at a given level is always simultaneously computing the control algorithms associated with the adjacent upper and lower levels as well as its own level. This is triple analytic redundancy.

In Figure 7-6 it is shown how the topology of communication network 2 can be so configured that each D/C element simultaneously receives data from the three nearest levels of S elements. Thus the D/C has adequate information even if any one or any two of the adjacent sensor suites fails (assuming that the target is in range of all three S elements).

Figure 7-7 combines the three preceding figures. There are five distinct sensor suites; assume the target is in range of all five; then if any one, or any two, or any three, or any four such sensor suites is lost, there is still enough information available to provide for the computation of appropriate commands to the illustrated actuator suite. There are now from one to five relevant S elements, so by elementary combinations there are a total of 31 different sensor-suite combinations, or sensor constellations which can be employed, depending upon how many of the S elements are up. Consequently one needs to solve the associated type of centralized control problem (which involves a tractable number of state variables) just 31 times (once for each possible sensor constellation), and then arrange for the C element associated with the A element to simultaneously process all 31 algorithms that would be appropriate for each of the three adjacent D/C elements; i.e., to process 93 control algorithms simultaneously.

Each of these 93 algorithms is relatively small as a computational burden, and so it is practicable to solve all 93 centralized control problems by standard means (such as robustified LQG procedures), and then to program the final C element with real-time programs for execution of all 93 algorithms in parallel. (Therefore the final C element would be well adapted to parallel-logic distributed-processor architectures, as well as to multiplexer techniques.)

Clearly the resultant command-guidance system would remain functional if even one out of the five adjacent Sensor Suites remained functional, and if even one out of the three adjacent Data Centers remained functional. Consequently the resultant system is obviously highly fault-tolerant and moreover will degrade gracefully as some of the Sensor Suites or Data Centers become dysfunctional.

Thus two of the three critical issues (producibility/expandability and reliability/maintainability) appear to have been met in this example of the applicability of FSD. It can also be demonstrated that the issue of security has been met, but space precludes here a deeper discussion of that point, which is addressed further in an appendix on FSD optimality at the conclusion of this chapter.

7.3 RECOMMENDED RESEARCH

Numerous authoritative surveys of the state-of-the-art in Distributed Control by such experts as Athans and Sandell have concluded that progress by known methodologies seems to be approaching its natural limits, and that the situation calls for some radically new ideas or breakthroughs in approaches to the subject.

The conjectured FSD concept pertains to two basic building blocks: C elements [programmable computers] and D/C element [non-programmable Data Centers manufactured with a firmware ACP (Access Control Program)]. There are also four rules of allegedly optimal interconnection of the blocks.

The possible relevance to the post-interceptor-launch midcourse-guidance problem of the candidate architecture displayed in Figures 7-3 through 7-7 suggests that inclusion of FSD in the tool kits already surveyed by Athans and Sandell may provide one example of the kinds of breakthrough for which they have been calling. Other examples, unforeseen at this time but of recognizable relevance to the SDI BM/C^3 when they appear, should be sought. Rather than reliance upon progress in the known approaches to large scale and distributed control process theory to evolve via traditional support channels, open solicitation of innovative approaches stressing relevance to the SDI BM/C^3 problem may provide research proposals of great applicability, including alternatives to the FSD concept which would allow its comparative value to be quantitatively assessed.

7.4 REFERENCES

M. Athans, "Advances and Open Problems on the Control of Large Scale Systems," Plenary Paper, IFAC 1978, Helsinki, Finland (Proceedings, pp. 2371-2382).

M. Athans, Guest Editorial, "On Large-Scale Systems and Decentralized Control," IEEE Trans. Aut. Control, vol. AC-23 (1978), pp. 105-106.

A.V. Balakrishnan, Applied Functional Analysis, Springer-Verlag, 1981.

R.W. Bass, "Robustified LQG Synthesis to Specifications," Proceedings, Fifth Meeting Army Coordinating Group on Modern Control, Dover N.J. (Oct. 26-27, 1983).

R.W. Bass, "Remarks on the Future of Fire Control Theory," Proceedings, Seventh Meeting Army Coordinating Group on Modern Control, Redstone Arsenal AL (Oct. 22-24, 1985).

R.E. Billings & R.J. Ridge, "Distributed Data Processing Using Functionally Structured Distribution," Computer Progress Magazine, Spring 1982, pp. 13-19.

R.F. Curtain & A.J. Pritchard, "A Semigroup Approach to Infinite Dimensional System Theory," Chapter 3 of Gregson 78.

M.J. Gregson (ed.), Recent Theoretical Developments in Control, Academic Press, 1978.

B.K. Harrison, "Large Scale Linear Systems," Chapter 7 of Zadeh 69.

C.D. Johnson, "Adaptive Controller Design Using Disturbance Accommodation Techniques," Int. J. Control, vol. 42 (1985), pp. 193-210; cf. Proc. 1986 ACC.

N.R. Sandell, et al; "Survey of Decentralized Control Methods for Large Scale Systems," IEEE Trans. Aut. Control, vol. AC-23 (1978), pp. 108-128.

P.P. Varaiya, "Decomposition of Large-Scale Systesm," Chapter 12 of Zadeh 69.

L.A. Zadeh and E. Polak, System Theory, McGraw-Hill, 1969.

APPENDIX - PRELIMINARY HEURISTIC "PROOF" OF FSD OPTIMALITY

DEFINITION. By optimality of a processing system we shall mean maximality of the following three characteristics:

1. Producibility/Expandability
2. Reliability/Maintainability
3. Security.

DEFINITION. We shall define a processing system as an FSD system, that is, one based upon Functionally Structured Distribution, if its architecture is based entirely upon the two following architectural elements, interconnected exclusively according to the following four rules of interconnection. The two architectural elements are:

1. C elements (programmable Computer elements);
2. D/C elements (non-programmable Data Center elements, namely database elements manufactured with firmware ACP (Access Control Programs), but otherwise non-programmable).

The four rules of interconection are:

1. Any C may contact any D/C and exchange information if security restrictions are met
2. No C may ever contact any other C directly
3. No D/C may ever transfer information directly with any other D/C
4. No D/C may ever initiate a contact with a C.

Note that the rules imply immediately that all external interchanges of information must be done through C elements. In the case of human input/output, the corresponding C element is often called a User Station. In the case of automated imputs, as from a Sensor Suite, or automated outputs, as to an Actuator Suite, the C element will be referred to as a C element.

CONJECTURE (R. Billings, 1982). The NASC (Necessary And Sufficient Conditions) for a processing system to be optimal is that it should be an FSD system.

PRELIMINARY HEURISTIC "PROOF" OF CONJECTURE. The heuristic demonstration of sufficiency will consist of exhibition of a functional system which as a practical matter seems to be optimal (non-improvable) with respect to the specified performance criteria, and which consists of many hundreds (eventually thousands) of C elements and tens (eventually hundreds) of D/C elements. Such a system is the commercial "Checkrite" system of Checkrite Corporation, Denver, Colorado.

Accordingly, it remains only to demonstrate necessity of the FSD characteristics.

Firstly, note that expandability suggests modularity, while it is well-established that, all other things being equal, reliability requires simplicity. That is, if the same end can be achieved in two different ways, the simpler is to be preferred. Also, reliability in the sense of fault-tolerance or self-healing or graceful degradation suggests redundancy, which is most simply achieved by modularity. Furthermore, modularity enhances maintainability, expecially if functionality is only gracefully degraded by the removal of any one module. For these reasons we shall take generic modularity to be a necessity.

It will become clear in the sequel that FSD modularity permits limitless expandability. In fact, only a small percentage of the capacity of each element need be devoted to "overhead" associated with the interconnectivity of the elements, and this percentage does not grow as additional elements are added. Also, in contrast to other approaches, each element can have its own OS (Operating System), and the various OSs need not be compatible. Thus, every computer in existence could be linked by the FSD approach without significantly degrading the total throughput capacity of the combined totality of separate computers.

In other generic approaches to distributed-processor architecture, no distinction is made between C elements and D/C elements. However, security maximization necessitates such a distinction. In fact, industry and government have already devoted vast resources to the optimization of "direct" security measures, including virtually "unbreakable" password systems. For present purposes, it will be assumed that such direct security measures are in fact optimal, and that the only security measures requiring further consideration are of the indirect variety. All indirect approaches to unauthorized access to a data base involve the introduction of an unauthorized program past the point guarded by the direct security system. However, this becomes categorically impossible if one splits off the data base from the remainder of the C element and establishes it as a non-programmable D/C element to which access may be made only in a direct manner through an optimized security system of the type already postulated. This was demonstrated pragmatically upon three occasions by the public offer of a $100,000 award (at the November, 1983 COMDEX in Las Vegas; at a subsequent COMDEX show in Atlanta; and at the July 1984 National Computer Conference in Las Vegas), which has also been announced nationally in BYTE magazine. Consequently we may take as "proved" the sufficiency and necessity of the presently defined split of architectural elements into C and D/C elements as regards the optimization of indirect security. In the sequel we shall refer to this kind of security as internal security.

It remains only to demonstrate the necessity of the four rules of interconnection of the two elements. We shall do this by supposing that each of the four rules is in turn abrogated, starting with the fourth rule, and demonstrating that the postulated optimality would thereby become degraded.

Suppose then that rule 4 were modified; i.e., that a D/C were allowed to initiate contact with a C element; how would it know when to initiate contact and what could it say if it were not appropriately programmed? So modifying rule 4 would, as already demonstrated, degrade the internal security of D/C elements, by requiring them to be programmable. Even if this factor were not present, modifying rule 4 would reduce system simplicity significantly, since the complexity of the communication network which can accept the initiation of contact from one side only is appreciably less than that required if the contacts could be bidirectional. Also the capacity of a C would be degraded if it had to have the information which would enable it to listen for possible contacts from D/C elements (and getting around this by introduction of an additional monitoring computer would also decrease system simplicity and increase costs). Consequently, rule 4 remains necesary.

Now consider modification of rule 3. In the first place, all of the disadvantages to modifying rule 4 would apply, mutatis mutandis. Similarly, one would have to allow the D/C's to be programmable if they were allowed to contact other D/Cs which would degrade internal security. Consequently, rule 3 remains necessary.

Next, consider the modification of rule 2. If Cs were allowed to be contacted directly, there would be an extra overhead on each application program, for each C would have to be able to monitor the communication network constantly for queries from other Cs. Also the complexity of the communication network itself would have to be increased, because the capability of acceptance of communication from only one side first simplifies both the communication network architecture and its protocols. Consequently, rule 2 remains necessary.

Finally, consider the modification of rule 1. Obviously the security restrictions must be required, or one of the prime desiderata would be lost. Furthermore, the purpose of separating the executable application programs from the data bases is to enhance throughput, by separating storage and retrieval (which may need to be accessed at kilohertz) from CPU operations (which may proceed at gigahertz); modification of this separation would introduce unnecessary bottlenecks, which can be seen as follows. Access to any information which must be shared between more than one C element creates a potential bottleneck. The first step in reduction of such a bottleneck is to take all information which is not needed by more than one C element and store it with the appropriate separate C elements. What is left is data that cannot be shared; it should be stored in a D/C element, where the attempt to access this data simultaneously by more than one C element does in fact constitute an unavoidable bottleneck. However, by this splitting between C and D/C elements, we have segregated the bottleneck and also minimized it. If the bottleneck is still too small, there now exists an orderly concept for splitting this D/C element into two or more separate D/C elements, for the purpose of reducing the bottleneck to an acceptable level. In summary, rule 1 permits one to segregate and minimize potential bottlenecks in a systematic way. Consequently, rule 1 remains necessary.

In conclusion, modification of any of the four rules of interconnection degrades the system relative to the specified performance criteria, and so is unacceptable. Without these rules, the amount of data in the associated communication network can become totally unmanageable; but with these rules, the system can grow very, very large. In fact, FSD provides a completely controlled environment in which the amount of data processible rises in direct proportion to the number of C elements added. The maximum potential size of the resultant processing system is literally limitless!

SECTION 8 - ESTIMATION/DECISION THEORY

C. T. LEONDES
UNIVERSITY OF CALIFORNIA - LOS ANGELES

A multilayer approach to strategic defense is identified as most desirable. Therefore, in each one of the threat trajectory phases of boost, post-boost, midcourse, late midcourse, and endoatmospheric means for threat track file development must be defined and developed as an absolutely essential precursor to intercept weapon assignment. A careful in-depth examination of the issues in each of these trajectory phases then defines the problems which estimation/decision theory must deal with, suggests candidate approaches for dealing with these problems and suggests research to be recommended where required.

8.1 ISSUES

8.1.1 Sensor Systems During Boost and Post Boost Phases

The BSTS (Boost Surveillance and Tracking System) is to gather initial target track file data and hand these data over to the SSTS (Space Surveillance and Tracking System) for continued track file generation as a precursor to weapon assignment during boost and post-boost phases, midcourse phase, and then to hand over track file data to late midcourse and endoatmospheric sensors such as AOA/AOS, TIR, and any other sensors which might be employed during the latter trajectory phases such as possible boosted probe sensor systems for the late midcourse and high endoatmospheric phases. The issues then include at least the following among others.

a. The sensor systems for BSTS and SSTS will almost certainly have to be 3 color (for target discrimination) multielement focal plane arrays. The Fletcher Committee suggested a multielement 3 color focal plane array of 10 million pixels or an array of 3,000 x 3,000 elements. Since this sensor is to be based on a BSTS or SSTS with an orbital operational lifetime of 10 years, the sensor system is not yet within the state of the art and is, therefore, an absolutely top priority essential developmental issue.

b. In addition to this, methods and criteria will have to be developed for these multi-element focal plane array sensors to allow for graceful degradation by suitable processing criteria.

c. Criteria and means for implementing them must be developed for determining when the useful lifetime of these multi-element focal plane arrays has been exceeded and must be replaced either by on board means through fall back equipment or some other means, including complete satellite replacement or replenishment or else by alternative coupled satellite configurations with fewer satellites; i.e., with the failed satellites eliminated from the overall BM/C^3 function.

d. Assuming now that the BSTS and SSTS have functioning sensor systems, threshold criteria, probably adaptive threshold criteria, have to be developed and implemented in the processing done at the focal plane array.

e. This then clearly establishes the very strong essential requirement for a totally adequate threat target signature data base. This target signature data base will have different sets of attributes during boost, post boost, midcourse, and (perhaps) late midcourse phases.

f. This target signature data base will be essential to the process or confining or limiting track file generation only to true threat targets and not at all to track file generation for decoys, debris, etc. The so called notion of "cradle to grave" track file generation of all objects, threat targets, decoys, debris, etc., would apparently doom an SDI system to failure from the beginning because the data processing requirements implied by "cradle to grave" tracking of all the many hundreds of thousands of all possible objects would represent an unattainable capability for a BM/C^3 system.

g. Assuming all of the above issues are recognized and dealt with, the SDI system would then utilize the BSTS and SSTS systems to determine which Soviet ICBMs have been fired, that is, which silos are empty and no longer targets for U.S. ICBMs. This is an issue which has to be handled carefully for the Soviets have demonstrated a silo reload capability for the reasons just suggested by the above process.

h. The next and final issue in this broad area is then the adequately accurate generation of threat tracks during the boost, post boost, midcourse, and late midcourse trajectory phases.

8.1.2 Precision Pointing and Tracking In Boost, Post-Boost, and Midcourse Phases

The BSTS and SSTS must have knowledge of the angular orientation of their sensor systems to an adequate degree, and this adequate degree will most certainly have to be with great precision. The BSTS requires this great precision during the boost phase primarily. The SSTS required this great precision during the boost phase, post boost phase, midcourse, and late midcourse phases. The only sensor system capable of providing the absolutely requisite and essential pointing or tracking accuracy is the stellar inertial system. The issues suggested by this then include among others the following.

a. The accuracy required by the stellar inertial system is closely coupled to and determined by the time of flight to the target of KEW systems, any possible midcourse guidance technique, and the FOV and terminal guidance accuracy capabilities of the KEW terminal intercept system. DEW systems, of course, have no time of flight, but, on the other hand require a pointing and tracking accuracy of 0.2 micro radians in the intensely high vibration/acoustic environment of a DEW. As a result DEW systems represent a most formidable, and probably unachievable, technology challenge for their possible utilization in SDI.

b. At present, estimates for the required accuracy of a stellar inertial reference system are about 2 micro radians and the state of the art is about 10 micro radians.

c. Another major and absolutely essential issue is the lifetime of the stellar inertial reference system. The desired lifetime is 10 years and this is at present not at all within the state of the art.

d. Another major issue is means for determining whether or not the stellar inertial system is functioning adequately, that is, to the degree of accuracy required. In other words as the performance of the stellar inertial reference system degrades in accuracy with time, as it will, how can it be determined when its performance is no longer adequate.

e. Another issue, therefore, is to what degree and how can techniques for graceful degradation or fault tolerance for stellar inertial systems be developed and implemented.

f. Finally, how can criteria be developed and implemented for determining when the stellar inertial sensor system has failed and is, therefore, totally useless.

8.1.3 Algorithms for Multitarget Tracking During the Boost, Post-Boost, Midcourse, and Late Midcourse Phases

The nature of the sensor systems for the BSTS and the SSTS is that multi element focal plane arrays and passive sensors. As a result they can do only angle-only tracking. This then suggests the following issues.

a. The possibility of threat intercept during the boost phase of ground based KEW systems, based, for example, in Alaska, clearly calls out the requirement for a rather short time line for threat track file establishment. This then suggests the issue of very fast target tracking processing algorithms.

b. One such possibility is instant threat target location through the utilization of trilateration from 3 different SSTS systems, and the continuation of this target tracking process over some short time line. This then suggests the issue of satellite instant data coordination among 3 different cooperating satellites.

c. If angle-only tracking is utilized as in the case of the use of a single multi-element focal plane array from a single SSTS this suggests the issue of effective angle-only tracking algorithms. This includes the effective resolution of stochastic observability issues.

d. Assuming an adequate target signature data base is developed for all trajectory phases, the issue of discrimination is paramount as to its computational complexity and data processing requirements that result therefrom. Discrimination in this case refers to discrimination among different threat targets under the assumption that the target signature data base is adequate to result in essentially the immediate elimination of other objects such as decoys, debris, etc. The destination in sensor data processing here is between that data processing done at the focal plane array which passes on to the next processor only the threat target data and at the same time the processing at the focal plane array "totally" "filters out" the decoys, debris, etc. The threat target processor then carries processing to develop threat target track files.

8.1.4 Sensor Systems During Late Midcourse, High Endo, and Low Endo Phases

These include the AOA (Airborne Optical Adjunct) which is a proof of principal system, the AOS (Airborne Optical System) which the system to be eventually utilized, the TIR (Terminal Imaging Radar), and any other latter trajectory phase sensors.

a. All of the issues suggested above for the optical sensor systems of the BSTS and SSTS apply somewhat similarly to the AOA and the AOS, and therefore, do not need to be repeated here.

b. The remaining class of terminal trajectory phases sensor systems will be active sensor systems. Passive sensor systems can effectively utilize a threat target signature base, if it is adequate, to accomplish threat discrimination. Active systems or radar systems must process the returned signal and, in the processing carried out, determine threat discriminants. This means the active sensor systems or radar systems must process returns from all objects, threat targets, decoys, debris, etc. The issue suggested here is that the magnitude of the task can undoubtedly be enormously alleviated if effective means can be developed for instantly handing over track file data developed by other sensor systems to the terminal radar sensor systems so that radar tracking can largely be confined to the threat targets. The principal radar return signal processing for threat discrimination would then be largely, if not essentially totally, confined to what might be referred to as clutter (decoy, debris, etc.) rejection, just as in the case of so called clutter rejection by "look-down" "shoot-down" airborne radars.

8.1.5 Space Time Histories of Threat Targets now well Defined

If all the issues raised in the above sections can be effectively addressed then an adequately essential threat target space time history will have been achieved. With this in place the assignment of intercept weapons and threat target destruction can be carried out in all phases of the trajectory. This then raises the set of issues in the next sections.

8.1.6 Weapons During Boost and Post-Boost Phases

Most desirable would be the destruction of essentially all threat targets during the boost phase. Any remaining threat targets would then, hopefully, be destroyed during the post boost phases. Basically, the targets in these phases would be non maneuvering targets and so advanced techniques in estimation and decision theory would be adequate as these techniques either exist today or require further advanced developments. It is in the endo atmospheric trajectory phases wherein the threat target can maneuver that advances estimation and decision theory for the more complicated arena of differential game theory will have to be developed. In any event, this suggests the following issues.

a. During the boost phase the preferential destruct weapon category would be the DEW weapons because of their "instantaneous" destruction effect. This raises the following issue. Formidable technical hurdles will have to be cleared before DEW systems have a hope of being developed.

b. Another major issue is that these DEW systems will require pointing and tracking accuracies of .02 micro radians once the threat target is acquired and tracking is initiated. This accuracy is well beyond the state of the art, and is not likely to be achieved particularly in the enormous acoustic and vibration environment of DEW systems.

c. Considering now the issues in space based KEW systems, there are 2 categories of such KEW systems, electric rail guns and rocket propelled systems. Electric gun systems face formidable technology challenges, require enormous space based power systems, and estimates for each such satellite weapon system are several billion dollars each or about 1/2 trillion dollars for the constellation of satellite weapon systems. Two decades from now when such KEW space based systems might be deployed, the constellation can well be expected to cost $5 trillion to $10 trillion, figures that will undoubtedly be totally unacceptable.

d. Space based rocket propelled KEW systems appear to be more feasible, but the surveillance, acquisition, and tracking time lines imposed on the BSTS and SSTS would represent a formidable challenge in order to achieve intercept during the threat boost phase, as is most desirable. This also suggests the issues of the tight coupling of track file accuracy, time of flight of the intercept system to the threat either during boost or post boost phases, the FOV (Field of View) of the intercept terminal sensor system which is also related to midcourse guidance accuracy, and finally the terminal guidance accuracy of the self contained intercept system guidance system.

e. This then suggests the issue of the entirely new concept of the ground based KEW system and all the issues just raised in (d.) above apply here also.

8.1.7 Weapons During Late Midcourse, High Endo, and Lower Endo Phases

These are now fairly well defined as the ERIS, HEDI, Braduskill, and SRHIT systems and their evolutionary derivations. Because these are ground based systems which, basically, are not burdened with the formidable, almost unsurmountable, technical and cost challenges of the space based intercept systems, the issues tend not to be so staggeringly challenging. There are challenging issues though.

a. In the later (endo) trajectory phases, the target can be expected to maneuver. This then raises the issue of the development of requisite techniques for differential game guidance and control in a stochastic and non linear (bounded control) environment, a problem of enormous challenge.

b. In all the terminal trajectory phases, the threat target can be expected to utilize countermeasures to defeat the intercept system terminal sensor system.

c. In the late midcourse and high endo atmospheric phases, the threat target can be expected to be in a cloud of decoys, debris, etc. The intercept system terminal sensor system will be required to be able to do threat target discrimination in the terminal phase.

8.1.8 BM/C^3 Issues

During the boost, post boost, midcourse, and late midcourse threat trajectory phases, the SSTS would apparently be the BM/C^3 focal point. The incredible dynamics of the situation in those threat trajectory would suggest SSTS BM/C^3 autonomy except for executive human command authority to execute, interrupt, or terminate BM/C^3 responsibility at the individual SSTS platforms, albeit in a closely coordinated manner not only among the various platforms but also among other elements as well, of course, such as BSTS, the space weapons platforms the terminal trajectory defensive systems, etc. This suggests a host of issues, some of which are noted below. The terminal defensive system elements would solve similar issues but different issues as well. Some of these are noted here.

a. Firm criteria have to be established to provide executive command authority with the totally reliable information to initiate, interrupt, or terminate the operation of the SDI system. This information could come from a variety of sources, but certainly among the primary ones would be BSTS and SSTS and the fusion of data from these two systems.

b. In light of the above, such techniques as fail safe, fail determination, and other means of determining and maintaining the "health" of this enormously complex SDI system must be developed for all essential elements such as sensors, data processors, communication links, intercept systems, etc.

c. As the "all out" engagement develops, the SSTS BM/C^3 platforms would have to employ "totally" optimal commitment of resources, intercept systems, for example, and this clearly suggests a host of issues including raising to a much higher level of development the techniques for this incredibly complex optimal resource allocation problem.

d. The same set of issues raised in a, b, and c above also apply to the SDI systems for the late midcourse, high endo, and low endo phases.

8.2 CANDIDATE APPROACHES

8.2.1 Sensor Systems During Boost and Post-Boost Phases

The issues were discussed extensively but not exhaustively earlier. Some of the candidate approaches for dealing with these issues will now be discussed.

a. The determination of the effective lifetime of a multielement (10 million elements) focal plane array will probably require the periodic application of test patterns to the processor at the focal plane array. It does not appear at present that a feasibly practical means for testing the array detectors themselves except by some mechanical means of presenting a suitable IR source before the array and scanning the source over the entire array mechanically. In combination, these means can also establish, perhaps, graceful degradation.

b. Adaptive thresholding techniques for signal detection can perhaps also be periodically checked and confirmed by the periodic use of a similar mechanical scanning mechanism before the face of the array as just described, but this time would involve the inclusion of simulated false targets. This being known, the continued "goodness" of an adaptive thresholding technique can be periodically verified.

c. Any possible use of 3 different multi element focal plane arrays on 3 different SSTS platforms in a coordinated trilateration target acquisition technique can, perhaps, be dealt with simply by verifying that each individual focal plane array is functioning properly by the methods just described above.

d. False alarms are, of course, a major serious problem. The schemes described above in verifying the correct functioning of the focal plane arrays, including any adaptive thresholding technique, may also be adequate for essentially simultaneously verifying that false alarms are being continually effectively dealt with.

8.2.2 Precision Pointing and Tracking

As noted in the section on issues, the only suitable sensor system for the precision pointing and tracking required during the boost, post boost, and mid course trajectory phases is a stellar inertial sensor system. The accuracy required for target tracking for track file generation, as well as weapon guidance and control, is 2 micro radians, whereas today's state of the art is 10 micro radians. Additionally, a 10 year lifetime is required for this system, and this is well beyond the current state of the art. Furthermore, once a target is acquired and tracked in the boost phase, the space DEW weapon system, which receives target track file data during the boost phase as "handed over" by SSTS, must have a tracking accuracy of .02 microradians in an intense acoustic and vibration environment. These then suggest the candidate approaches now listed for the precision pointing and tracking problem.

a. The target tracking accuracy of 2 micro radians is adequate for the guidance and control of KEW weapons with their terminal sensors. Therefore, as one candidate approach to precision pointing and tracking with respect to KEW weapons, a careful tradeoff analysis can establish a pointing and tracking accuracy which alleviates the accuracy requirements as much as possible while, at the same time, maintaining system interface effectiveness. This would involve a careful tradeoff analysis between target tracking accuracy, time of flight of the KEW from the launching space platform to the vicinity of the target where the weapon FOV (Field of View) can acquire the target, and the magnitude of the intercept weapon terminal FOV sensor as well as the intercept weapon lateral maneuver capability.

b. The tracking accuracy for DEW weapons as used during the boost phase exclusively for HEL's and the boost and post boost phase for high energy neutral particle beam weapons is recognized as .02 micro radians. This would appear to be virtually an unattainable tracking accuracy but would certainly call for the development and utilization of the most accurate pointing control techniques possible.

c. The lifetime of the stellar inertial sensor is required to be 10 years. It would seem that the major candidate approach available to achieve is continual advances in design techniques.

d. Testing methods to be developed for and carried out on board the SSTS platform, as well as the weapon system platform to verify the sensor system "health" and to continue to monitor it, will have to be developed.

8.2.3 Algorithms for Multitarget Tracking (Including Angle-Only Tracking)

Since the sensor systems utilized on the space based platforms, BSTS and SSTS, are passive sensor systems, that is, IR (infrared) focal plane arrays, threat target tracking is, therefore, carried out by angle-only tracking techniques. This raises many fundamental issues as noted earlier in part in paragraph 8.1.3. Candidate approaches for dealing with these issues will now be discussed.

a. Angle-only tracking presents fundamental problems in observability, and so one of the candidate approaches will be to develop effective algorithms for achieving observabiity.

b. If an adequate threat target IR signature data base can be developed then instant detection of the targets can be achieved. As a result, a suggested approach for developing short time target tracks would be the use of trilateration from 3 SSTS satellites.

c. In the case of both a and b above, assuming that the threat targets are detected immediately, that is, decoys, debris, etc., are readily rejected by the sensors, then discrimination becomes a matter of discriminating amongst the threat targets. An approach suggested here would be the development of track files and rejecting inconsistencies as soon as they can confidently be rejected.

8.2.4 Sensor Systems During Large Midcourse, High Endo, and Low Endo Phases

These sensors will be a combination of focal plane array (passive) sensors as carried on the AOA/AOS and (active) electronically agile or phased array radars. The candidate approaches for dealing with the issues raised by these sensors include the following.

a. The motion of the AOA/AOS platforms enhances the observability of the threat targets, and these platform motions may be optimized in order to enhance observability or reduce R.M.S. error in the threat target track files.

b. There is the possibility of the conceptual system of boosted loitering vehicles with focal plane array to develop threat target track files during the late midcourse or exoatmospheric trajectory phase as well as during the high endoatmospheric phase.

c. The sensors in a and b above would detect the threat targets and, hopefully, instantly reject false targets. As a result these systems could contribute to an effective sensor fusion process by handing over their threat target track files to the terminal active sensors.

d. The electronically agile phased array radars, which are the terminal active sensors, would be able to acquire the threat target which might be in a cloud of debris and decoys and do so virtually instantly. As a result the commitment of interceptors could begin instantly based on the terminal radar's ability to discriminate the threat amongst the clutter which is the decoys, debris, etc.

8.2.5 Space Time Histories of Targets now well Defined

By developing an adequate threat target signature data base in the various threat trajectory phases of boost, post-boost, late midcourse, and endoatmospheric it is, perhaps, feasible to develop threat target track files throughout the trajectory and hand these over from one sensor system platform to the other as the threat target proceeds from one trajectory phase to the next.

8.2.6 Weapons During Boost and Post-Boost Phases

The DEW weapons can simply react to the threat target track files and be used against the threat boost vehicle during the boost phase. The DEW in this case could be either an HEL or a high energy neutral particle beam weapon during the boost phase on a high energy neutral particle beam weapon during the post boost phase. The formidable technical challenges and high cost of these systems were noted earlier in the paragraph on issues, 8.1.6, so no further discussion of DEW systems will be presented here. KEW systems are useful during all trajectory phases, and are either rocket propelled systems or electric rail gun systems. Of these two generic classes of KEW systems the rocket propelled system would appear to be more viable. Candidate approaches will be presented for this class of KEW systems.

a. A ground based KEW system can be ground based in Alaska. If the BSTS and SSTS surveillance systems can generate threat track file data according to a short time line then the KEW can be boosted to intercept the threat target during the boost (booster vehicle) and post boost (RV's) phases. The KEW terminal intercept system will have a terminal sensor with a certain FOV (Field of View). As a result, the surveillance system threat track file generation time line requirements are determined by (1) KEW boost time, (2) KEW midcourse guidance accuracy, (3) KEW terminal intercept sensor system FOV, (4) any maneuver capability the threat booster target might have, (5) any other factors.

b. A key issue in candidate approaches in this area deals with any countermeasures the threat target might generate to defeat the KEW terminal sensor system. If the terminal sensor is an EO system, then chaff dispensing countermeasures have to be dealt with and suitable counter countermeasures developed. If the KEW terminal sensor is a mm wave radar then similar counter countermeasures might be required.

c. If the KEW is space based, then the candidate approaches just discussed in (a.) and (b.) above apply again here also.

8.2.7 Weapons During late Midcourse, High Endo, and Lower Endo Phases

These are currently defined as ERIS, HEDI, SRHIT, and Braduskill or their future follow-on derivatives.

a. Since the threat target might have the potential for evasive maneuvering during the endoatmospheric phases, then the intercept vehicles guidance and control laws will have to be based on advances in differential games in stochastic environment and for nonlinear systems.

b. In addition, the threat target may employ countermeasure techniques and so the guidance and control of each of the intercept vehicles will have to include a counter countermeasures capability.

8.2.8 BM/C^3 Approaches

Whereas the various threat trajectory phases have different characteristics, there are also similarities.

a. During the boost and post boost phases, the coalignment of attitude sensors for the BSTS and SSTS vehicles will have to be essentially that of the weapons platforms or about 2 micro radians.

b. The SSTS will "act" in a totally responsible manner under human command authority during a battle.

c. Levels of initiation of SDI engagement, withdrawal, and conclusion will be defined and executed on the BM/C^3 platforms.

d. The same or similar BM/C^3 approaches will be implemented in the latter threat target trajectory phases.

8.3 RECOMMENDED RESEARCH

Based on the issues and approaches listed in Sections 8.1 and 8.2 a number of research problems will be listed here in the various paragraphs of Section 8.3.

8.3.1 Sensor Systems During Boost-Post Boost Phase

a. Research in fail soft focal plane arrays

b. Research in adaptive threshholding techniques for focal plane arrays

c. Research in algorithms for coupling the processing of sensor systems for coordinated BSTS and SSTS systems

d. Research in reassignment of BSTS and SSTS in the event of their failure

e. Research in failure detection and correction wherever feasible of BSTS and SSTS failures

f. Research in zero false alarm probability of all out threat attack

8.3.2 Precision Pointing and Tracking

a. Research in the development of 10 year lifetime stellar inertial precision attitude reference systems

b. Research in fail soft techniques development for stellar inertial reference systems

c. Research in the development of algorithms and technology to achieve 2 micro radian tracking accuracy for threat booster acquisition

d. Research in the development of BSTS and SSTS (multi element satellite) attitude control techniques, that is, precision adaptive control for distributed parameter systems

8.3.3 Algorithms for Multitarget Tracking (Including Angle-Only Tracking)

a. Research in discrimination techniques between threat target tracks and efficient algorithms for this process

b. Research in angle only tracking techniques including establishment of threat target observability techniques

c. Research in suboptional angle only tracking algorithms in order to achieve requisite accuracy and computational simplicity

d. Research in 3 satellite trilateration techniques for BSTS and SSTS satellites

e. Research in reassignment algorithms for continuing the development of track files in the event of any one BSTS or SSTS satellite failure.

8.3.4 Sensor Systems During Late Midcourse, High Endo, and Low Endo Phases

a. All of the research issues raised in paragraph 8.3.3 apply to AOA/AOS platforms

b. Research in radar sensor processing for threat discrimination and track file generation

c. Research in sensor fusion and track file generation for AOA/AOS and radar sensor systems

8.3.5 Space Time Histories now well Defined

a. Absolutely essential research in threat target signature data base generation in all trajectory phases

8.3.6 Weapons During Boost and Post-Boost Phases

a. As noted in earlier sections, DEW weapons require .02 micro radian tracking accuracy. Therefore, research is required in the precision adaptive control of elastic structures in a dense acoustic and vibration environment.

b. High energy neutral particle beam weapons score a soft kill. Therefore, research is required in when this is effected

c. KEW weapons may encounter counter measures and so research in terminal homing in a counter measures environment is essential.

d. The threat may attempt evasive maneuvers during boost and post boost phases and so research is required in stochastic nonlinear differential games for KEW terminal vehicle guidance and control.

e. Extensive research is required in the development of the closely coupled optimization of surveillance time track file generation, KEW fly out trajectory and fly out time, and KEW terminal vehicle guidance and control sensor systems, activators, and control laws.

8.3.7 Weapons During Late Midcourse, High Endo, and Lower Endo Phases

a. Many of the research problems noted above in 4.1.3.6 also apply here.

b. Research in boost trajectory optimization techniques for terminal phase threat trajectories for the intercept vehicle when the threat might be very highly maneuvering.

c. Research in optimization of "shoot-look-shoot" criteria and algorithms.

8.3.8 BM/C^3 Research Issues

a. During boost and post boost phases and as the threat track files are developed the SSTS must execute the responsibilities of battle manager. Therefore, BM/C^3 research is essential in resource, that is, intercept weapons, allocation in an optimal way.

b. As the engagement proceeds SSTS BM/C^3 platforms may be destroyed or failed. Research is necessary into the development and definition of robust SSTS BM/C^3 functions.

c. This research must include different elements of each SSTS BM/C^3 platform such as sensors, attitude references, control devices, data processors, etc., and fail soft techniques must be developed in research.

d. All of (a), (b), and (c) above must also be BM/C^3 research for the terminal battle management platforms.

SECTION 9 - DATA MANAGEMENT

Karen D. Gordon

The Mitre Corporation

This section addresses the data management technology area. It focuses on the "database system" approach to data management, where a database system is viewed as being composed of two complementary components: a database and a database management system. The database is an integrated collection of data that is used by multiple applications; the database management system is a software package or hardware/software system that is dedicated to the management of the collection of data. Together, the database and the database management system provide a service to higher level applications; namely, they store and manage data on behalf of the higher level applications. The purpose of this section, therefore, is to identify some of the key issues that must be addressed in developing a database system for the BM/C^3 component of an SDI system.

The remainder of this section is organized as follows. Section 9.1 introduces the issues. Section 9.2 details the issues, and outlines candidate approaches to the issues. Section 9.3 summarizes directions for future research.

9.1 INTRODUCTION OF ISSUES

This section begins by presenting an overview of database systems. It then points out the distinguishing features of distributed database systems, which merit special consideration in light of the geographic and spatial distribution of the SDI system components. It concludes by pointing out the challenges that the SDI presents to database system technology.

9.1.1 Overview of Database Systems

Prior to the emergence of database system technology in the late 1960s, the standard approach to data management was for each application to have its own private set of files. But in the database system approach to data management, data used by different applications is merged into an integrated collection of data, i.e., a database, that is managed by a dedicated system known as a database management system. The database management system may be a software package that runs on a general-purpose computer system, or it may be part of a special-purpose hardware/software system that not only manages the data but also stores the data. Such a hardware/software system is referred to as a backend database system.

The development of database system technology was motivated by several objectives:

a. To achieve centralized control over data, which had come to be recognized as a valuable asset of an organization

b. To make data generally available; i.e., to "unlock" it from specific applications

c. To reduce redundancy, by enabling applications to share data

d. To reduce the inconsistencies that arise from uncontrolled and unmonitored redundancy

e. To achieve data independence; i.e., to make applications immune to changes in the physical organization of data by making the physical organization transparent to the applications.

In meeting the above objectives, database systems introduce new problems of their own:

a. How to maintain the consistency of shared data; i.e., how to control the executions of concurrent applications, so that their reads and writes overlap in a logically consistent manner, even in the presence of failures

b. How to achieve an acceptable level of performance for a given application, when the data accessed by the application is managed by generalized software rather than by application-specific software

c. How to maintain the security of data that is stored in a shared, integrated database.

These problems are explicitly addressed by database system technology; that is, database management systems incorporate specific mechanisms to deal with them. While the existing mechanisms are adequate for many applications, they fall short of meeting the requirements of other, more stressing applications. Therefore, the problems remain as issues today, even though they have received considerable attention over the past ten to fifteen years.

9.1.2 Overview of Distributed Database Systems

In a distributed database system, data is physically distributed over multiple computer systems, which are interconnected by a communication network; but, at the same time, the data is logically viewed as belonging to a single, unified database. Each computer system, or site, possesses an autonomous processing capability. That is, each site can autonomously process applications that require only local data. However, the sites also cooperate with one another, so that applications requiring data from multiple sites can be processed and so that the global consistency of the database can be maintained. Distributed database systems thus emphasize both site autonomy and cooperation among sites.

Several factors contributed to the development of distributed database system technology. On one hand, the development was motivated by several objectives:

a. To better meet the needs of decentralized organizations, by decentralizing the database systems

b. To support incremental growth, which is an inherent feature of distributed systems

c. To reduce communication costs, by making data resident at sites that frequently use it

d. To improve performance, by making data more readily available to applications that use it, as in c.

e. To improve reliability and availability, through data replication as well as through distribution.

On the other hand, the development was made feasible by advances in several technology areas: microprocessors, communication networks, (centralized) database systems, and distributed operating systems.

While distributed database systems offer many advantages over centralized database systems, they do so only at the cost of added complexity. The problems of data consistency, performance, and security that are incurred by centralized database systems become even more difficult in a distributed environment. For example, data consistency is complicated by the fact that data is replicated (for performance and reliability purposes); performance is complicated by the low bandwidth, high delay communication paths that exist between sites; and security is complicated by the presence of the communication network, which represents a new avenue for unauthorized access to data.

In addition, distributed database systems introduce new problems, including the allocation of data across the nodes of a distributed computer network and the integration of heterogeneous database systems. The issues of <u>data allocation</u>, <u>data consistency</u>, <u>performance</u>, and <u>security</u> are singled out in this report as being the most critical data management issues for SDI BM/C^3. While the integration of heterogeneous systems may indeed prove to be a challenge in the SDI context, it is not an SDI-unique problem. Instead, it is a problem that is being faced by many corporations and government components, as the trend toward the interconnection of computer systems grows. Therefore, the integration of heterogeneous systems is not suggested as a major BM/C^3 research issue, since it is believed that others will be compelled to develop solutions to the general problem. Research prototypes of heterogeneous database systems have already been built and extensively studied at the Computer Corporation of America [Land82] and at the Honeywell Corporate Computer Science Center [Devo80]. Although these prototypes do not encompass all the features that might be required of heterogeneous systems in the SDI context, they are indicative of the major research efforts that are being conducted in the area.

9.1.3 The SDI Environment

The SDI environment presents a number of challenges to database system technology:

a. Data resides at geographically and spatially distributed sites, which means that communication between sites is slow and costly.

b. Many of the sites are constantly moving, which means that the topology of the underlying network is constantly changing.

c. The sites, as well as communication links between sites, face a hostile environment, which implies frequent site failures and communication outages.

d. Update rates are high.

e. Retrieval rates are high.

f. Performance (i.e., real-time response) is critical.

g Reliability is critical.

h. Security is critical.

Taken individually, most of these factors can be met by current database system technology. However, when taken together, they present a uniquely stressing set of requirements that cannot be met by current technology.

Of course, not all of the above factors apply to all data at all times. For example, the update rates and retrieval rates may not be high at all times. Nevertheless, all of the factors do apply to certain critical data, such as data on the states of the offense, defense, and environment, at the time of an attack. This report focuses on the cases in which all of the factors must be considered.

9.2 CANDIDATE APPROACHES

This section discusses the four data management issues identified above: data allocation, data consistency, performance, and security.

9.2.1 Data Allocation

The allocation of data to the sites of a computer network can be viewed as a two-step process [Ceri84]. In the first step, the data is decomposed into units known as fragments, which are the logical units of allocation. In the second step, the fragments, as well as replications of the fragments, are assigned to the physical sites of the network. The purpose of replications is to increase the performance, reliability, and availability of the database system.

In the case of a relatively stable network, which has been the traditional assumption in distributed database research, both the fragmentation and allocation can be done at design time. But in a highly dynamic network, such as that of the SDI system, the allocation must be dynamic.

In the SDI context, the dynamic allocation of data is, in a sense, a secondary resource allocation problem. The primary resource allocation problem is the high-level assignment of BM/C^3 applications (including coverages, or battle areas, of the applications) to BM/C^3 components. The allocation of data must complement this high-level allocation of applications. In this way, the performance and effectiveness of the SDI system as a whole can be optimized.

The high-level resource allocation problem is beyond the scope of this section; it is addressed in Sections 5, 7, and 8 of this report. With respect to the data allocation problem, research on algorithms for the allocation of data in a highly dynamic network is called for. The algorithms should be capable of responding to the dynamic topology of the SDI network and to the dynamic allocation of BM/C^3 functions across the network. Since the algorithms will have to be executed in real time, they should be computationally efficient.

Research on the dynamic allocation of data is currently being conducted by Y.I. Gold and F.J. Maryanski at the University of Connecticut [Gold84] and by C.V. Ramamoorthy at the University of California at Berkeley [Rama85]. This research should be followed, and its applicability to the SDI environment should be investigated.

9.2.2 Data Consistency

Data consistency is a fundamental issue in database system technology. As such, it has been, and continues to be, the focal point of considerable research. Various mechanisms for maintaining data consistency, in both centralized and distributed environments, have been proposed in the literature [Bern81], [Ceri84], [Kohl81], [Roth 77]. Most are based on the concept of a transaction.

A transaction is a user-bounded or application-bounded (e.g., by BEGIN_TRANSACTION and END_TRANSACTION commands) sequence of database operations that, by definition, constitutes a unit of consistency. That is, it is assumed that the execution of a transaction transforms a database from one consistent state to another, the implication being that partial executions lead to inconsistencies. Since reads do not change the state of a database, writes (i.e., updates) are the critical operations. Thus, a transaction must be an atomic unit of execution, in the sense that either all or none of its updates must be executed. For example, in a funds transfer, either both the debit and credit must be executed or neither must be. Moreover, in the case of a distributed database system with replicated data, all copies must be included in the coordination process. A transaction must also be an atomic unit of execution in another sense. Namely, the execution of a transaction must be logically isolated from the executions of other transactions. Therefore, maintaining data consistency entails maintaining the internal consistency (i.e., the "all or none" property) of each individual transaction, as well as the mutual consistency (i.e., the logical isolation) of concurrent transactions, even in the presence of failures.

The standard mechanisms for maintaining data consistency are:

a. Recovery mechanisms, such as logging and shadow paging,

b. Commit and termination protocols, and

c. Concurrency control algorithms.

It should be pointed out that these mechanisms do not work in isolation, but instead work together to ensure the consistency of data.

The aim of logging and shadow paging mechanisms is to ensure the internal consistency of transactions in the presence of failures. As the database management system is processing a transaction, it records and/or retains enough information so that the transaction's operations can be undone or redone if necessary (i.e., if a failure occurs).

The aim of commit and termination protocols is to ensure the internal consistency of distributed transactions, i.e., transactions that involve updates at multiple sites of a distributed database. Commit protocols are designed to coordinate the sites participating in a distributed transaction, so that all sites either unanimously commit or unanimously abort the transaction. The coordination is achieved through successive rounds of messages that are exchanged among the participating sites. Termination protocols are designed to be used in conjunction with commit protocols, to enhance the availability offered by the usage of commit protocols alone. Upon the failure of one of the sites participating in the commitment of a transaction, the remaining operational sites invoke a termination protocol. The termination protocol allows the operational participants to correctly terminate the transaction, without having to wait for the failure to be repaired.

The aim of concurrency control algorithms is to ensure the mutual consistency of transactions; i.e., to synchronize the executions of concurrent transactions, so that the transactions' reads and writes are interleaved in a logically consistent manner. Maintaining logical consistency means avoiding anomalies such as "dirty reads" and "lost writes." A dirty read occurs when a transaction reads data that has already been updated by what should be a "later" transaction. A lost write occurs when a transaction writes over the update of what should be a later transaction.

The two basic approaches to concurrency control are the locking of data and the timestamp ordering of transactions. It should be noted that locking implies waiting, which leads to the possibility of deadlock. So, locking must be coupled with deadlock prevention or detection mechanisms, which in turn lead to the restarting of transactions. Timestamp ordering, on the other hand, begins with restarting, namely, the restarting of transactions that are found to be out of order. Then, in order to minimize restarts, which are relatively expensive operations, waiting is introduced. Therefore, each of these approaches to concurrency control involves both waiting and restarting.

Having reviewed the features of the standard mechanisms for maintaining data consistency, let us see why they fall short of meeting the requirements of the SDI environment. The key is to recognize that the objective of the mechanisms, and not just the mechanisms themselves, is the problem. The mechanisms are designed to maintain the absolute consistency of distributed data in the presence of failures and concurrency. While this objective is attractive as a theoretical objective and is valid (even mandatory) in some environments (e.g., banking), it is not viable in the SDI environment or in other critical real-time environments. The reason is that absolute consistency can be achieved only at the cost of timeliness and availability, which cannot be sacrificed in the SDI environment. The concurrency control mechanisms depend upon waiting and restarting, which introduce delays and block transactions from proceeding. The commit and termination protocols depend upon message passing, which introduces further delays. Moreover, isolated sites and network partitions can be blocked from processing transactions. The blocking occurs because, in the absence of communication, the cooperation necessary to maintain absolute consistency is not possible. Therefore, the only way to maintain absolute consistency is to block partitions; in that way, they are certain to introduce no inconsistencies, since they can make no changes.

Since timeliness and availability cannot be sacrificed for the sake of consistency, alternatives to the traditional database system approach of absolute consistency must be examined. The extreme alternative would be to view the database at each site as being local to the site, rather than part of a unified distributed database, and to have no cooperation among sites. More promising alternatives, which aim for a balance of site autonomy and cooperation among sites, lie between the two extremes. These balanced alternatives are the ones that should be addressed in SDI data management research efforts. Two general approaches to achieving an effective balance are presented below.

Local Databases with Higher Level Coordination

In this alternative, the data at each site (i.e., platform) is considered to constitute a local database, as far as the database management systems are concerned, and yet the databases are to some extent coordinated. The coordination derives not from the database management systems, but from higher level components of the BM/C^3 system. For example, the coordination may be implemented by specific applications (e.g., track correlation, weapon allocation, etc.), or it may be implemented by a more general component such as a distributed controller. In either case, system effectiveness is enhanced through the introduction of controlled and monitored data sharing and data redundancy. In particular, the coordinating component directs the passing of critical data among sites.

Distributed Database with Emphasis on Availability

In this alternative, the data at each site is considered to be part of a unified distributed database, but absolute data consistency is traded for enhanced timeliness and availability. The goal is to maximize the autonomous processing capability of sites, while at the same time maintaining a unified database. The burden of meeting this goal is placed on the database management system.

To increase the autonomous processing capability of sites, the blocking that is imposed on sites by traditional distributed database management systems must be reduced. Three causes of blocking are considered here: communication failures, replication control, and concurrency control.

Communication failures can lead to isolated nodes and network partitions. In the traditional approach to distributed database management, network partitions must be blocked from processing certain transactions, since consistency cannot be guaranteed in the absence of communication. But, if inconsistencies can be tolerated, the network partitions can be allowed to continue processing transactions, in which case they may diverge. The problem then becomes how to detect and resolve inconsistencies. This problem is being investigated by H. Garcia-Molina and others at Princeton University [Alon] [Garc83b]. The general concensus seems to be that resolving inconsistencies requires application-specific knowledge.

Replication control can lead to "blocking" in the following sense. When a site makes an update to local data that is replicated at other sites, it cannot complete its update (i.e., it cannot make the update visible to local applications) until the update has been coordinated at all the other sites. The coordination involves communication, which can lead to unacceptable delays in the SDI and other critical real-time networks with widely distributed components. A way to get around this problem is to relax consistency constraints by allowing updates to be optionally deferred to replications at remote sites, as in the performance-oriented algorithms of Distributed INGRES [Ston79]. A variation of this general approach, in which updates are synchronized within a group of "hot" or up-to-date sites, is proposed in [Gane84] and [Rama85].

Concurrency control algorithms are designed to limit the amount of overlapping that can occur among the executions of concurrent transactions. In meeting this objective, they introduce blocking in the form of waiting and restarting, as discussed above. The blocking can be reduced in various ways. One approach, which applies to timestamp ordering algorithms, is to maintain multiple versions of data items [Bitt85] [Reed78]. In this approach, transactions are never restarted because of read operations (as they can be in the basic timestamp ordering algorithms); instead, each read operation is directed to the version with the largest timestamp less than the timestamp of the read operation itself. Another approach is to allow read operations to specify the degree of consistency that they require, as suggested in [Bitt85], [Garc82], and [Mary]. A third approach is to incorporate semantic knowledge into concurrency control algorithms [Garc83a], which enables the algorithms to allow more concurrency than when they must rely on strict serializability as the only consistency criterion.

9.2.3 Performance

This section addresses the topic of performance. It begins by pointing out some general performance issues, i.e., issues that apply to the performance of the BM/C^3 system in general and not just to the performance of the data management component. It then goes on to discuss database system performance and, in particular, approaches to improving database system performance.

Performance Sizing

In order for gaps between technological capabilities and SDI performance requirements to be identified, and thus for research needs to become apparent, the performance requirements must be known. But performance requirements can be determined only in the context of specific SDI architectures. Therefore, competing SDI architectures must be refined to the extent that reasonably tight and credible bounds on performance requirements can be deduced.

Due to the overwhelming complexity of interactions in the SDI environment, experimentation (e.g., simulation) is critical to the performance sizing process. That is, experiments must be conducted to determine the loads that are placed on the various components of the SDI BM/C^3 system under various architectures and various conditions.

Performance Verification

Performance is critical in the SDI context. Therefore, to ensure the effectiveness of an SDI system, not only its functionality but also its performance must be verified. Again, experimentation is critical. Although the experiments must rely on simulation to some extent, they should incorporate actual hardware and software components as they become available.

Moreover, formal performance verification methodologies and tools need to be developed. The methodologies should be designed so that performance is explicitly addressed in all phases of system design and development.

Database System Performance Improvement

Certain data-intensive BM/C3 functions, such as track correlation and weapon allocation, may demand the use of special-purpose hardware and/or software to meet performance requirements. The hardware/software can be introduced at different levels of the BM/C3 architecture.

At the lowest level, special-purpose database access methods can be introduced. For example, in spatial database systems [Fink84], [Gutt84], [Robi81], [Rous85], special access structures are implemented that make it easy to find the "nearest neighbors" of an object or point in n-dimensional space. The efficient and direct spatial search offered by this approach may prove to be useful to some of the critical BM/C3 functions. However, further research, especially in the context of specific BM/C3 functions, is needed.

At the next level, special-purpose hardware/software can be introduced to perform the data management function. Such special-purpose architectures are referred to as database machines. Various approaches to the design of database machines have been proposed in the literature [Date83], [DeWi81], [Epst80], [Hawt82], [Nech84], [Smit79]. The approaches that offer the largest performance gains are those based on parallel processing, as in the Teradata database machine [Nech84]. The applicability of parallel architectures to the BM/C3 data management problem should be explored.

At the highest level, special-purpose architectures can be introduced to perform some of the critical BM/C3 functions themselves. These architectures would be application-specific; that is, they would be designed to be dedicated to a specific function. They would offer enhanced performance, but only at the cost of generality and flexibility. Therefore, they should be considered only if performance requirements cannot be met by the use of more generally applicable architectures.

9.2.4 Security

The SDI BM/C^3 database system(s) will presumably be required to store and manage data of different security levels (i.e., Unclassified, Confidential, Secret, Top Secret). Thus, the issue of multilevel secure data management arises. First, the question of whether or not multilevel security is actually required must be addressed. The alternative would be to utilize a system high mode of operation, in which all users must be cleared to the level of the most highly classified data. The problem with this approach is the cost and risk associated with the large number of personnel clearances involved. The tradeoffs between these two approaches--multilevel secure data management and system high mode of operation--need to be investigated.

Assuming that multilevel secure data management is a requirement, the problem then becomes how to achieve it. This problem was thoroughly analyzed at the 1982 Air Force Summer Study on Multilevel Data Management Security [Comm82]. The study identified both near-term solutions and long-term research issues, which are summarized below.

Near-Term Solutions

One group of study members was tasked to formulate near-term solutions to the multilevel secure data management problem. In the solutions, the database management systems are viewed as being untrusted (i.e., subject to security holes or corruption by Trojan Horse code). Multilevel security is achieved by placing a trusted "filter" in front of the untrusted database management systems. Three specific architectures based on this trusted filter approach were singled out as being the most promising. They each assume the existence of only two security levels: "Lo" and "Hi."

The first architecture is referred to as a kernelized database management system. In this architecture, Hi and Lo data are stored together, but they are logically separated by a trusted operating system based on security-kernel technology. The Hi data is managed by a Hi database management system (Hi DBMS); and the Lo data is managed by a separate Lo database management system (Lo DBMS). The trusted operating system ensures that the Lo DBMS can access Lo data only and that the Hi DBMS can access Hi data only. On the user side, the trusted filter ensures that Lo users can access the Lo DBMS only, but Hi users are allowed to access both the Hi DBMS and the Lo DBMS. In this way, Lo users can get to Lo data only, while Hi users can get to Hi and Lo data. The disadvantages of this approach are the overhead associated with the secure operating system and the fact that Hi queries have to deal with two database management systems.

The second architecture is based upon the physical separation of data. In the simpler of the two variations of this architecture that were proposed by the study group, data is physically separated into a Hi database and a Lo database, each of which is managed by its own database management system. Thus there is a Hi backend database system and a Lo backend database system. The trusted filter ensures that Lo users can access the Lo database system only, but, again, Hi users are allowed to access both systems. While this approach does not suffer from the secure operating system overhead incurred by the kernelized approach, it does still force Hi queries to deal with two database systems.

The third architecture is based upon a mechanism referred to as an integrity lock. In the integrity lock approach, there is a single database system, which stores and manages both Hi and Lo data; but the Hi and Lo data are cryptographically separated. That is, the trusted filter appends a cryptographic checksum to each data item and its associated security label as they are stored in the database, and then verifies the checksum upon retrieval. In this way, changes to data and/or labels can be detected. The problem with this approach is that it is subject to some Trojan Horse vulnerabilities, since the untrusted database management system has access to all data. The advantages and disadvantages of this architecture are detailed in [Clay83] [Grau84].

The applicability of these three architectures to the SDI environment needs to be considered. While none of the three completely solves the multilevel security problem (e.g., none addresses the inference problems noted below), they do each offer enhanced security.

Long-Term Research Issues

Another group of study members was tasked to identify long-term research issues in multilevel secure data management. They pointed out that database system security, while similar to operating system security (which has received considerably more attention than database system security) in many respects, does present unique problems. The source of most of the problems is the fact that database system security must deal with a finer level of granularity than operating system security. In particular, database systems may need to protect data at the record level or below, whereas operating systems generally protect data at the file level. As a result, database systems face higher frequencies of access to protected objects. In addition, database systems must be concerned with the unauthorized disclosure of information through the inference of classified data from unclassified data (as in aggregation, derivation, and isolation [Burn85]).

The threat that inference poses in the SDI environment should be evaluated. If the threat turns out to be serious, then research to address the threat must be undertaken. The study group recommended that the problem of inference be approached through the classification and protection of database views.

9.3 SUMMARY OF RECOMMENDATIONS

This section concludes the discussion of data management by summarizing the research needs suggested above:

a. Data allocation. Distributed, adaptive algorithms for the dynamic allocation of data in a highly dynamic network need to be developed.

b. Data consistency. Alternatives to absolute data consistency need to be investigated. These alternatives may involve moving from local databases toward more coordination, or moving from traditional distributed databases toward more availability.

c. Performance sizing. Performance requirements need to be determined. Only then can the shortcomings of current technology be identified.

d. Performance verification. Formal performance verification methodologies and tools need to be developed. The methodologies should incorporate experimentation, in such a way that performance is demonstrated on actual hardware and software when feasible.

e. Database system performance improvement. Methods for improving the performance of database management systems, in the context of the data-intensive SDI BM/C^3 functions, need to be investigated. These methods may involve special-purpose database access methods or special-purpose parallel database machines.

f. Multilevel security versus system high mode of operation. The trade-offs between these two approaches to security need to be analyzed.

g. Implementation of multilevel secure data management (assuming that it is a requirement). The applicability of the near-term solutions (proposed in the 1982 Air Force Summer Study) to the SDI environment needs to be examined. The threat that inference poses in the SDI environment needs to be evaluated, and then addressed if it is considered to be serious.

Progress in these areas is critical to the development of an effective SDI BM/C^3 data management system.

9.4 REFERENCES

Alon R. Alonso, et al., "Distributed Computing Research at Princeton," Princeton University, Princeton, New Jersey, undated.

Bern81 P.A. Bernstein and N. Goodman, "Concurrency Control in Distributed Database Systems," ACM Computing Surveys, Vol. No. 2, 13, June 1981, pp. 185-221.

Bitt85 George W. Bittner, "Multiversion Concurrency Controller," Master's Thesis Proposal, University of Connecticut, Storrs, Connecticut, October 1985.

Burn85 R.K. Burns, "DBMS Integrity Lock Mechanism," WP-26145, The MITRE Corporation, Bedford, Massachusetts, 1985.

Ceri84 S. Ceri and G. Pelagatti, Distributed Databases: Principles and Systems, McGraw-Hill, Inc., New York, 1984.

Clay83 B.G. Claybrook, R.D. Graubart, M.D. Makuta, and M.M. Zuk, "Architectures for Secure Database Management Systems," MTR9103, The MITRE Corporation, Bedford, Massachusetts, 1983.

Comm82 Committee on Multilevel Data Management Security, "Multilevel Data Management Security," Technical Report, Air Force Studies Board, National Research Council, National Academy Press, Washington, D.C., 1982.

Date83 C.J. Date, Introduction to Data Base Management Systems, Vol. II, Addision-Wesley, Reading, Massachusetts, 1983.

Devo80 C. Devor and J. Weeldreyer, "DDTS: A Testbed for Distributed Database Research," Honeywell Report HR-80-268, Honeywell Corporate Computer Science Center, Bloomington, Minnesota, 1980.

DeWi81 D.J. DeWitt and P.B. Hawthorn, "A Performance Evaluation of Database Machine Architectures," Proceedings of the Seventh International Conference on Very Large Data Bases, 1981, pp. 199-213.

Epst80 R. Epstein and P. Hawthorn, "Design Decisions for the Intelligence Database Machine," Proceedings of the 1980 National Computer Conference, 1980, pp. 237-241.

Fink84 R.A. Finkel and J. L. Bentley, "Quad-Trees -- A Data Structure for Retrieval on Composite Keys," Acta Informatica Vol. 4, 1984, pp. 1-9.

Garc82 H. Garcia-Molina and G. Wiederhold, "Read-only Transactions in a Distributed Database," ACM Transactions on Database Systems, Vol. 7, No. 2, June 1982, 209-234.

Garc83a H. Garcia-Molina, "Using Semantic Knowledge for Transaction Processing in a Distributed Database," ACM Transactions on Database Systems, Vol. 8, No. 2, June 1983, pp. 186-213.

Garc83b H. Garcia-Molina, et al., "Data-patch: Integrating Inconsistent Copies of a Database after a Partition," Proceedings of the Third Symposium on Reliability in Distributed Software and Database Systems, 1983.

Gold84 Y.I. Gold and F.J. Maryanski, "Dynamically Reconfigurable Distributed Database Systems: A Research Proposal," University of Connecticut, Storrs, Connecticut, submitted to and accepted by the U.S. Army Research Office, 1984.

Grau84 R.D. Graubart, "The Integrity Lock Approach to Secure Database Management," MTR9161, The MITRE Corporation, Bedford, Massachusetts, 1984.

Gutt84 A. Guttman, "R-TREES: A Dynamic Index Structure for Spatial Searching," Proceedings of ACM SIGMOD 1984, 1984, pp. 47-57.

Hawt82 P.B. Hawthorn and D.J. DeWitt, "Performance Analysis of Alternative Database Machine Architectures," IEEE Transactions on Software Engineering, Vol. SE-8, No. 1, January 1982, pp. 61-75.

Kohl81 W.H. Kohler, "A Survey of Techniques for Synchronization and Recovery in Decentralized Computer Systems," ACM Computing Surveys, Vol. 13, No. 2, June 1981, pp. 149-183.

Land82 T. Landers and R.L. Rosenberg, "An Overview of MULTIBASE," Distributed Data Bases, H.J. Schneider (editor), North-Holland Publishing Company, 1982, pp. 153-183.

Mary F. Maryanski, "Concurrency Control in a Dynamic Environment," University of Connecticut, Storrs, Connecticut, undated.

Nech84 P.M. Neches, "Hardware Support for Advanced Data Management Systems," IEEE Computer, Vol. 17, No. 11, November 1984, pp. 29-40.

Rama85 C.V. Ramamoorthy, "Conceptual Design of a Distributed Data Management System," October 1985.

Reed78 D.P. Reed, "Naming and Synchronization in a Decentralized Computer System," Ph.D. Dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1978.

Robi81 J.T. Robinson, "The K-D-B Tree: A Search Structure for Large Multidimensional Dynamic Indexes," Proceedings of ACM SIGMOD 1981, 1981, pp. 10-18.

Roth77 J.B. Rothnie and N. Goodman, "A Survey of Research and Development in Distributed Database Management Systems," Proceedings of the Third International Conference on Very Large Data Bases, 1977, pp. 48-62.

Roth77 J.B. Rothnie and N. Goodman, "A Survey of Research and Development in Distributed Database Management Systems," *Proceedings of the Third International Conference on Very Large Data Bases*, 1977, pp. 48-62.

Rous85 N. Roussopoulos and D. Leifker, "Direct Spatial Search on Pictorial Databases Using Packed R-trees," *Proceedings of ACM SIGMOD 1985*, 1985, pp. 17-31.

Smit79 D.C.P. Smith and J.M. Smith, "Relational Data Base Machines," *IEEE Computer*, Vol. 12, No. 3, March 1979, pp. 28-39.

SECTION 10 - NETWORKS

Nils R. Sandell
Alphatech

The feasibility of space-based ballistic missile defense (BMD) depends critically on the significant advances in communications and computation technology that have been achieved over the past decade. However, fully realizing the benefits of the hardware technology requires that new concepts in communications network algorithms and protocols be developed.

Current network concepts, algorithms, and protocols have been developed under the assumptions that the network will be topologically fixed over time, or at worst suffer occasional, minor changes due to component failures or repairs, and that the demand for communication resources will fluctuate slowly. Unfortunately, the communications networks required to support battle management/command, control and communications (BM/C^3) for BMD do not satisfy these assumptions. Important system elements (satellites, aircraft, rockets) are in motion, so that the connectivity of the network will undergo dramatic changes every few minutes. Should the BMD system ever need to be utilized, it would suffer extensive hard (physical destruction) and soft (jamming) damage at multiple nodes simultaneously. Moreover, in the worst case scenario of a simultaneous launch of all Soviet missiles, the communications load on the network would increase suddenly from an extremely light load associated with system status reporting and testing to its maximum design load.

Thus research is required in the development of new communications network algorithms and protocols to support the requirements imposed by the stressing BMD application. In this section we will discuss the technical issues associated with the development of algorithms and protocols, describe potential approaches and recommend research directions. We will focus on network layer issues here; physical and data link layer considerations are discussed in the next section while issues associated with host-level layers are discussed in the computer systems and data management section.

10.1 ISSUES

Figure 10-1 is a visualization of a portion of a communications network used to interconnect BMD system elements that are not physically collocated. Following ARPANET terminology, we depict host computers that perform BMD functions and IMPs (Interface Message Processors) that connect Hosts to the network and that perform specialized communications functions. The total communications network includes Hosts, IMPs, and communication links; the terminology communications subnet or backbone network is sometimes used to refer to the IMPs and communications links alone.

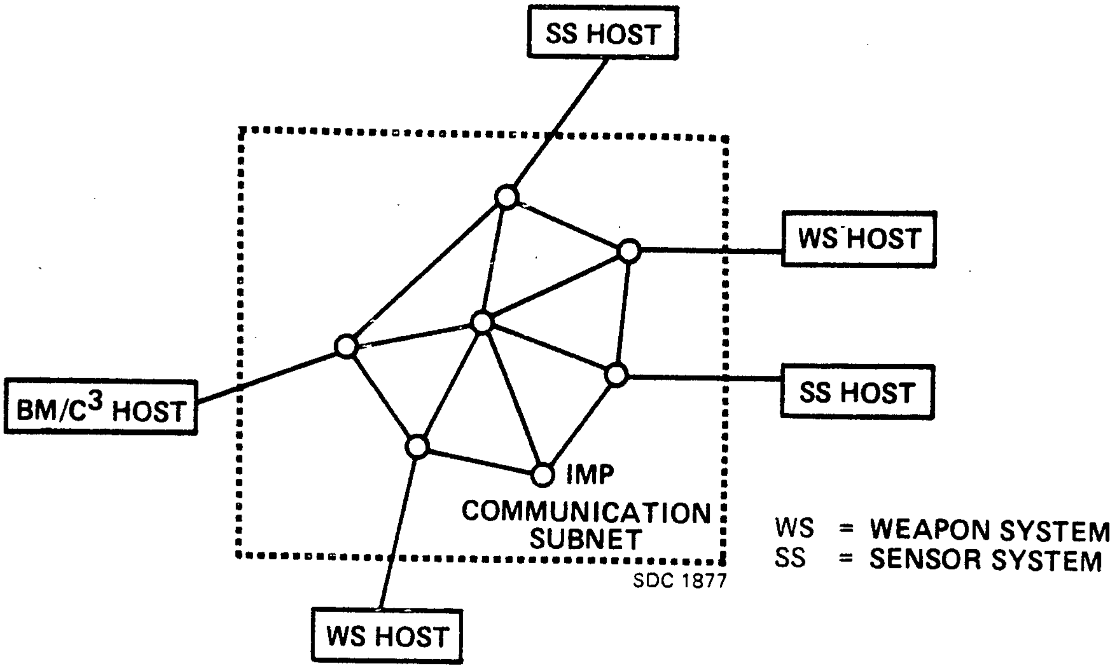


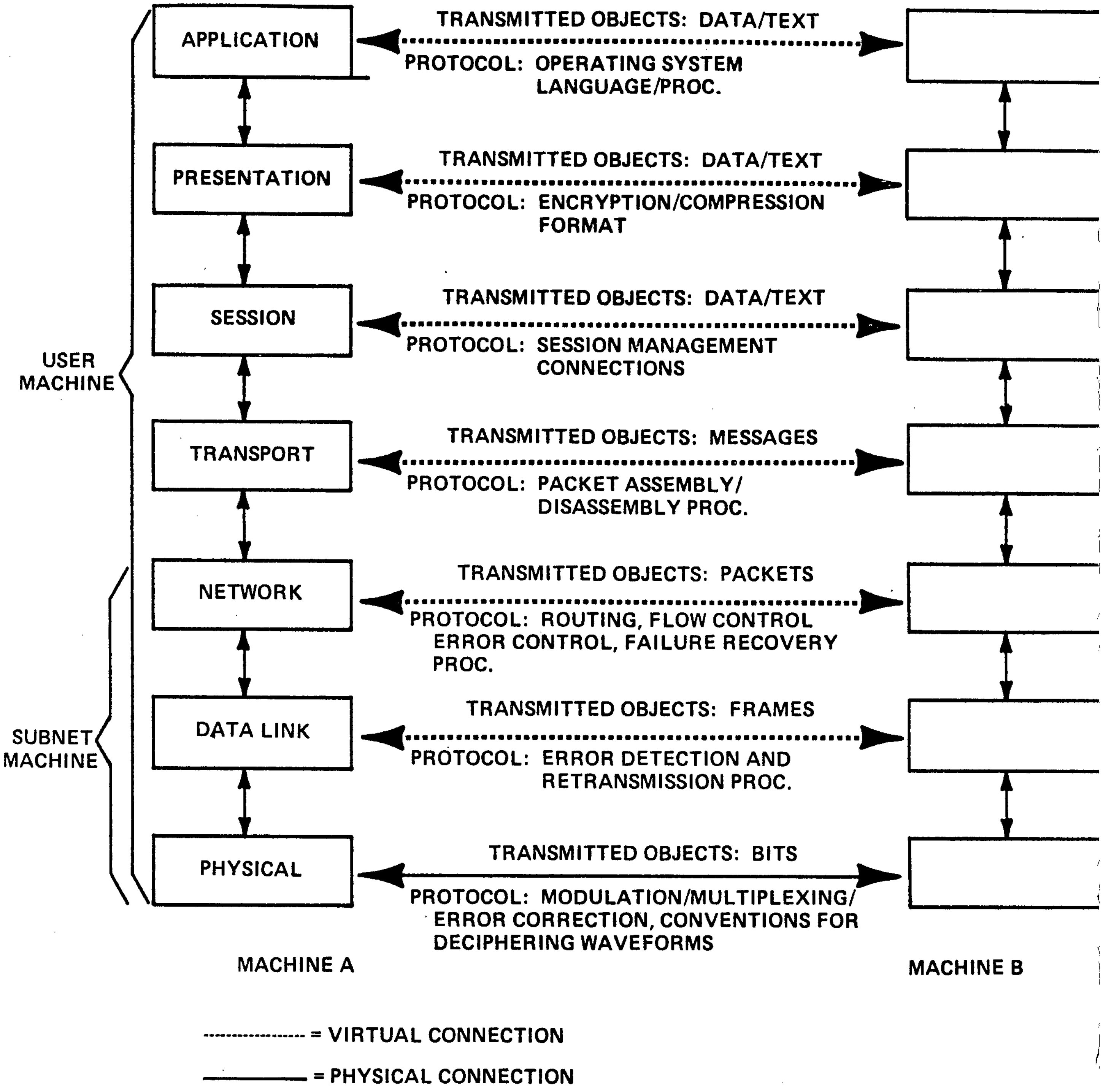
Figure 10-1. Data Communication Network Visualization

To place our discussion in this section in perspective, we here give a brief overview of a general method of structuring the issues associated with design and operation of communication networks.

The Open System Interconnection (OSI) model [1] developed by the International Standards Organization (ISO) serves as the starting point for the development of a network architecture. This model (Figure 10-2) consists of seven layers of which only the lower three (physical, data link, and network) are relevant for the communication subnetwork nodes. The other four layers (transport, session, presentation, and application) reside in the machines of users (hosts) that are connected to the subnetwork. Each layer has a set of functions which collectively provide a service to the immediately higher layer, and a set of protocols which implement these functions. The focus of this discussion is the network layer, so we will concentrate on the relevant functions and protocols associated with this layer.

The service provided by the physical layer to the data link layer is to convert an analog asynchronous channel into a synchronous (but unreliable) bit channel. The service provided by the data link layer is to convert an unreliable synchronous bit channel into a reliable frame channel across the two ends of each communication link. However, the network layer can provide two alternative types of service to the transport layer: virtual circuit service, whereby the network layer delivers all packets correctly, without losses or duplicates, and in the proper order, and datagram service where none of the above is guaranteed.

The terms virtual circuit and datagram are also used in a different context which has an engineering character and relates to the internal structure of the communication subnetwork. Thus, we say that the network layer uses virtual circuits if, given a user pair conversation, every packet of that conversation goes through the same sequence of links as it travels through the communication subnetwork. This sequence is known as a virtual circuit (VC for short). It is established at the time the user pair conversation is set up, and it is torn down when the conversation is terminated or if a communication link used by the VC fails in which case the VC must be rerouted.


APPLICATION
TRANSMITTED OBJECTS: DATA/TEXT
PROTOCOL: OPERATING SYSTEM LANGUAGE/PROC.
PRESENTATION
TRANSMITTED OBJECTS: DATA/TEXT
PROTOCOL: ENCRYPTION/COMPRESSION FORMAT
SESSION
TRANSMITTED OBJECTS: DATA/TEXT
PROTOCOL: SESSION MANAGEMENT CONNECTIONS
USER MACHINE
TRANSPORT
TRANSMITTED OBJECTS: MESSAGES
PROTOCOL: PACKET ASSEMBLY/ DISASSEMBLY PROC.
NETWORK
TRANSMITTED OBJECTS: PACKETS
PROTOCOL: ROUTING, FLOW CONTROL ERROR CONTROL, FAILURE RECOVERY PROC.
SUBNET MACHINE
DATA LINK
TRANSMITTED OBJECTS: FRAMES
PROTOCOL: ERROR DETECTION AND RETRANSMISSION PROC.
PHYSICAL
TRANSMITTED OBJECTS: BITS
PROTOCOL: MODULATION/MULTIPLEXING/ ERROR CORRECTION, CONVENTIONS FOR DECIPHERING WAVEFORMS
MACHINE A
MACHINE B
---------------- = VIRTUAL CONNECTION
———————— = PHYSICAL CONNECTION

SDC 1881

Figure 10-2. Open System Interconnection Model

Alternatively we say that the network uses datagrams if it is possible for two successive packets of the same user pair conversation to travel via different routes through the communication subnetwork. What is confusing about this terminology is that through the use of appropriate protocols it is possible for the subnetwork to provide virtual circuit service while using datagrams. Indeed, the ARPANET operates this way. If the network must provide virtual circuit service it is best to use virtual circuits since then the engineering implementation of the virtual circuit service reliability requirements are somewhat easier. However, more communication overhead is required to recover from link or node failures when virtual circuits are used rather than datagrams. Therefore it may be advantageous to use datagrams in an environment where there are frequent link and node failures.

Aside from management of virtual circuits or datagrams, the main functions of the network layer are end-to-end error control, routing, flow control and failure recovery. We provide a brief overview of the issues surrounding each of these.

10.1.1 End-to-End Error Control

Even if individual communication links are made perfectly reliable through the use of error detection and retransmission protocols, it is possible for packets to get lost inside the network due, for example, to a node being destroyed. For this reason it may be important to implement an acknowledgement system whereby a packet is buffered at its origin and retransmitted if its receipt is not acknowledged by its destination within a specified period of time. Some of the issues involved in end-to-end error control are to design the system in such a way that unnecessary retransmissions do not occur frequently, and to ensure that packets transmitted more than once are accepted at their destination only once.

10.1.2 Routing

Once a packet is accepted inside a data network it will travel along a sequence of links (a route) from origin to destination. The routing protocol is the procedure by which the packet's route is determined. In a network using datagrams where two packets of the same user pair may travel along different routes, a routing decision must be made at each node reached by a packet regarding the next outgoing link to be used. In a virtual circuit network, a routing decision is made at the time each virtual circuit is set up. The routing algorithm is used to choose the communication path for the virtual circuit. The main performance issue in routing is how to distribute traffic within the network in a way that no link gets overloaded. The effect of good routing is to reduce queuing delays at bottleneck links and to allow the network to handle more traffic than would otherwise be possible.

10.1.3 Flow Control

There are times when the externally offered load is so large that the network cannot possibly handle it even with optimal routing. If no measures are taken to restrict the entrance of traffic into the network, queue sizes at bottleneck links will grow indefinitely and eventually exceed the buffer space at the corresponding nodes. Packets arriving at these nodes will have to be discarded and later retransmitted (due to data link error control protocols), thereby wasting communication resources. The net effect will be throughput degradation and potentially intolerable delay inside the network. The function of the flow control algorithm is to prevent a portion of the offered traffic from entering the network in order to avoid this type of congestion. There are three main issues in flow control -- striking a good compromise between throttling users and keeping average delay per message at a reasonable level, maintaining fairness for all users while preventing a portion of the offered traffic from entering the network, and preventing throughput degradation and deadlock due to buffer overflow.

10.1.4 Failure Recovery and Topology Updating

When a network is undergoing changes, an algorithm is needed that broadcasts up-to-date information regarding the up-down status and the communication capacity of each link to the entire network. The issues that require attention in such an algorithm are far from trivial since information regarding link status must be communicated over links that are themselves subject to failure. Furthermore, one has to consider the case where the network becomes disconnected in which case it is impossible to keep the entire network informed of current link status. Additional issues include the number of messages required (communication complexity) and speed (time complexity) to successfully update the topological database throughout the network.

10.2 CANDIDATE APPROACHES

As described above, the purpose of the network layer is to provide a virtual packet channel connecting two communicating processes through the backbone network. This goal is generally accomplished by means of several algorithms, the most important of which are:

a. The end-to-end error control algorithm which ensures that node failures do not result in packet losses or duplicate deliveries

b. The routing algorithm which guides packets from origin to destination through the backbone network

c. The flow control algorithm which reduces the flow of traffic into the network when congestion sets in, and

d. The topology updating algorithm which guarantees that information regarding link failures and repairs is disseminated correctly throughout the network.

In what follows, we describe several alternative approaches to design of network layer algorithms. These approaches are suitable for developing algorithms which share three key characteristics:

a. They are distributed, thereby enhancing network survivability

b. They are adaptive to variations in network connectivity and communications loads, and

c. They are (near) optimal, so that communication resources can be used efficiently and communications delays can be minimized.

10.2.1 End-to-End Error Control

The need for end-to-end error control depends upon whether the network layer offers virtual circuit or datagram service to the transport layer. If datagram service is provided, end-to-end error control must be implemented (if at all) in the transport layer or above, i.e., by software running in hosts rather than IMPs. If virtual circuit service is provided, then, end-to-end error control algorithms are needed. Unfortunately, these algorithms account for a significant portion of the complexity and overhead associated with network layer protocols.

Protocols for end-to-end error control have been successfully implemented. Of course, there are design choices to be made in the detailed specification of these protocols that would need to be investigated during the design of a communications network for SDI. However, a more basic issue is whether or not end-to-end error control is worthwhile to implement giving its cost in overhead and complexity. This question canot be answered independently from the requirements placed on the communications network.

For example, a large portion of the message traffic in a BMD system will be reports of target variables from sensor systems. If these reports are time-stamped, it may not matter if two reports from the same sensor on the same target occasionally arrive out of time sequence. Likewise, if the reporting rate is sufficiently high, it may not matter too much if a target update message is lost occasionally.

10.2.2 Routing

Most existing data networks employ shortest path routing. By this we mean that each communication link is assigned a positive length and each packet is routed along a path that has minimum length among those connecting its origin and destination nodes. The length of each link may depend on the current congestion level as in the current ARPANET* algorithm [4], thereby building into the algorithm a tendency to select relatively uncongested paths. Unfortunately, the algorithm itself can influence the pattern of congestion within the network with a feedback effect resulting that can cause oscillatory behavior. This phenomenon has been observed in the ARPANET and is analyzed in detail in [3]. Another major drawback of shortest path routing is that at any given time, it utilizes only one path per origin-destination pair even if other congestion-free paths are available. As a result, throughput may be unnecessarily limited, and average delay per packet may be unnecessarily large.

Because efficiency and maximum utilization of communication capacity are essential under conditions of stress likely to arise in the context of battle management, sophisticated routing algorithms that out-perform the shortest path method are desirable. There are three candidate approaches that we will discuss:

a. Optimal quasistatic routing
b. Optimal dynamic routing
c. Limited flooding for high priority messages under conditions of stress.

These approaches are not mutually exclusive and ways of effectively combining them should be investigated. The shortest path approach should also be considered to provide a yardstick by which the effectiveness of other methods can be measured.

*In the ARPANET, a single packet message with 968 bits of data has 240 bits of protocol information attached to it. In addition, a variety of special purpose packets with <u>no</u> data content are needed to allocate buffer space, distribute routing information, etc.

10.2.2.1 Optimal Quasistatic Routing Algorithms

Algorithms in this category are based on the assumption that the input traffic to the network is quasistatic. By this we mean a situation where the offered traffic statistics for each origin-destination pair change slowly over time and, furthermore, individual offered traffic sample functions do not exhibit frequent large and persistent deviations from their averages. A typical quasistatic network is one accommodating a large number of interactive processes for each origin-destination pair and in which the law of large numbers approximately takes hold. In such an environment, it is valid to base routing decisions on average levels of traffic input which can be estimated from past history measurements.

Quasistatic routing is based on a static mathematical programming formulation. Traffic flowing into the network is modeled by assuming that for each ordered pair of distinct nodes in the network (a so-called origin-destination or OD pair) there is a constant average arrival rate in data units/sec (data units may be bits, frames, packets, etc.) of message traffic that must be routed from the origin to the destination node. Depending on the route or routes chosen for each OD pair in the network, the average flows (also in data units/sec) on the various links of the networks will vary. Since the average delay on any link is a function of the average flow on that link and its capacity (also in data units/sec), the average delay that message traffic experiences in the network is a function of the routing. Quasistatic routing algorithms seek to choose the routing to minimize the average delay.

An example of a quasistatic routing algorithm is the gradient projection algorithm [4], [5]. Extensive computational experience has verified that this algorithm typically converges to an optimal solution in very few iterations. The fast convergence can be attributed to the use of second derivatives in a manner that is reminiscent of Newton's method. Through the automatic scaling provided by the use of second derivatives, the iteration does not depend on knowledge of the input traffic rates of various OD pairs and automatically adapts to any level and pattern of traffic input. As a result, it provides a routing algorithm that operates near-optimally for all possible network and load configurations.

A second important aspect of this algorithm is that it is well suited for distributed operation. Each node can execute iterations of the algorithm asynchronously and independently from any other node. What each node requires is knowledge of the average flow on each link of the network, and the capacity of that link. This information can be broadcast throughout the network either by flooding (as currently done in the ARPANET), or through the algorithm that keeps the nodes informed of recent changes in the network topology (see the sequel).

10.2.2.2 Dynamic Routing Algorithms

Optimal quasistatic routing tries to optimize the steady state distribution of packet arrival rates at the transmission queues, but pays no attention to the transient levels of traffic in various parts of the network. In cases where, due to some unpredictable event, there is a large queue buildup in some parts of the network, an optimal quasistatic algorithm may be slow in recognizing the problem and alleviating the congestion. This is because the time between routing updates is typically rather large (say in the order of several seconds to minutes) for a quasistatic routing algorithm. One may try to speed up the updating rate, but then a problem arises in that link flow rates cannot be accurately measured by a time average if the averaging interval is too small. It is not known to what extent this is a significant difficulty--research on this matter is in progress (see Tsitsiklis and Bertsekas [6]). Thus, due to these limitations of quasistatic routing algorithms, a dynamic routing algorithm that bases routing decisions on the current state of queues in the network and tries to optimize the transient as well as the steady state congestion level based on dynamic predictions of load transients would clearly be desirable if it could be practically implemented.

The major candidate algorithm for dynamic routing is based on a problem formulation due to Segall and Moss [7]. The original algorithm of [7] was greatly simplified and improved by Hajek and Ogier [8]. This algorithm is suitable for distributed implementation. Each node receives information on the current network topology and the queue state of every other node, performs some calculations, and directs traffic to the appropriate links.

There are several difficulties to overcome before this algorithm can be made truly practical. First, because the routes for each destination can change fast, it is difficult to apply this algorithm in a virtual circuit network -- indeed, any dynamic routing algorithm is very difficult to implement in such a network because of the delays associated with disconnecting and reestablishing virtual circuits. Second, the queue state information needed for operation of the algorithm is subject to communication delays which may be substantial, (e.g., where satellite links with long propagation delays are involved). Despite these difficulties, dynamic routing may offer important advantages in the BMD context, where the dynamics of network load and structural changes can be anticipated to a certain extent due to predictable satellite motion and ballistic missile trajectories, and is worth evaluating as an alternative or supplement to optimal quasistatic routing.

10.2.2.3 Limited Flooding Algorithms

In a data network that is subject to link and node failures, there is no way of knowing whether a packet transmitted along a route thought to be intact will indeed reach its destination. For example, some link on the route may have already failed (or may fail while the packet is in transmit), but this information may not yet have reached the origin node of the packet. For this reason, it may be necessary for some types of traffic to implement an end-to-end acknowledgement scheme whereby a packet is retransmitted by the origin node if a negative acknowledgement is received or a positive acknowledgement is not received within a specified time interval. The resulting delays may be substantial, and can become even larger if virtual circuits are used and the routing algorithm requires some time to reestablish virtual circuits that have been affected by link failures. For this reason, it may be advisable to use a routing scheme with built-in redundancy for those packets for which there is a strict limit on the delay that can be tolerated.

The most effective method to ensure that a packet reaches its destination in the fastest possible time is to use flooding. By this we mean a scheme whereby the originating node sends the packet to its neighbors, the neighbors send the packet to their neighbors, and so on. The algorithm terminates with a number of packet transmissions which is between L and L/2 (where L is the number of links in the network) by virtue of a scheme that numbers packets and prohibits a node from transmitting a packet twice or sending a packet back to a node from which the packet was received. Unfortunately, the number of required transmission is still excessive, and is the same regardless of the distance in number of hops between origin and destination.

In order to reduce the number of packet transmissions, it is possible to consider flooding on a limited basis whereby a packet travels along several paths to its destination, but does not go over every network link. This can be done by considering an acyclic directed graph (ADG) that is rooted at each destination as shown in Figure 10-3. Here, every link has a direction associated with it, and flooded packets for the corresponding destination are required to travel in the specified link direction only. It is easily seen that as long as every node (except for the destination) has at least one outgoing directed link, there will be at least one available path for each node

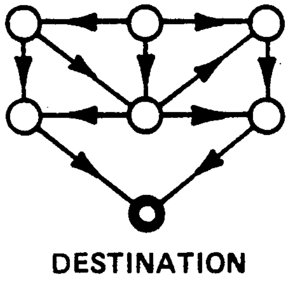


Figure 10-3. Acyclic Directed Graphic (ADG)

to the destination. Furthermore, typically there are more paths available to nodes that lie far from the destination than nodes that lie close to the destination. This is consistent with the idea that there is more need for redundancy for long paths that are more susceptible to topological changes.

A good method for constructing acyclic graphs of the type described above is by means of a shortest path method. If N_i is the minimum number of hops from node i to the destination, we orient each link from the node of larger distance to the node of smaller distance, and break ties on the basis of some criterion that maintains acyclicity of the graph (for example, in case of a tie orient a link from the higher to lower node identity number).

There are particularly simple distributed asynchronous algorithms for maintaining such an ADG in the presence of topological changes. An example is [9]:

FULL REVERSAL METHOD: Each node (other than the destination) that has no outgoing link reverses the directions of all its incoming links.

Figure 10-4 provides an example of the sequence of successive iterations of this algorithm. The nodes that reverse at each iteration are denoted by R.

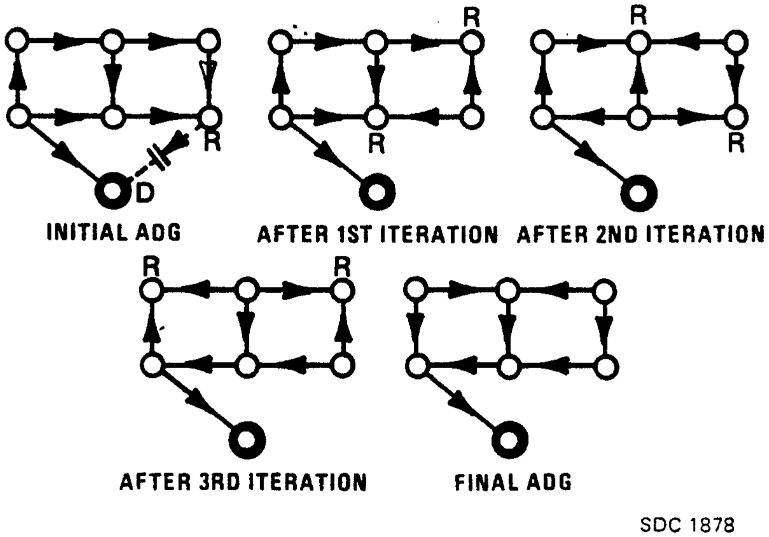


Figure 10-4. Full Reversal Method

The algorithm starts with an ADG where some node has no path to the destination, generates a sequence of ADG's, and terminates with an ADG in which every node has a path to the destination.

The algorithm above requires a considerable amount of communication to converge. There are, however, other similar but more sophisticated algorithms that require much less communication [9].

All these algorithms can be shown to work correctly. Furthermore, it can be shown that the direction of any link between two nodes that have a directed path to the destination in the initial ADG will not be reversed. Therefore, the reversal process will be localized within the portion of the network that lost all its paths to the destination. This property makes these algorithms attractive for situations where there are massive link failures.

10.2.3 Flow Control

The most popular flow control methods are based on windows, either end-to-end (as in the ARPANET) or link-by-link (as in the TYMNET). By a window W between points A and B in the network, we mean a restriction on the number of packets that A has transmitted to B, but has not yet received acknowledgements for. When the number of such packets reaches the limit W, node A does not transmit any further packets until a new acknowledgement returns. In end-to-end window flow control, the points A and B are the backbone network entry and exit nodes for a session. In link-by-link flow control, the points A and B are the head and tail nodes of a link along a session's path. The main advantage of window schemes is that they react fast to congestion (within one round trip) by throttling packets when acknowledgements are slow to return.

On the other hand, window schemes also have some serious disadvantages when implemented with fixed window size. One would like flow control to be inactive when the network is uncongested. To accomplish this, the window size multiplied with the transmission time of the packet must exceed the least possible forward delay of a packet plus the least possible return delay of the acknowledgement. This means that, if the network includes satellite links involving large propagation delay, window sizes should be large. On the other hand, with large window sizes, it is more difficult to control congestion because windows from several sessions may pile up at a congested node and create a queue size that is proportional to the typical session window size. Thus, the main difficulty with windows is greatly magnified in networks involving satellites. For example, any session going through a 1 M bit/s link, using 1000 bit packets, with 0.25 s propagation delay requires a window of a least 500 packets in order to achieve unimpeded transmission under light traffic conditions. If several of these windows can pile up at some node in the network, intolerable delays will occur. For this reason, it appears necessary to adopt link-by-link in addition to or in place of end-to-end windows in a network involving satellites. Even with link-by-link windows, it is possible to have large delays at some nodes unless there is a strategy to reduce the window sizes where delays get excessive.

Unfortunately, it is quite difficult to adjust directly window sizes in an intelligent way in response to congestion. However, Gallager and Golestaani [10] have proposed a promising approach in which the processes of routing and flow control are combined. Their formulation involves the formulation of a joint optimization problem in which both the routing variables and the OD pair input rates are computed. This approach has been refined and techniques developed for converting the flow rates computed by the combined routing and flow control algorithm into window sizes [11].

10.2.4 Topology Update Algorithms

As we approach the subject of disseminating topological update information, we must first recognize that it is impossible for every node to know the correct network topology at all times. Therefore, the best that we can expect from an algorithm is that it can cope successfully with any finite number of topological changes within finite time. By this we mean that if a finite number of changes occur up to some time and no other changes occur subsequently, all nodes within each connected portion of the network should know the correct status of each link in that portion within finite time.

In our subsequent discussion, we will assume the following:

a. Network links preserve the order and correctness of transmissions, i.e., there is a Data Link layer protocol that works correctly. Furthermore, nodes maintain the integrity of messages that are stored in their memory.

b. There is a protocol for bringing up links and declaring them down.

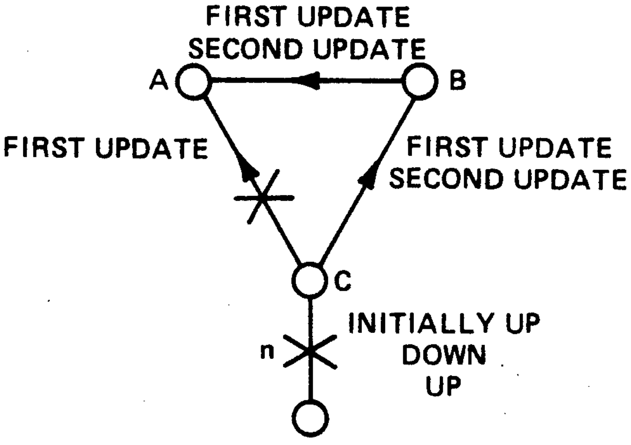
c. A permanent link status change will be detected by both end nodes of the link, although not necessarily simultaneously. By this we mean that there is a time interval T such that any link with end nodes A and B that is declared to be down (up) by A will, within time T, be either declared up (down) by A or be declared down (up) by B.

In rare occasions, assumption a. is violated because damaged data frames may pass the error detection test of Data Link Control (a very low probability event), or because a packet may be altered inside a node's memory due to hardware malfunction. Therefore, we must be careful to ensure that a rare error of that type does not have long lasting detrimental effects on the overall topology update system. Note that we will not assume that the network will always remain connected. In fact, the topology update algorithm together with the protocol for bringing up links should be capable of starting the network following a reset.

Two algorithms should be considered: (a) The ARPANET algorithm where topological change information (indeed average delay information on each link) is flooded throughout the network periodically as well as on a contingency basis [2]. (b) The Shortest Path Topology Algorithm (SPTA) that was recently developed by Spinelli [13], and that avoids some of the difficulties associated with the ARPANET algorithm.

In the ARPANET algorithm, each node sends a packet containing the status of all its outgoing links (referred to as its local topology) to its neighbors upon detecting a change in the status of any one of these links. The neighbors send this packet to their neighbors and so on with the exception that a node does not send back a packet to a node from which it has already received it. This algorithm, called flooding (or broadcasting), works well for a single topological change, but fails in its pure form when there can be multiple changes.

An example is shown in Figure 10-5 that illustrates the fundamental issue in the topology update problem: the difficulty of distinguishing between old and new information. In this example, link n is initially up, then it goes down, then up again. If the two updates travel on the path CBA faster than the first update travels on the link CA and link CA fails after the first update, but before the second update travels on it, then the last message received by A asserts that link n is down, while the link is actually up.



SDC 1879

Figure 10-5. Example Where Flooding Fails

In the ARPANET, this problem is resolved by sending update packets at regular intervals and by marking them with a sequence number. The two main disadvantages of the ARPANET algorithm are: a) a large amount of regular overhead, b) potentially large delays in updating the topology information of disconnected portions of the network. The ARPANET algorithm has also some additional irritating difficulties such as the problem of wraparound of the update sequence counter. For a discussion of these difficulties together with possible remedies, we refer to the paper by Perlman [12]. Nonetheless, the ARPANET algorithm has proved to be a successful protocol in practice (in a peaceful environment), and is a prime candidate for consideration in the BMD context. A potential improvement in communication complexity would be to replace the flooding scheme with broadcasting along a spanning tree coupled with a distributed algorithm for constructing and maintaining the spanning tree [14], [15].

The SPTA avoids both disadvantages of the ARPANET algorithm in that it does not rely on regular updates to maintain correctness after two pieces of the network disconnect and later connect. Furthermore, update packets do not carry sequence numbers with the attendant wraparound problems. Roughly speaking, the SPTA resolves the problem of distinguishing old from new information by assigning a "reliability index" to each piece of information received at a node. The node believes the "most reliable" information regarding the status of a link. The "reliability" of existing information is reevaluated when new information comes in, and existing information is discarded once it is either superceded by new information or is proven "unreliable."

It is shown in [13] that the algorithm works correctly and that it handles single link failures with the same amount of communication overhead as the ARPANET algorithm. However, the SPTA is entirely event-driven, and does not require the substantial overhead associated with the regular periodic updates of the ARPANET algorithm.

10.3 RECOMMENDED APPROACH

To develop the required networking concepts, the following approach is recommended:

10.3.1 Baseline Assumptions

Baseline assumptions must be made concerning threat systems, Systems Architectures, and BM/C^3 Architectures to provide a realistic context for the networking research. At least two different Systems Architectures should be examined, one emphasizing space-based assets and one emphasizing ground-based (including pop-up) assets, along with corresponding BM/C^3 Architectures. It should be emphasized that detailed analyses are not required; it is just necessary to record a set of baseline assumptions that are in reasonable agreement with current SDI thinking.

10.3.2 Communication Requirements

After baseline assumptions are identified, communications requirements must be determined for the various combinations of Systems and BM/C^3 architectures. Connectivity requirements can be determined by examining the data flows between nonphysically colocated functions of the BM/C^3 Architecture. Message flow rates can be quantified based on estimates of message length and BMD System target loading. Message priority classes must also be determined. Just as was the case for the effort to define baseline assumptions, detailed analyses of communications requirements are not needed, but rough estimates must be available to guide the networking research.

10.3.3 Communications Architectures

For the various combinations of Systems and BM/C^3 Architectures, communications architectures must be derived. Communication relay nodes must be added to the System Architecture and link capacities adequate to satisfy communications requirements must be specified. It should be emphasized that the intent of this effort is not to specify an optimal communications architecture, but rather to develop a reasonable set of assumptions to provide a framework for the networking research.

10.3.4 Subproblem and Scenario Determination

Based on the foregoing analyses, a communications network subproblem should be defined, e.g., communications during late midcourse, that is representative of the overall problem. Assumptions should be made concerning destruction of network nodes and reduction of link capacities due to jamming and nuclear effects. As before, detailed analyses of environmental effects are not desired but simply reasonable approximations.

With the completion of the analyses described above, the development of algorithms and protocols can proceed.

10.3.5 End-to-End Error Control

End-to-end error control protocols must be designed. Analyses should be performed to quantify tradeoffs associated with packet length, timeout interval, use of negative acknowledgements, etc. The overhead associated with end-to-end error control should be quantified so that tradeoffs can be made at the applications level to determine if end-to-end error control is really required or if datagram service is adequate.*

10.3.6 Routing

Both static and dynamic routing algorithms should be developed for comparison, and the use of flooding for key message classes investigated. The algorithms must be evaluated by simulation (as described below).

10.3.7 Flow Control

Mathematical algorithms must be derived for flow control. The quantities computed by these algorithms must then be related to parameters that can be set in flow control protocols (e.g., window size). The algorithms must distinguish between message priority classes.

10.3.8 Topology Update Algorithms

Algorithms and associated protocols must be designed to provide the information on network topology (including link capacities) required for routing and flow control.

*This issue cannot be resolved based on networking considerations alone. Note that a hybrid approach - use of end-to-end error control only for certain critical message classes may well be optimal.

10.3.9 Simulation

A simulation should be designed to evaluate the end-to-end error control, routing, flow control, and topology updating algorithms and protocols developed. The simulation should accept time-varying message arrival rates and network topology (including provision for time-varying link capacities and destruction of nodes), and be capable of fully evaluating all the algorithms and protocols developed. However, physical and data link layer processes need not be simulated in detail, e.g., it is unnecessary to simulate the bit-by-bit clocking out of frames onto a channel. The simulation should be exercised on the network subproblem identified above and an optimal set of algorithms and protocols (including detailed parameter values) should be determined.

10.3.10 Interface to BM/C^3 Simulation

An interface to a BM/C^3 simulation at the Army Strategic Defense Command's Applications Research Center should be developed. This interface should compute message arrival rates and network topology for input to the communications network simulation.

10.4 REFERENCES

1. Tannenbaum, A. S., Computer Networks, Prentice Hall, NJ 1981.

2. McQuillan, J. M., I. Richer, E. C. Rosen, and D. P. Bertsekas, "ARPANET Routing Algorithm Improvements, Second Semiannual Report," Bolt, Beranek, and Newman, Inc., BBN Report prepared for ARPA and DCA, Oct. 1978.

3. Bertsekas, D. P., "Dynamic Behavior of Shortest Path Routing Algorithms for Communication Networks," IEEE Trans. on Aut. Control, Vol. AC-27, 1982, pp. 60-74.

4. Bertsekas, D. P., and E. M. Gafni, "Projected Newton Methods and Optimization of Multicommodity Flows," *IEEE Trans. on Aut. Control*, Vol. AC-28, 1983, pp. 1090-1096.

5. Bertsekas, D. P., and E. M. Gafni, "Projection Methods for Variational Inequalities with Application in the Traffic Assignment Problem," *Math. Programming Studies*, 1982.

6. Tsitsiklis, J. M., and D. P. Bertsekas, "Distributed Asynchronous Optimal Routing in Data Networks," *Proc. of 23rd Conference on Decision and Control*, Las Vegas, NV, December 1984, pp. 1133-1138.

7. Segal, A., and F. Moss, "An Optimal Control Approach to Dynamic Routing in Networks," *IEEE Trans. Auto. Control*, Vol. 27, 1982.

8. Hajek, B., and R. G. Ogier, "Optimal Dynamic Routing in Communication Networks with Continuous Traffic," *Networks*, Vol. 14, 1984.

9. Gafni, E. M., and D. P. Bertsekas, "Distributed Algorithms for Generating Loop-Free Routes in Networks with Frequently Changing Topology," *IEEE Trans. on Communications*, Vol. COM-29, 1981, pp. 11-18.

10. Gallager, R. G., and S. J. Golestaani, "Flow Control and Routing Algorithms for Data Networks," *Proc. 4th Int. Conf. Computer Communication*, Atlanta, GA, October 1980.

11. Gafni, E. M., *The Integration of Routing and Flow Control for Voice and Data in a Computer Communication Network*, Ph.D. Thesis, Dept. of EECS, M.I.T., August 1982.

12. Perlman, R., "Fault-Tolerant Broadcast of Routing Information," *Computer Networks*, Vol. 7, 1983, pp. 395-405.

13. Spinelli, J. M., Broadcasting Topology and Routing Information in Computer Networks, M. S. Thesis, Dept. of EECS, M.I.T., May 1985.

14. Gallager, R. G., P. A. Humblet, and P. M. Spira, "A Distributed Algorithm for Minimum-Weight Spanning Trees," ACM Trans. on Programming Languages and Systems, Vol. 5, 1983, pp. 66-77.

15. Humblet, P. A., "A Distributed Algorithm for Minimum-Weight Directed Spanning Trees," IEEE Trans. on Comm., Vol. COM-31, No. 6, 1983, pp. 756-762.

SECTION 11 - COMMUNICATIONS

Frederic Weigl
Rockwell International

11.1 INTRODUCTION OF ISSUES

Communications for a ballistic missile defense (BMD) system such as SDI are highly diverse, requiring interconnection of ground, airborne, and satellite assets. Connectivity may be achieved using point-to-point communications links or broadcast networks (such as from a high orbit sensor satellite to low orbiting satellites). Communications media will range from cable and fiber optics for ground-based assets, through the radio spectrum from VLF/LF for interconnection to existing networks of the World-Wide Military Command and Control System (WWMCCS) and Minimum Essential Emergency Communications Network (MEECN) to SHF or millimeter wave for satellite communications, and possibly to optical communications for satellite cross links. Accurate data rate requirements can be developed only within the context of a specified network architecture and defined data exchange requirements. However, initial estimates based on high level consideration of data flow within a BMD system by the Fletcher Commission and others indicate maximum data rates below 10^7 or 10^8 bits/second [1].

The SDI communications network will be large, interconnecting up to several hundred space assets, ground facilities and airborne elements dispersed around the globe. Many of the communications nodes will be in constant relative motion, complicating network control and acquisition/synchronization problems within the communications system. As a result of this mobility and the need for the communications system to adapt to the loss of nodes and links during an engagement, substantial network complexity may result. Communications links must operate over long propagation paths (up to 170,000 km for satellite-to-satellite links [1]) in a potentially hostile environment of electronic countermeasures (ECM) and nuclear effects. In addition, it is unrealistic to consider that large blocks of radio spectrum will be reserved from peacetime encroachment for the possible use of SDI, so the communications system must operate in an environment of potentially high terrestrial background noise.

Probably the biggest issue facing the design of a communications system for the SDI is that of developing a robust, adaptive, distributed network architecture capable of responding quickly to highly dynamic changes in traffic load and network topology. The network must be self-healing, using dynamic reconfiguration to route around lost links and/or destroyed assets. The communication network must maintain its ability to gather and disseminate necessary status information and control messages to support its operation in the face of rapidly changing traffic load and network configuration. These topics are discussed in Section 10 and are not considered further here.

A related and interacting issue with communications networking is that of providing adequate security in the distributed, adaptive type of network likely to evolve for SDI. The need to provide security of the BM/C^3 functions is clear in a ballistic missile defense system. The large size and diversity of the network, the possibly large number of personnel having access to it, and the required network adaptability/reconfigurability complicate the security problem. Development of network security requirements, concepts, and approaches must be addressed early in the development of the overall BM/C^3 system architecture (refer to Section 11.2 for discussion of this topic).

Reliable and timely communications are of primary importance in SDI BM/C^3. Although jamming is not considered to be a particularly serious threat because of the achievable directionality of satellite-to-satellite links and the fact that communications among and between ground and airborne assets and satellites is primarily over defense-controlled territory [1], communications links must nonetheless provide a measure of ECCM capability to allow reliable operation in the presence of spaceborne and terrestrial jammers and other countermeasures and background interference. Error control and correction technology is important in this environment. Use of such techniques can improve the quality of distributed data bases within the system and reduce requirements for retransmissions. Coding techniques can also be used to reduce the amount of data to be transmitted, thereby improving network throughput and reducing message delay. These issues are discussed in Section 11.3.

Nuclear effects constitute a serious threat to communications, especially for ground and airborne assets. Nuclear disturbance of the atmospheric environment can have severe effects on the propagation of radio signals, disrupting communications for extended periods. Blast, electromagnetic pulse (EMP), and radiation effects from leakage warheads must also be understood and mitigated by special design approaches before the BM/C^3 system can be considered viable. Nuclear effects are particularly important during early phases of deployment when leakage can be high. These issues are discussed in Section 11.4.

A requirement also exists for intercommunication between the SDI BM/C^3 system and offensive systems if coordination is to be achieved between defensive and offensive operations. Operational requirements for such interface points between defensive and offensive networks must be defined early to be accommodated in defensive BM/C^3 system design and to avoid unnecessary disruption or duplication of existing offensive networks. This issue is discussed in Section 11.5.

A number of additional areas of technology require development for effective deployment of an SDI BM/C^3 system. Although initial estimates of required data rates appear within the current state-of-the-art, the availability of higher speed, hardened components for both transmission and signal processing can substantially improve the cost-effectivity and performance of the communications system. However, substantial effort is already underway to address these requirements. Laser and millimeter wave links are desirable for satellite communications because of their narrow beam widths and high data rate capabilities. Technology issues here include the reliability of available components for use in space assets. Improvements in pointing and tracking accuracies of antennas for such links will allow better advantage to be taken of their directionality to counter jamming threats and to reduce power requirements. Work performed in directed energy weapon development should be applicable to this problem also. Because these technologies are already receiving substantial attention or are of secondary importance, they are not discussed further here.

11.2 NETWORK SECURITY

11.2.1 Problem Identification

It is conceivable that an adversary would attempt to exploit every potential vulnerability of an SDI communication system in an attempt to disrupt the command and control necessary to achieve an effective defense. That is, if a first strike is ever attempted by the enemy, it is not reasonable to assume that the BM/C^3 system is not going to be attacked. If communications can be disrupted or reduced effectiveness achieved by destruction of critical BM/C^3 nodes, jamming of RF links, injection of false information into the communication system, or by the identification of the critical BM/C^3 nodes or other critical elements through traffic flow analyses, an adversary will use all means available to him to accomplish this. The nature of the SDI system suggests that the threat is likely to emphasize passive attack techniques during the development, deployment, and peacetime operation of the system, and mainly consist of active attacks during hostilities. Passive attacks are aimed at extracting information from the system through deciphering and analyzing message contents, addressing information, and traffic volume. Active attacks are aimed at disruption of communication and are likely to consist of destruction, jamming, and injection of false information. The critical command and control links for SDI BM/C^3 must be protected against exploitation and disruption. Unauthorized disclosure of data must be prevented, both to intruders and valid network users. Data authentication measures are necessary to prevent the use of erroneous data, where unauthorized modification of data is caused by either mechanical error or malicious sources. Denial of service must be possible, and automated recovery must help maintain a minimum level of service through the communications net.

Development of a security policy for SDI BM/C^3, that is the set of rules governing how the system manages, protects and distributes sensitive information, must consider the following factors:

- The large size and distributed nature of the network
- The potentially short duration and highly dynamic nature of an engagement
- The varying levels of classified information present
- The probable existence of multiple communities of interest having different information access needs and security requirements
- The relative inaccessibility of space based nodes
- The probable need for decentralized network control.

11.2.2 Present Technology

A communication system is said to be trusted if it can be relied on to enforce the security policy of the system. The concept of a reference monitor has been defined to be that mechanism within a trusted system which enforces the security policy by controlling all accesses to sensitive data. The implementation of the reference monitor in any given communication system can vary depending on the level of trust of the network hosts, the operating mode of the hosts, and the threats to the network media.

In a trusted host, the reference monitor may be implemented by a security kernel which mediates all attempts by users to access classified data. The kernel will enforce DoD mandatory access control by applying real or implicit access control according to clearance levels of subjects (e.g., users). In a communications system which is operating in multilevel mode (i.e., not all users are cleared to the highest level of data in the system), unless all of the hosts are trusted to operate in multilevel mode, the network itself must perform the reference monitor function. In this case, the network would be required to provide labels on information transmitted over the net or implicit labels associated with virtual circuits. The network will authenticate each connection prior to establishing it. These checks can be performed on a host-to-host basis or on a process-to-process basis.

A network can use various mechanisms to implement its reference monitor function. The principal means of protecting data from compromise or modification is by end-to-end encryption (E^3). E^3 implements a virtual, single level subnetwork between subject-object pairs. It provides a means of operating multiple communities of interest, separated cryptographically from each other. One means of applying E^3 is through the use of Internet Private Line Interfaces (IPLI) [1]. The IPLI and its associated crypto device are positioned between the host computer and the network. The IPLI provides the access control by checking the validity of the destination in its tables before sending data to the network. The IPLI provides for isolating communities of interest at a specific sensitivity level. The disadvantage of the IPLI is that assignments to communities are static and cryptographic keys must be manually distributed and loaded at each site. Also, hosts can't communicate with hosts outside their community.

Work in recent years to improve E^3 systems has been aimed at developing techniques for remote key distribution. This would allow more flexibility in authorized access; i.e., host-to-host, process-to-process, or per connection individualized keying. This type of network needs an access controller and key distribution center in addition to its E^3 devices. The access controller mediates all requests by E^3 devices to allow communication. If the connection is authorized, the key distribution center generates a key for use by the communicating parties and ships it to them under their own master keys.

11.2.3 Approach

The security requirements for the SDI communications system must be analyzed, and the security architecture of the system must be determined based on those requirements; e.g., threats, sensitivity of data, perishability of data, clearance levels, trusted vs. untrusted hosts. Because of the critical functions of weapon control, the varying sensitivity levels of data, the widespread network, and the nature of the threats, it is expected that BM/C^3 networks will need to be trusted to control access among users. End-to-end encryption will probably be used to protect the data, as well as to separate the data on the network into communities of interest. The need for the network to be dynamically reconfigurable rules out any encryption scheme which calls for manual distribution of keys. The techniques for remote key distribution and network access control need to be applied to SDI networks. Some of the technology being developed in programs such as Blacker should be applicable to BM communications. Blacker is using E^3 and remote key distribution to provide a multilevel secure packet switching system. Components which are being developed include a key distribution center, an access control center, and a front end E^3 device. Network security requires the integration of security mechanisms into the communications protocols. The draft network criteria [3] call for the specification of a trusted network's architecture in terms of a reference model, such as ISO-OSI layered protocols. Security mechanisms, such as security labels, applied to the protocols must be described. The correspondence between these security mechanisms and the security features employed in the trusted network components must be established.

11.3 CODING AND WAVEFORMS

11.3.1 Problem Identification

The development of coding for such applications as error control and data compression was motivated primarily by problems in communications. Coding, however, has many other applications including data protection in computer memories and on digital tapes and disks, and protection against circuit malfunction or noise in digital logic circuits. Codes have also been used for data compression and in the design of statistical experiments.

Applications to the SDI BM/C^3 system communications problems will be diversified. Digital data will be transmitted between terminals in the BM/C^3 system and between and among airborne and spaceborne platforms. Coding will be used to achieve reliable communication even when the received signal power is close to the thermal noise power, and as the electromagnetic spectrum becomes even more crowded with signals, coding will become extremely important because it permits communication links to function reliably in the presence of such interference. In the BM/C^3 system, coding will be essential to protect against intentional interference using ECCM techniques.

Many communication systems have limitations on transmitted power. For example, power is very expensive in communication relay satellites, and is expected to be an important issue in the design of the BM/C^3 system. Coding to achieve error control is an excellent way to reduce power needs or to operate over degraded channels because messages received weakly at their destination can be recovered correctly with the aid of the code. The waveform used to encode the data will provide error protection from noise as well as intrusion protection from jammers. It can also reduce the probability of intercept of the transmissions. The power of the waveform is determined by the degree of protection provided to the data. In a highly dynamic network of widely separated and constantly moving nodes such as envisioned for SDI BM/C^3, link acquisition and synchronization issues may place constraints on waveforms to be used. Rapidly time-varying waveforms will seriously complicate link acquisition and synchronization when propagation times are long and time-varying as in links to orbiting satellites. Occasional packets of data may be lost because of synchronization problems, routing problems, loss or degradation of node and link assets, or any combination of these. Suitable error-control codes can protect against this loss with missing packets and potentially missing data deduced from known, correctly received packets.

Coding applications also will be important within the BM/C^3 system to reduce traffic load and resulting delays. Large data flows will exist between subsystems and elements within each subsystem. However, preliminary estimates of data flows indicate that, while significant in size, they should not pose a capacity problem. These data will be shared among multiple interconnected subsystems and terminals. Bus architectures, dedicated lines, and wideband optical links will be shared by numerous data and voice message transfers.

Waveform coding can be used to ensure proper performance. And, the perishability of information, a timeliness issue in any BM/C^3 system, can be minimized through proper application of coding to reduce data throughput requirements.

11.3.2 Present Technology

A vast amount of research has been devoted to coding and coding theories that can be used in the development of the BM/C^3 communication system architecture. Error detection and correction (EDAC) coding has been successfully applied to data communication systems like packet switching and the Joint Tactical Information Distribution Systems (JTIDS). Data compression coding techniques have been applied to numerous problems most notably for imagery transmitted between space platforms and ground-terminals. In addition, data compression techniques have been applied equally well in the commercial world for the successful transmission of video. Waveform technology has been used to protect digital communications from jamming and exploitation by an adversary. The most notable application is to JTIDS which uses a very powerful waveform coding technique for both EDAC as well as electronic counter countermeasures (ECCM). In addition, the waveform is frequency hopped.

Some research has been performed into developing missing data or correcting corrupted data that goes far beyond the capability of standard EDAC codes and coding techniques. Waveform multiplexing using orthogonal coding techniques [2, 3], such as Walsh Functions, can provide information about large blocks of message code that may have been lost prior to reaching a termination point in the network. In a packet-switched network, such techniques could be used to reconstruct a garbled packet which is beyond the capabilities of an EDAC code or to construct a totally missing packet.

While, at first glance, no evidence exists to suggest that the expected traffic load will demand extraordinarily large data rates to satisfy the delay and throughput requirements, coding will permit an additional flexibility that can enhance the survivability of the BM/C^3 network and facilitate the use of real-time protocols. Because the area of coding is well advanced, solutions to error control, waveform engineering and data combining problems are not beyond the present art.

11.3.3 Approach

There are two areas of research regarding the introduction of coding into the BM/C^3 system that need attention if the communications architecture for battle management and command and control is to be realized. First waveform issues including the application of spread spectrum techniques must be addressed. A standard waveform, MIL-STD 188-148, has been accepted by the government for use in ECCM applications, but it has severe limitations including the frequency hop rate that suggest new, more innovative ideas must evolve.

The second area is research into the use of coding for data compression and reconstruction. This area includes not only EDAC choices, which can be expected to vary as a function of the mission of the hardware subsystems in both complexity and capability, but also signal processing and combining to achieve a reduction in information data rate without loss of information transfer (compression and reconstitution) and to provide data block regeneration. It is proposed that investigation be performed to determine:

- A family of waveforms, including sphere of application, to modulate the expected signal types
- EDAC coding that maximizes throughput
- Data compression techniques for maximizing information transfers using minimum encoded data bits
- Data reconstruction techniques for developing missing data
- Approaches to message piecing that provide updates only to changed portions of a previous message (a form of template).

From these investigations would evolve algorithms and data transfer protocols that would enhance timely message delivery thus reducing the problem of working with old (perishable) data.

11.4 NUCLEAR EFFECTS AND HARDENING DESIGN

11.4.1 Problem Identification

There is a need to consider the effects of nuclear weapons on SDI BM/C^3 networks. In particular, terminal defense networks may be particularly vulnerable to such effects since ground-terminal radio circuits must pass through the atmosphere. Nuclear disturbance of this environment is of special concern.

It is well accepted that exo-atmospheric nuclear bursts can seriously degrade offensive communication circuits, but it is sometimes assumed that a successful SDI would stop such weapons and thus be free to operate in a non-nuclear environment. Instead, however, a potential threat of massive and numerous nuclear detonations resulting from an enemy strategy of salvage fusing is possible. Such a situation could have disastrous consequences to SDI if not planned for and mitigated.

In salvage fusing scenarios, enemy warheads would detect SDI counter-attacks and automatically self-destruct. In addition to salvage fusing, some weapons could be designed for high-altitude detonation for the primary purpose of disrupting communications. Leakage warheads are also of concern, particularly during early phases of deployment when the full defensive capability is not in place. Thus, in a central force exchange, both exo- and endo-atmospheric explosions from hundreds of warheads might be expected.

A severe disturbance of the upper atmosphere would result, along with direct nuclear effects on terminal-defense systems and equipment. The nuclear effects on propagation of radio signals would be severe, and detonations occurring near ground, airborne and space assets would have a significant effect on equipment and devices, including damage from EMP, radiation, and thermal/shock and direct blast.

The threat of such a "worst-case" environment must be investigated, understood and mitigated by special design approaches before SDI Battle Management C^3 systems can be considered viable.

11.4.2 Nuclear Effects on SDI BM/C^3

In considering the many problems caused by nuclear weapons, the principal concern is with the effects on radio propagation. Disruption of critical command and control circuits could occur at the very times when dependable operation is most needed. The disruptions could be minor or massive, depending on the attack scenario, and could endure for much longer than the "windows of criticality" when SDI coordination data must flow reliably.

One must postulate protracted nuclear effects on all bands as a worst case, and seek equipment and system designs with sufficient margin to operate reliably in spite of such a hostile environment. From a nuclear effects standpoint, the need for most SDI circuits to operate in the upper frequency spectrum (because of the need for wide bandwidths and high data rates) is an advantage in that the effects die out in seconds to minutes, unless repeated seeding attacks are postulated. But in any case, the effects of even a single nuclear burst are significant if they occur at a critical time and must be considered.

Also, the HF and even the VLF/LF bands should not be disregarded in this consideration since certain SDI architectures may depend on these lower frequencies for status report-back circuits from subscribers located at very long ranges or for interconnection to offensive networks. In the HF and low VHF bands (2-100 MHz), work is underway to develop a radio system than can search through these frequency bands for new modes of radio propagation brought on by high altitude detonations themselves and then use these "bomb modes" to establish communications networks automatically [5].

Planning for nuclear effects on SDI must also consider EMP and other direct-attack effects. The design disciplines required to mitigate against radiation, EMP, etc. threats are well understood, and specifications can be patterned after existing offensive applications in these areas. Additional work needs to be done to improve ways to protect against repeated upset caused by repeated detonations. Techniques have been developed for Minuteman that allow electronics to be reinitialized from hard memory within 10 msec following an EMP event. Extension of such techniques to SDI should be considered.

Blast effects from leakage warheads must also be considered, particularly for ground-based elements of the system. One might argue that SDI will certainly protect its BM/C^3 sites; therefore, hardened sites will not be required. The opposing argument is that SDI BM/C^3 sites will share targeting priority with national command centers and ICBM launchers, so the antennas must be hardened to a high degree at least during early phases of deployment. Assuming that hardened sites are required, there is work yet to be done in hardened antenna technology for such sites.

Low frequency antennas, that is, below about 2 MHz, must be buried to achieve any degree of hardness. A buried individual antenna element suffers a decrease in efficiency from its above-ground counterpart of 99 to 99.9 percent (-20 to -30 dB). This technology was developed in the early 1960's and is deployed in Minuteman Wing-6 in Missouri.

At higher frequencies, a hard/soft concept can be employed where a blast-soft antenna is backed up by multiple replacements stowed in a hardened configuration. This technique was deployed at other Minuteman sites. These antennas have normal efficiencies, but they are not serviceable during the trans-attack. When one is blown away, several minutes may be required to erect the next one and get it operational. Obviously, the required number of backup antennas is equal to the number of blasts encountered.

An antenna may be buried in a low loss material which will survive the blast effects in an effort to regain efficiency. Experiments are currently under way (1985) with promising results at Offutt AFB. This hardening technique offers even more promise at VHF/UHF as those antennas are inherently smaller and the burial problem is more tractable.

To counter the generally expected lower efficiency of hardened antennas, transmitter power must be increased. Alternatively, phased arrays of hardened antennas could be used to concentrate the radio signal toward a known, but not necessarily fixed target receiver location. Phased arrays provide directivity and can be electronically steered very quickly via computer control. The most formidable hindrance to phased array design, mutual coupling between elements, is largely mitigated by burying the elements. On the other hand, other aspects of phased array technology such as beamwidth/sidelobe control in widely spaced arrays, and phase/amplitude control through widely spaced amplifier-antenna groups is just beginning to receive attention in the VLF-HF bands.

Buried antennas have been successfully tested on meteor-burst links. Here again, high transmitter power must be employed to counter lowered antenna gain. High radiated power on meteor burst links translates to a decreased mean time between usable trails on a specific path, hence a higher traffic capacity.

Exhaustive tests of hardened antennas for UHF air-to-ground or for satellite communication have not been performed. Fitzgerald's work at NBS Boulder in the late 1970's was marginally successful, but he was working under constraints of antennas mounted on Missile-X vehicles in hardened tunnels which contained a large amount of reinforcing steel. His work should be revisited from the aspect of optimizing the hardening for the antenna rather than optimizing the antenna for the hardening.

11.4.3 Approach

Because of the significant degradation of SDI BM/C^3 networks that can result from nuclear effects, the mitigation of such effects must be included in network, system and equipment designs. The process of designing for operation in a nuclear environment begins by postulating nuclear scenarios that might be expected in SDI situations, and standardizing on a few that represent the bounds of possibilities. These can then be used as the basis for system and equipment design improvements.

Example SDI links should be analyzed or modeled as test cases early in the network design process, and anti-scintillation and anti-jam criteria developed from these examples for use by system designers. With threat-mitigation features assumed, networks can then be designed for specific architectures and can be subjected to nuclear effects analysis using system models.

In the same manner, criteria for expected EMP and direct-attack threats should be developed and applied to emerging designs. Existing databases from the nuclear effects community will be a good starting point for this process as well.

With respect to hardened antenna design, improved analysis techniques taking various nonuniformities of ground and insulating materials into effect are currently under development [4]. Such efforts are important and should be continued. There are system engineering tradeoffs to be made between transmitter power and antenna efficiency. Phased arrays can be helpful in point-to-point communication but offer almost no advantage in a broadcast, full azimuth coverage mission. Testing and research would be valuable in grading hardness vs. depth in various types of soil. Materials research might result in a hard, low-loss matrix material for encapsulating VHF and UHF antennas. However, it is not reasonable to expect that hardened antennas will achieve the efficiency of their soft counterparts.

11.5 DEFENSE/OFFENSE C^3 NETWORK COORDINATION

The need for defense/offense coordination is paramount in battle management. Cross notification and/or confirmation between defensive and offensive forces of detected missile launches, early identification of sea launched missiles as friend or foe for boost phase kill, and intercommunication of attack size and probable targeting are just a few examples of areas where defense/offense coordination is beneficial or necessary for SDI BM/C^3.

As SDI C^3 architectures are considered, comparisons are invariably made with existing networks of the WWMCCS and MEECN, and the already formidable concerns of SDI grow even more complex as similarities and desirable interfaces are recognized. In many cases, SDI planners can easily see how existing data circuits could be adapted to their purposes, but less obvious is the effect on strategic offense capabilities by so doing. The WWMCCS must remain viable under any conceivable scenario, so extensive use of WWMCCS circuits for SDI purposes may not be viable.

But some commonality must eventually exist between the two requirements, even if developed separately, if for no other reason than that communications to the same single point (the Commander-in-Chief) must be provided in the same real time. Also, there exists only so much spectral space, and the spectrum requirements of SDI are substantial. Operational, technical, and even nuclear effects coordination between the two extremely large applications must thus be a part of any BM/C^3 effort for SDI.

Requirements for defense/offense coordination must be established early as part of the overall requirements for SDI BM/C^3. Appropriate interface points to the WWMCCS and MEECN networks must be identified if needed. These requirements and interfaces must be considered in the subsequent design of the SDI BM/C^3 system.

11.6 RECOMMENDATIONS

Recommendations for research related to communication issues for SDI BM/C^3 are summarized below:

Network Security - Security requirements and policy for the SDI BM/C^3 system must be established based on defined threat scenarios and identified communities of interest within the system. Initial research should be directed at the identification and quantification of the security threat. It should be recognized that the peacetime threat may be different from that in wartime. In an actual engagement the perishability of data may allow fall back modes of reduced security operation for some types of communications traffic in response to battle damage or dynamic network configuration requirements. If feasible, the use of such operational modes might allow a gradual degradation of security for some types of communications traffic to facilitate overall system performance.

Communications connectivity requirements must be developed and evaluated to define communities of interest within the overall system. Based on these analyses, the defined security threat, levels of classified information and personnel clearances, the need for multilevel or other types of security policy can be assessed. Responsibility for enforcement of the security policy must be allocated between the communications system and other BM/C^3 elements. These security requirements must be considered interactively in the development of network and control architectures and protocols for SDI BM/C^3. Technology developed on other programs, such as Blacker, should be reviewed for applicability to SDI. Security protocols developed for programs such as Blacker, AUTODIN II, and the DARPA Survivable Radio Network program, should also be reviewed for applicability to the SDI BM/C^3 network.

Based on this work, a network security design can be developed and the need for further technology development identified.

One of the significant issues in the design of SDI network security is the development and implementation of a trusted host implemented by a security kernel that mediates attempts to access classified data. Attempts have been made to design and develop a trusted host, but none has met with complete success as defined by NSA. One such system was attempted for DCA's AUTODIN II packet communication system but proved only partially successful. The SDI testbed modeling and simulation capabilities and facilities will be required to analyze concepts for a trusted host if used in SDI and to model and simulate the host system. In addition, a hybrid hardware/firmware/software simulator to exercise the candidate host and its algorithms would provide a very effective mechanism of design and development. The simulator would be programmed to act as a user community in an attempt to discover kernel access status where specific denial of a particular process to a simulated user or other process has been overridden, and security, therefore, breached.

Coding and Waveforms - Initial research efforts should be directed at the definition of the communications environment in which the BM/C^3 system is to operate. This includes a definition of the ECM threat for each type of link anticipated, e.g., ground-to-ground, ground-to-air, ground-to-satellite, satellite-to-satellite, etc. Consideration should be given to anti-jam, low probability of intercept, low probability of detection, and transmission security requirements. In addition, noise levels, both natural and man-made, and propagation effects, both natural and nuclear-induced, should be projected for each type of link for the range of operational scenarios and time frame anticipated.

Using this data and projected data rate requirements, the effectiveness and relative costs of various ECCM waveforms can be evaluated for each type of link. An important consideration in this evaluation, particularly for time varying waveforms used to defeat repeat jammers and to provide transmission security for links to satellites, is waveform acquisition and synchronization over links with long and time-varying propagation times. Based on these evaluations a family of standard waveforms, including sphere of application, can be defined to modulate expected signal types within the BM/C^3 system.

Research should be performed to define appropriate EDAC coding to maximize communication throughput for the expected environment. The use of data reconstruction techniques including waveform multiplexing using orthogonal coding techniques such as Walsh Functions, should be studied as an alternative or supplement to more standard EDAC coding techniques. Effectiveness and cost of such techniques should be traded-off in terms of system performance against alternatives of message ACK/NAK schemes with retransmissions, and message flooding (the transmission of multiple, identical messages by alternate routes and/or at different times).

The use of data compression techniques and message piecing should be studied to reduce traffic load in the communications network. Such use must be evaluated in terms of the network data error rates achievable with the selected waveforms and coding techniques since such techniques may be more sensitive to communications errors. Techniques for integrating traffic types that previously have been studied under the umbrella of the Integrated Services Digital Network (ISDN) need to be extended from a coding and waveform standpoint to determine their applicability to the SDI network.

To assess the power of a candidate coding technique or the performance of a particular waveform adequately will require computer modeling and simulation capabilities. Mathematical investigations have been used to quantify the performance of codes and waveforms, but these investigations have been somewhat limited by the simplifying assumptions of the models needed to quantify the environment through which the signals being encoded must pass. Statistical measures of effectiveness have been derived using computational techniques, such as Monte Carlo simulations, and these techniques must be applied to new waveforms and coding approaches to understand their capabilities to accomplish bit and symbol error control, synchronization, and electronic threat protection including antijam (AJ), low probability of detection (LPD) and low probability of intercept (LPI). These models, which could be used to quantify the SDI BM/C^3 environment, must be upgraded and extended to include nuclear effects and more generally to lessen any dependence on assumptions previously needed to accommodate mathematical tractability. Data compression and orthogonal combining techniques must be investigated either mathematically or through simulation to study their effectiveness for enhancing communications throughput and message delay minimization.

Nuclear Effects and Hardening Design - A range of nuclear environments must be developed from postulated nuclear scenarios for SDI engagements. Initial research should be aimed at specifying nuclear scenarios that might be expected in SDI situations and standardizing on a few that represent the bounds of possibilities. Many such scenarios exist, and the classified RISOP and DNA cases used in analysis of WWMCCS effectiveness could be used as a starting point. Because of the salvage fusing phenomena however, scenarios that contain many high altitude bursts should be emphasized. The effects of enemy jamming on radio circuits must also be considered as part of the overall threat, and nuclear effects on the jamming signals themselves must be included.

Large classified databases from radio and atmospheric effects models have been generated by DNA and AFWL, and these should be accessed by experts in the nuclear effects community and applied to SDI. Although the scenarios will be different, much of the work necessary to define these effects and work out approaches to design mitigation has already been done.

As a next step, example SDI radio and SATCOM links should be analyzed as test cases early in the network design process, and anti-scintillation and anti-jam criteria developed from these examples for use by system designers. With threat-mitigation features assumed, networks can then be designed for specific architectures and can be subjected to nuclear effects analysis using system models. Techniques to be considered include alternate routing, redundant communications over different media or frequency bands, frequency agile systems, and advanced coding techniques. Research directed at improving the computer modeling and simulation of nuclear effects on communications links and equipment is needed to support this effort.

Research should be performed to develop criteria for expected EMP and direct-attack threats for the various elements of the communications system, i.e., ground-based, airborne, and satellite. Particular attention should be directed toward mitigating the threat of continued system upsets caused by repeated nuclear detonations.

Antenna hardening through burial has gained in prominence, but relatively little is understood about the propagation impacts expected from burial. In addition, phased arrays of hardened antennas offer promise, but algorithm and hardware design research is needed to determine power budgets required, potential gains, electronic steerability and element coupling. Antenna hardening for airborne and spaceborne platforms needs treatment to determine the impact on communication availability caused by coupling, radiation, shock, blast, and particle dosing.

Defense/Offense C^3 Network Coordination - Requirements for defense/offense BM/C^3 coordination and appropriate interface points must be established as part of the overall requirements for SDI BM/C^3.

11.7 ACKNOWLEDGEMENT

Substantial contributions were made to this section by Dr. R. H. Bittel and Mr. R. P. Buckner of Rockwell International. Their help is gratefully acknowledged.

11.8 REFERENCES

1. James C. Fletcher, et al., Report of the Study on Eliminating the Threat Posed by Nuclear Ballistic Missiles (U), Vol. V, February 1984.

2. H. F. Harmuth, Transmission of Information by Orthogonal Functions, Springer, New York, 1969.

3. J. L. Walsh, "A Closed Set of Normal Orthogonal Functions," American Journal of Mathematics, Vol. 45, pp. 5-24, 1923.

4. Gerald J. Burke, Numerical Electromagnetics Code - Method of Moments, User's Guide Supplement for NEC-3 for Modeling Buried Wires, Lawrence Livermore Laboratory, October 1983.

5. D. O. Weddle and R. J. Waldschmidt, "Advanced HF/Extended Range Communications System," Rockwell International IR&D Technical Plan, Vol. XI, December 1985.

6. Stephen T. Walker, "Network Security Overview," IEEE Security and Privacy Conference Proceedings, April 1985.

7. Department of Defense Trusted Network Evaluation Criteria, Draft, 29 July 1985.

SECTION 12 - COMPUTER SYSTEMS

DR. WILLIAM J. KENNY
CONTROL DATA CORPORATION

12.1 ISSUES

This section discusses the major issues associated with the development of computer systems to support SDI BM/C^3 processing functions. In this discussion, computer systems are interpreted to include the combination of hardware and software needed to meet performance requirements.

The computer systems play an important role in relating the sensors, weapons, platforms, and human resources into a complete layered defense system. The large scale of the complete system and its necessarily distributed characteristics implies a need for reliable, high-performance distributed computer systems.

It is not clear that the necessary computer systems will evolve from the rapidly developing commercial computer developments, and it is necessary to investigate the SDI computer needs and focus research and development efforts on the SDI unique issues. Such efforts cannot only improve the entire BMD performance but have the potential to permit more effective use of sensors and weapons, which in turn can improve system effectiveness and reduce costs.

The SDI BM/C^3 computer requirements are due to the need to distribute the processing functions over a number of remote platforms, especially in space. The need to simultaneously meet aggressive requirements for throughput, capacity, dependability and survivability in a hostile environment calls for a highly integrated design approach.

The development of adequate computer systems is dependent on satisfactorily addressing the following key issues, listed in order of priority:

1. Development of computer systems architectures to meet BM/C^3 performance requirements. This includes allocation and partitioning of requirements to specific platforms.

2. Reliability and fault tolerance to meet system life and critical battle period reliability.

3. Interconnecting networks at several levels, both between segments and between processor units on a given platform.

4. Design flexibility to accommodate changes in threat, technology, and treaties.

5. Coordination with other SDI processing requirements to ensure compatibility and commonality.

6 Radiation hardening, which is necessary to a survivable system.

7. Security of the computer system from unauthorized access or denial of use due to intent or accidental interference.

There are other issues, but these are either of lesser criticality or are being met by other research or development programs and are therefore not singled out for SDI attention.

The following sections take up the seven general issues listed above and discuss possible research and development approaches for each.

12.2 ALTERNATE APPROACHES

12.2.1 Development of Computer Systems Architectures to Meet BM/C^3 Performance Requirements

A number of recent and current studies at the system and subsystem level are aimed at developing BM/C^3 performance requirements. While the indications are that the computer throughput and size capacity requirements will be challenging and the software development complex, there are no requirements for specific computer systems on each system platform. The issue of concern here is that the computer architectures must be developed for use in specific platform environments and should be suited to processing requirements. It is therefore very important to partition the BM/C^3 problem to individual platform segments and allocate system requirements for processing on a platform basis. The platforms will be distributed on the ground, air, and space and each will place different performance, environmental, and reliability requirements on the computers involved.

The spaceborne computers will present the most critical design problem due to the combination of size, weight, power, reliability, and nuclear requirements. A design approach that skews the processing requirements allocation to ground segments can be expected to alleviate the overall system processing problem. The cost penalties due to space processing must be introduced early into the system-level trade-offs along with the consideration of communications, autonomy, survivability, and other critical factors. An adequate solution to the spaceborne computer needs is the key to risk reduction for the total BM/C^3 computer solution and research efforts should be concentrated there.

Subissues associated with development of spaceborne BM/C^3 computer systems are:

- Throughput needs are estimated at from 50 MIPS to over 1 GIPS for a single platform. These estimates may not represent a desirable allocation between space and ground. In any case, it appears that a solution calls for multiple computers on multiple platforms, interconnected by high performance networks. Emphasis should also be placed on decomposition of the BM/C^3 problem for parallel processing.

- Memory size for spaceborne processing is expected to exceed 10^7 bytes per platform. There will be a large random access requirement. There will also be a need for a portion of the memory to be nonvolatile to provide for nuclear event and fault recovery. At the present time, there is no satisfactory space-qualified bulk store.

- Further BM/C^3 algorithm definition and characterizations are needed to guide computer architecture design. Data errors, signal/noise levels, threat levels, etc., influence algorithm approach and behavior. Computer sizing and performance needs will be strongly nonlinear functions of problem size. There will be an interaction between algorithm selection and candidate computer architectures (parallelism, storage hierarchy, network structure).

  Development of algorithms with specific consideration for real-time processing and compatibility with computer architecture can be expected to reduce the processing requirement significantly (by a factor of two or more) from that needed through the use of general-purpose library functions.

  The BM/C^3 algorithms and associated data-management processing are expected to be characterized as general data processing (as opposed to "signal processing") with need for both numeric (fixed and floating point) and symbolic processing.

  An important need is to provide the simulation, analysis, and evaluation tools and facilities needed to adequately specify and design the BM/C^3 processing functions and their integration into the overall BMD system. These tools and facilities must be provided as a carefully selected set to support a hierarchical design process, since full scale detailed simulation of even subsystems and interfacing environments would require enormous resources.

  Early algorithm development can also suggest possible architecture features such as specialized coprocessors or instructions to improve performance/hardware ratio.

  There is a potential conflict in the need to drive computer architecture design with specific application-related requirements and the nature of the SDI BM/C^3 requirements which will continue to evolve for several years. The development and uses of a representative strawman problem and requirements set can serve to decouple the computer and system research tasks and permit simultaneous development. Indeed, early progress in computer architecture development can be fed upward into system design activities to improve allocations to subsystem requirements.

- Software development has been raised as an issue because of the problem scale, complexity and verification needs. The recommended approach consists of:
  - Emphasis on software design specifications directly traceable to the system and subsystem requirements as allocated to each host platform. This serves to divide and conquer the overall problem. In addition, achieving a good specification is half the battle, since many residual software errors are traceable to ambiguous or erroneous specifications.
- Coordination of software design with sizing analysis and algorithm development effort to improve problem understanding at a detailed level.
- Support research for new automated software development and test tools based on CAD engineering methods. These methods have yielded revolutionary improvements in computer and VLSI design effort and have resulted in error reduction and performance increases not otherwise attainable. There is a strong motivation to similarly automate software development and testing.

Supporting Research Areas. Research areas that relate to the problem of developing computer architectures to meet BM/C^3 requirements are:

- Study of architectures suitable for high-performance processing in space. Develop candidate architecture concepts employing parallelism at various levels of graininess to increase throughput and fault tolerance. Identify and evaluate specialized architectures and processor adjuncts for high-precision fixed and floating-point arithmetic, data base access, and symbolic processing. Architecture concepts include homogeneous or heterogeneous networks of specialized or general purpose processors, processors incorporating multiple specialized functional elements, data flow or control flow architectures, and vector processors. The area of parallel computing research is currently very active [Hayn82, Kais84, Neve85, Gehm85, Anton85, Pao85] and the SDI program can profitably draw upon these activities. In addition, there are a number of current and planned SDI oriented computer architecture programs sponsored by BMD, RADC, and DARPA that can be leveraged.

- Development of fast nonvolatile read/write memory for space.
- Development of hierarchical set of simulation and analysis tools.
- Development of automated software design and testing tools.
- Research on BM/C^3 algorithms for distributed parallel processing.

12.2.2 Computer System Reliability/Fault Tolerance/Availability

The critical need for overall dependability, that is the combination of reliability, fault tolerance and availability, of the BMD system translates directly to stringent requirements for highly-reliable, fault-tolerant computers. While all computer elements must exhibit this dependability, the driving need is the severe requirements for the spaceborne processors due to:

- Long-mission life on unmanned space platforms requires programmable reconfiguration and recovery capability.
- High availability for critical battle periods under environmental and weapons effect stress.
- The spatial distribution of processors which impacts the capability to coordinate data bases.
- Need for autonomy, at least for some critical periods.

Fault tolerance and the associated fault-detection capabilities cause a large impact on system throughput, size, and cost and tends to drive processing from space platforms to ground or other manned platforms where repairs by replacement is possible. Because of this effect, reliability considerations should be introduced into the initial requirements. It is necessary to grade reliability requirements by function, since not all functions are equally critical, and an indiscriminate worst case specification will force an expensive over design.

It is believed that the most cost effective approach to computer dependability is implementation using high-reliability components and processes and the reduction in components count through the use of VLSI (or Wafer Scale Integration). There are, however, difficulties due to the desire to use advanced technology and the uncertainty in reliability prediction for future technology. As a result, there is a need for improved methods for reliability prediction based on more realistic modeling of failure mechanisms.

Commercial experience with automated manufacturing also indicates that development of reliable component manufacturing, packaging, assembly, and test processes can be a means of attaining dramatic improvements in computer reliability and this general approach should also be applied to the processes for building and testing the BM/C^3 spaceborne processors.

The natural and hostile radiation environments in space cause a significant transient error rate that must be countered by design to reduce susceptibility and test and redundancy features to detect and recover from such errors.

The need to constrain size, weight and power for spaceborne computer systems motivates the development of efficient fault-tolerance techniques; that is, techniques whose ratio of dependability improvement/hardware increment is high. For example, coding techniques are efficient compared to triple modular redundancy with voting techniques. While efficient coding techniques are applicable to storage elements, there is a need to find comparable techniques for arithmetic and logical units and control functions. Concurrent error detection schemes [UILL84, Schu85, Wong83] and totally self-checking techniques [Halb84, Cook73, Gait85] are representative techniques for consideration for application at logic function levels. A hierarchy of techniques at all levels needs to be evaluated. Various system-level approaches involving redundant processors and software implemented fault tolerance are potentially useful [Neve85, Hsia85, Koo85].

In view of the above discussion, the requirements and environments associated with reliable BM/C^3 spaceborne processors are unique to the military and one should not expect commercial solutions to be adequate. Research should be supported to:

- Improve reliability prediction based on theoretical fault-mechanism models
- Develop automatic techniques applicable to manufacture of inherently reliable computers
- Develop efficient fault tolerance and error-detection techniques, incorporate these into candidate BM/C^3 architectures and evaluate for effectiveness.

12.2.3 Interconnecting Networks

The nature of the BM/C^3 problem requires a distributed computer system consisting of spatially separated ground, air and space platforms, and with multiple computers on each platform.

The required interconnecting networks exhibit two distinct characteristics, requirements and design problems:

- Interplatform Networks
  This includes space to ground, space to space, air to space, air to ground, and various relay links. The variable geometry between platforms, line-of-sight considerations, propagation characteristics, security, and reliability requirements dominate the design requirements. Operation of these networks requires distributed computer resources to maintain links, antenna pointing, message

routing, protocol processing, and encryption/decryption. The data processing to support network communications is expected to be a significant portion of the BM/C^3 processing load. This processing must be performed on each platform in the network and requires local data bases describing network status and configuration.

- Intraplatform Networks

  These provide the high-speed interconnects between the multiple computers needed on each platform to accommodate the BM/C^3 processing load as well as the communications to other on-platform subsystem processors, sensors, and actuators. The two most critical requirements are the high processor-to-processor bandwidth (of the same order as processor-to-memory bandwidth) and a fault-tolerance capability to recover from a large number of failures without manually replacing modules or connections. The major problems to be addressed are:

  - Design of a high performance network with massive fault tolerance.
  - Development of algorithms for rapid and automatic (autonomous) reconfiguration of remaining operational resources into a useful processing complex.

Research areas needed to support.issues unique to SDI include:

- Definition and design of interplatform communication processing functions. Specifically, investigations are needed to determine the computer features needed to support communications and methods for using the network to recover from network or computer failures.
- Development of high performance automatic computer interconnect networks capable of reconfiguration after multiple failures. This problem is analogous to the automatic placement and routing problem that has been successfully attacked for printed circuit and VLSI design. The concepts would be generalized and extended to network reconfiguration and recovery.

12.2.4 Design Flexibility

The long development cycle for computer architectures, computer hardware and software can result in rapid obsolescence if the computer system is not designed for flexible application and to be readily extendible. Flexibility of design is of particular importance for the spaceborne computers due to the long mission times and the relative inaccessibility for maintenance and upgrade.

Performance and reliability goals drive the computer architecture toward special purpose design, which can normally be expected to be more efficient in the use of hardware. This is in conflict with the need for the computer system to provide the functional flexibility to accommodate political, threat and technology changes that occur during the system design and throughout its mission life.

The recommended approaches to meet the flexibility issue include:

- Develop and employ automated tools and methodology to reduce the design cycle for both hardware and software. Techniques for coordinated, concurrent hardware and software development are particularly recommended, especially those related to functional specification, evaluation, and verification. Techniques such as SREM (Software Requirements Engineering Methodology) [Alfo85] and concurrent software and hardware design [Macd84] are representative of approaches to be further automated. If the very successful design automation tools developed for hardware could be adapted and applied to software, the potential for increased software development productivity would be enormous.

- Use general-purpose computer architectures to the greatest extent consistent with performance goals. This will support algorithm flexibility and also improve hardware commonality and modularity. The interconnecting networks should also be designed for flexibility of topology, operating modes, and extensibility.

There are two related research areas to be pursued which, while of importance to the SDI effort, are of broad interest for all military systems:

- Development of tools and methodology to reduce computer system design cycle, especially for software.
- Development of broadly applicable flexible high-performance computer architectures.

12.2.5 Coordination with other SDI Processing Requirements

The spaceborne BM/C^3 processors will share platforms with sensors and supporting signal and data processors. There are benefits to be attained by coordinating the design of SATKA and BM/C^3 function processing, since collocated processing can take advantage of hardware commonality and improved interfaces and integration. Coordination approaches include:

- Subsystem interfaces should be designed for compatibility and efficiency.
- Common processor designs are desirable. These can improve fault tolerance by increased pooling of spare resources. This is particularly important when justifying the incorporation of innovative architecural features where a weighted evaluation based on the pool of all processing functions serviced on the platform should be performed. Common architectures will also reduce software costs and improve software reliability.
- Common interprocessor network designs improve reconfigurabiity and flexibility for load redistribution. The same network architecture might serve in both signal and data processing.
- Potential for interfaces between function areas (different subsystems) on a given platform to be spread within a processing complex.

The research needed is to perform a coordinated analysis of all computing functions on a platform basis in order to define potential commonality areas.

12.2.6 Radiation Hardening

Radiation hardening will be an important requirement for all BMD computing systems, but will be particularly critical for spaceborne processors, which will be subject to long term cosmic radiation as well as high levels due to weapons effects. Size, weight, and power constraints in space limit the robustness of the electronics and the practical amount of shielding.

Design considerations associated with radiation hardening include:

- VLSI trends toward reduced feature size and decreased power consumption impact hardenability. Single event upset (SEU) phenomenon may limit reduction in VLSI feature size.
- Shielding weight penalties for space-based systems bias shifting of processing function allocations to ground-based segments wherever possible.
- Radiation effects will impact communications between spatially distributed system elements. This will tend to result in impaired performance and more reliance on autonomous processing.
- Since there is no commercial interest in the problem, military support is needed for solution.
- While the radiation problem is common for all SDI functions, two aspects of the BM/C^3 functions are unique:
  - Certain decisions and commands are critical and upsets due to radiation must avoid erroneous outputs. Special software techniques and memory protection features are needed.
  - During an attack phase, time lines are short and stringent requirements are imposed on circumvention and recovery processes. The need is for special software recovery techniques and nonvolatile memories.

Fortunately, the general characteristics of the radiation problem have been a subject of research for many years and the primary additional need is to apply the data and hardened technologies to the BMD computer system design and to engineer specific solutions for BM/C^3. It is assumed that the VHSIC program will focus on the hardening needs for submicron devices and that the ongoing hard-technology developments (GaAs, CMOS/SOS, etc.) will continue to receive government support. For the SDI program, it will be necessary to apply the output of these efforts to constrain architectures and their implementations. It is also important that the needs for SDI, especially if hardening requirments exceed current goals, be communicated to these ongoing programs.

12.3 COMPUTER SECURITY

BMD computer security needs special consideration due to the system's large, complex distributed characteristics, the critical nature of the mission, and the long mission life during which the system may be subjected to compromise or penetration.

Computer security is interpreted to encompass the prevention of unauthorized access or use and system sabotage during development or use. The problem is complicated because of the multilevel classification and access control of the information involved. Also, all users will not necessarily have clearance to all classification levels, requiring interfaces to include authorization and identity verification.

Providing a full multilevel secure computer system in each platform can adversely impact performance due to additional checking and data segregation required. Related to this is a companion increase in development and deployment costs. It may be possible to operate all spaceborne processors in a "system high" mode and rely on encryption for communication to other platforms. It may be practical to restrict multilevel secure modes to ground-based segments in this way.

A distributed system poses special design problems because of the many interfaces exposed to penetration attempts.

There is an interrelationship between security and fault-tolerance in that a failure prone system is insecure. There is a potential to combine fault-tolerance and security techniques to improve overall efficiency. Recent work on combined error detection/correction/encryption codes are representative of the approach [Rao84]. Other techniques include memory protection, file access controls, failure monitoring, and proven executive kernels.

Since the software system reliability strongly influences security, it is desirable to develop a secure approach to software development, verification, and maintenance to prevent incorporation of flaws or weaknesses, either intentionally or inadvertently.

Research in techniques for distributed computer security should be supported. This may include use of simulation techniques to evaluate and verify security design.

12.4 RECOMMENDATIONS

The preceding sections discussed the principal issues involved in the development of computer systems for SDI BM/C^3 function processing. For each area, design approaches were suggested along with an identification of needed research studies. It is recognized that several of the research areas may already be subject to current activities for other applications and, while the need for BM/C^3 may be significant, support may be justified on the basis of other DoD needs. In other cases, the BM/C^3 problem clearly justifies specific research support, but the benefits may be broadly distributed to several DoD programs.

The summary of recommended research shown in Table 12-1 indicates both the recommended support and the beneficiaries for each research activity. The notation used is:

Recommended support:

- BM/C^3 or DoD

- S = specific support, A = application of other research, G = general support
- Benefits - X indicates beneficiary, either BM/C^3 and/or DoD in general.

Since the issues were discussed in approximate priority order, the listing order in Table 12-1 reflects this ranking also.

It should be noted that the needs are directed primarily at specification, design, and evaluation activities. We have not called for inventions or breakthroughs. The overall goal is to take a realistic measure of the specific computer requirements to reduce the risks associated with unbalanced requirements allocations or overspecification. Two areas that come closest to calling for breakthroughs are the reliable processes and the reduction in design cycle for software and this research would be of general benefit to all military programs involving computers.

Table 12-1. Recommended Research Summary

		RECOMMENDED SUPPORT		BENEFITS	
ISSUE	RESEARCH ACTIVITY	BM/C^3	DOD	BM/C^3	DOD
REQTS/ SIZING/ ARCH.	HIGH PERF. SPACE ARCH.	S		X	X
	NONVOLATILE MEMORY	A	S	X	X
	COMPUTING REQTS.	S		X	
	SIMULATION FACILITY	S		X	
REL/F.T.	FAULT TOLERANT TECH.	A	G	X	X
	RELIABLE PROCESSES	A	G	X	X
NETWORKS	COMMUNICATIONS S/W	S	G	X	X
	MULTIPLE-F. T. NTWKS.	S	G	X	X
FLEXIBILITY	REDUCTION OF DESIGN CYCLE, HDW/SW	A	G	X	X
	FLEXIBLE HIGH PERF. PROC. ELEMENTS	S		X	X
PLATFORM COORDINATION	COORDINATED ANALYSIS OF PLATFORM FUNCTIONS	S		X	X
RAD. HARD.	VHSIC RAD. HARD.	A	G	X	X
	HARD TECHNOLOGY DEV.	A	G	X	X
SECURITY	DISTRIBUTED COMPUTER SECURITY TECHNIQUES	S	G	X	X

SDC 1876

This listing gives an overview of the broad areas for an integrated BM/C^3 computer system research program. Further effort is required to define specific tasks and approaches for each area. It is recommended that initial efforts focus on the BM/C^3 spaceborne computer architecture study, which would coordinate with and incorporate results from supporting investigations related to fault tolerance, networks, and data management.

12.4.1 Spaceborne Computer Architecture for BM/C^3 Processing

This research task is motivated by recognition of the critical nature of the BM/C^3 processing functions, especially in space, and the concept that an efficient architecture should be designed to reflect the algorithm characteristics.

The initial research is recommended to develop architectural concepts and evaluate cost/performance characteristics based on strawman processing tasks representative of spaceborne BM/C^3 processing functions.

This research should jointly incorporate realistic performance and capacity goals, reliability/fault tolerance/hardness requirements, and implementation technology constraints for the 1990-2000 time frame in order to permit effective design trade-offs.

Candidate architectures will need to support distributed processing and incorporate multiple processors in local and spatially distributed networks. Processing tasks include data management for large, complex databases, correlation, assignment functions, coordinate conversion, filtering, logical operations, and inference processing. Processing characteristics will include high precision fixed and floating-point arithmetic and symbolic processing. Problem complexity requires general-purpose processing capability, but performance requirements can be expected to motivate incorporation of high degrees of parallelism, special-purpose coprocessors or adjuncts for arithmetic, symbolic, or database operations. There is a need to research the design of fault-tolerant networks interconnecting the processing nodes. These should be treated as an integral component of the computer system architecture. The interaction of innovative architectural concepts, such as data flow with fault-tolerant techniques should be studied to identify complementary architectural/fault-tolerant features.

The study should include selection and evaluation of both hardware and software error detection and fault tolerance techniques. Techniques and tools for application software development for highly parallel, fault-tolerant processing, should also be investigated.

The study approach should include a quantitative high-level evaluation of the effectiveness and cost impacts of the comparative architecture concepts based on computer-aided analysis, simulation, and emulation.

The study will be leveraged by ongoing parallel and high-performance architectural research supported by BMD, RADC, NASA and DARPA, as well as broad commercial and academic activities.

12.4.2 Test Bed Requirements

The quantitative performance/effectiveness evaluations essential to the spaceborne-computer architecture for the BM/C^3 study require the use of a high-performance multiple computer test bed capable of supporting analytical evaluation of performance/reliability computations, simulation and emulation of parallel processing computer systems incorporating innovative architecural features. Each node should be interconnectable with a flexible topology network with programmable protocols. The number and capacity of test bed nodes should be selected to adequately represent the spaceborne-processing distributed architecture features and evaluate performance using reasonable simulation/emulation times. In addition, the test bed needs a "host" computer for management and control, mass storage subsystem, and interactive operator/experimenter interface subsystems (terminals or micros). Networks of commercially available parallel processor arrays are potential test bed candidates, especially if the processors support emulation.

12.5 REFERENCES

[Alfo85] M. Alford, "SREM at the Age of Eight; The Distributed Computing Design System," IEEE Computer, Vol. 18, No. 4, April 1985, pp. 36-46.

[Anton85] R.T. Antony, "A Distributed Machine Intelligence Architecture for Battlefield Signal and Information Processing," Harry Diamond Labs Project 567454, August 1985.

[Cook73] R.W. Cooke, and W.H. Sisson, T.F. Storey, W.N. Toy, "Design of a Self-Checking Microprogram Control," IEEE Trans. on Computers, Vol. C-22, No. 3, March 1973, pp. 255-262.

[Gait85] N. Gaitanis, "A Totally Self-Checking Error Indicator," IEEE Trans. on Computers, Vol. C-34, No. 8, Aug. 1985, p. 758.

[Gehm85] D.L. Gehmlich and K.L. Schroder, ""Multiprocessor and Tracking System for SDI Applications," GOMAC, 1985, pp. 199-202.

[Halb84] M.P. Halbert and S.M. Rose, "Design Approach for a VLSI Self-Checking Mil-Std-1750A Microprocessor," IEEE 1984, FTCS-14, pp. 254-259.

[Hayn82] L. Haynes, et al., "A Survey of Highly Parallel Computing," IEEE Computer, Jan 1982.

[Hsia85] B.M.Y. Hsiao, "Fault Tolerant Computing in the VLSI Era," GOMAC, 1985, pp. 185-190.

[Kais84] S.H. Kaisler, "Parallel Computing Workshop Report," IEEE Computer Society, Computer Architcture Technical Committee Report, Feb 1984.

[Koo85] R. Koo and S. Toueg, "Checkpointing and Roll-Back Recovery for Distributed Systems," Technical Report, Cornell University (Contract No. MDA903-85-C-0124).

[Macd84] M.H. MacDougall, "Instruction-Level Program and Processor Modelling," IEEE Computer, Vol. 17, No. 7, July 1984, pp. 14-24.

[Neve85] C. Neveu, "A Fault Tolerant Distributed System for Weapons Platforms," GOMAC, 1985, pp. 195-198.

[Pao85] Y.H. Pao and J. Kim, "Artificial Intelligence Processing Techniques and Supporting Architectures," Final Report, June 1985, (Contract No. F30602-82-K-0045) RADC.

[Rao84] T.R.N. Rao, "Joint Encryption and Error Correction Schemes," University of SW Louisiana, 11th Annual International Symposium on Computer Architecture," June 1984, pp. 240-241.

[Schu85] M. A. Schuette and J.P. Shen, "Processor Self-Monitoring Using Signatured Instruction Streams," Carnegie-Mellon University for SRC (Private Communication 1985).

[UILL84] University of Illinois, Coordinated Science Laboratory, "Reliable High Performance VHSIC Systems," Final Report, Sept. 1984 (Contract No. N00039-80-C-0556).

[Wong83] C.Y. Wong and W.K. Fuchs, J.A. Abraham, and E.S. Davidson, "The Design of a Microprogram Control Unit with Concurrent Error Detection," IEEE (1983), FTCS-13, pp. 476-483.

SECTION 13 - FAULT TOLERANCE

Malcolm W. Johnston
C. S. Draper Laboratory

13.1 SCOPE

This technology area provides those design attributes that promote robustness for dependability of the general-purpose digital computation and interconnection resources and functions, both hardware and software elements. The scope includes digital systems required by sensor, weapon, SATKA, and other platform and ground-based functions, but is primarily focused on the BM/C3 function. These digital resources will be the key integrating medium for the SDI system, therefore it is essential that a high level of confidence can be placed in their integrity. The processing and control elements must not only dependably implement functions such as BM/C^3, but they will contribute directly to the dependability of the sensor and weapon systems they monitor and service.

Dependability is here meant to include simplex computer reliability (MTBF) and tolerance to design errors, random faults/failures, physical damage, "malicious" intervention (e.g., embedded traps in the operating system), and natural and nuclear-induced sources of interruption or destruction such as radiation. Operational and procedural elements were not addressed. Unless it is important to distinguish betweem them, errors, faults, and failures will all be referred to as faults (as in fault tolerance).

Several dependability issues are listed here but discussed under other technology categories because they are so intimate to those areas.

- Security and dependability are mutually supportive concepts in that it is not possible to have a secure system that is not dependable while, at the same time, the error checking of security techniques provides another means to improve reliability and fault tolerance. These security, anti-jamming, and encryption issues are discussed in Section 9, Data Management, and Section 11, Communications.

- Inter-platform network protocols, error control, failure recovery, etc., represent a major source of SDI system vulnerability. These global interconnection issues are treated in Section 10, Networks.

- Reliability (in its narrowest sense) is an attribute that is provided through design, manufacturing, and testing techniques that reduce the failure rates of non-redundant computers; i.e., fault avoidance techniques. These are discussed in Section 12, Computer Systems. Fault and damage tolerance is the focus of this section.

Strawman SDI architectures and BM/C^3 system requirements will eventually have to be postulated to support meaningful research (see Section 13.3, Task 0). In the meantime, several initial observations on the unique characteristics of the SDI application can be made.

On the negative side, the SDI BM/C^3 architecture must be designed to provide precise and time critical digital information management and control functions in a demanding environment. The peak processing, communications, and response time requirements will coincide with enemy precursor defense suppression efforts which probably will result in massive physical damage to the U.S. defense assets and a disturbed communications environment.

On the positive side, inherent in the strategic defense application is the need for a robust system architecture which utilizes multiple layers (e.g., boost, post boost, midcourse, terminal), with each layer composed of many replicated elements or nodes (e.g., sensor, weapon, BM/C^3 platforms), and where each node, in turn, may employ several dissimilar systems or phenomenologies (e.g., redundant information available from multiple sensors). The beneficial result of the need for this "system-of-systems" is a structure which provides, at this higher system level, replication, partitioning, dispersion, and dissimilarity, four key ingredients to most fault tolerance approaches. These application characteristics should be used to advantage.

The discussion is organized as follows: Section 13.2 describes the major fault tolerance issues, including the justification factors leading to each issue and SDI-unique requirements. Candidate approaches to resolving each of the issues are then outlined, including current state-of-the-art and anticipated advances in technology. Section 13.3 suggests a step-by-step recommended research plan which includes a prioritization of the technology efforts. Section 13.4 concludes with a characterization of the fault tolerance technology status. An appendix contains several background papers.

13.2 ISSUES AND CANDIDATE APPROACHES

13.2.1 Coverage

13.2.1.1 Issues

Present aerospace computer system failure rates, or their more familiar inverse, mean-time-between-failures (MTBF), fall far short of providing the levels of mission success required by life critical functions (by four or five orders of magnitude). Therefore, this reliability or fault avoidance shortfall must be offset by the selective use of multiple replications of system elements (or coded redundancy where applicable). In addition, the system design must embody the mechanisms necessary to provide the coverage (probability of detecting, identifying, and recovering from random faults) necessary to ensure that these spare resources can be utilized. That is, fault tolerance must be provided.

A synergistic relationship exists between component-level MTBF and coverage. An MTBF level exists below which, even with perfect coverage, it becomes impractical to add the number of spares necessary to satisfy target success probabilities (interconnection complexity is too great, maintenance rate necessary to replace failed spare elements may not be practical, etc.). Fortunately, if coverages approaching 100% (e.g., 99.99%) are feasible, great leverage can usually be applied towards meeting target success probabilities with only moderate increases above this minimum MTBF level and/or in the maintenance cycle. On the other hand, overall system dependability falls off sharply as coverage degrades because of incorrect responses to failures.

While significant improvement in MTBF would simplify critical system design, applications such as the SDI, which are vulnerable to damage, must still provide spare elements along with the coverage, interconnections, and reconfigurability necessary to utilize them.

Though casual approaches to fault tolerance, mostly for commercial implementation, have received much recent attention, progress in fault tolerant computer technology for applications that demand very high levels of dependability has been slow. This is due to the introduction of very low probability, difficult-to-handle failure modes as significant factors. These stressing fundamental requirements are associated with clocking, data consistency and congruency, fault containment, resource contention, vexatious failure modes (e.g., correlated, transient, Byzantine General, etc.), and the desire for transparency of fault tolerance techniques to applications software. These issues are only now becoming well understood.

13.2.1.2 Candidate Approaches

Several techniques are presently used to detect, identify, and recover from faults in the processor/memory part of digital systems. These can be classified in four major categories as follows:

Temporal Checks - Watchdog timers are used extensively in computers to detect processor failures. Typically, if a certain address is not written-to every few milliseconds, an interrupt is generated which can be used to reset or fail-safe the processor. Watchdog timers can also trap runaway software faults. In addition, most central processing units (CPUs) use bus timeouts to trap faults in which a memory does not respond. Also, instruction re-try is used by mainframe computers as a way of recovering from these faults.

Integral Checks - Integral checks are built into the basic number representation. Codes are an example of this type of fault-detection and/or correction method. The simplest and most common representation of coded information is the parity bit. It can detect single-bit failures. Parity is widely used in the memory of most minicomputers and in the data path of most mainframes. Hamming code is an example of a more complex form of code. It cannot only detect, but also correct errors. Most mainframe computers use Hamming code or some other variation of it to detect and correct memory faults. Arithmetic codes (i.e., codes that are preserved through arithmetic operations on variables) are receiving renewed interest.

Diagnostic Checks - Diagnostic checks are undertaken to uncover latent faults. Read-Only Memory check-sums, illegal opcode tests, tests of voters, and test problems are examples of diagnostic checks used to uncover faults in parts of the computational core which are not exercised by regular programs.

Replication, Comparison, and Voting - The previous three classes of fault detection and correction methods have relatively low coverage. To achieve the very high coverage, as well as the transient protection and fast response times required for life-critical systems, the most common method is to replicate the hardware.

Hardware can be replicated at a low level or at a high level. An example of the former is the Raytheon/Air Force Spaceborne Fault Tolerant Computer (SFTC). This computer utilized redundant elements of the CPU rather than replication of the whole CPU to achieve high fault detection coverage. Most other fault tolerant computers use high-level replication which means replication at the CPU/memory level or even at the computer level. What distinguishes these architectures from each other is the way replicated channels are operated.

Systems such as the MD-80 Digital Flight Guidance System use two identical computers, but they operate such that their results are only approximately equal even under no-fault conditions, which results in the need to utilize "redline" tolerance levels to flag failures. Other examples of approximate replication are the AFTI/F-16 Flight Control System and the NASA Ames Redundant Asynchronous Multi-Processor System (RAMP). Another way to operate redundant channels is to place them in the standby mode. The AT&T ESS No. 1A is a dualredundant system but at any given time only one computer is in charge. Transition to the backup computer is made when the primary computer fails. The Voyager spacecraft's Command Computer Subsystem (CCS) is also an example of this approach, without the off-line cross checking of the ESS.

Systems such as the SRI SIFT, the Shuttle DPS, and the CSDL FTMP and FTP are examples of exact or congruent replication. That is, the results of the redundant processors or computers match exactly or bit-for-bit when there are no faults in the system. The exact replication approach provides, conceptually, the highest coverage and masking of errors and transients for continued, uninterrupted service with data integrity. This is especially important in command/control applications where errors of omission can be as costly as errors of commission.

The methods used to detect and recover from faults in the interconnection media can be classified in the three categories discussed below:

Temporal Redundancy - Two most notable examples of this category are the CRC checks used in Local-Area Networks and the Bose-Chaudhri (BC) codes used in communication networks. The CRC check consists of a form of check-sum transmitted at the end of each message. The BC codes are used to correct for burst errors or multiple errors which are quite prevalent in communication networks.

Spectral Redundancy - Spectral redundancy implies sending the same information over the same communication medium but using different frequencies. Wavelength-Division Multiplexing, if used to transmit the same data on different frequencies, would be an example. Its known uses are limited due to lack of maturity of Wavelength-Division Multiple Access (WDMA) technology and also due to the highly correlated failure of all the redundant information if the communication medium fails.

Spatial Redundancy - Spatial redundancy simply means redundant communication links. Typically most avionic systems use redundant buses. Another form of spatial redundancy is the mesh network such as the ARPANET, the AT&T telephone network, and the CSDL mesh network.

13.2.2 Connectivity

13.2.2.1 Issues

Connectivity is implicitly part of the coverage problem. Recovery from a failure requires system partitioning which is "sympathetic" to reconfiguration, interconnections that support dispersion for damage tolerance and replication for fault tolerance, and flexible interconnection of spare elements, e.g., to support pooled sparing. Intra-node connectivity must include inter-computer communications and input/output transmissions to sensors and effectors, though not necessarily via the same interconnection system.

Interconnection design must gracefully accommodate functionality changes over the long, evolutionary SDI system lifetime and heterogeneous elements, technologies, protocols, instruction sets, etc. An "open" system is required. However, costs and interoperability requirements also suggest the need for carefully selected areas of standardization (global interfaces, data representation, etc.). This dichotomy of requirements has not been solved for today's strategic or tactical BM/C^3 applications. The SDI application will be more demanding, particularly if the NATO/allies subset is considered, and represents an enormous technical and political challenge.

13.2.2.2 Candidate Approaches

A great deal of experience has been accumulated at both the global inter-node level (e.g., ARPANET) and at the intra-node or platform level. The former, inter-node connection, is discussed in Section 10, Networks.

Topologies for the latter, intra-node connection, include the bus, ring, mesh, tree, or full cross-strapping. The multiplex bus is most vulnerable to faults and damage. Full cross-strapping is the most secure, although it is also the most complex. For fault and damage tolerant applications, it appears that mesh or fully cross-strapped topologies are most attractive.

Recent and ongoing work is addressing these problems at the intra-node level. A NASA-CSDL mesh concept utilizes circuit switched nodes (the ARPANET is also a mesh, in that case using message switching nodes), and the NASA-SRI/SIFT utilizes a fully cross-strapped topology with dedicated point-to-point links.

While these and other efforts are developing promising approaches to the need for robustness and rapid reconstitution, techniques for graceful accommodation of heterogeneous users still require considerable attention. The commercial world offers a lead to be followed here.

13.2.3 Control

13.2.3.1 Issues

This section addresses only those aspects of control that pertain to providing a fault and damage tolerant digital system which can dependably implement BM/C^3 functions. In this sense, control is also implicitly part of the fault detection, identification, and recovery (or coverage) problem. Sections 5 and 7 address the higher-level control issues associated with battle management itself.

The control of distributed processing resources that support SDI battle management functions will differ from past applications (mostly commercial) in several significant ways:

- The parallel processing tasks will be time critical and interactive.
- The system will depend on robust access to input data from multiple sensors and it will be required to output commands reliably to multiple effectors (e.g., weapons).
- Timing or synchronization requirements, for instance for data consistency or congruency, will have to contend with possible communication delays. (It will help to have BM/C^3 algorithms which are robust to corrupted or lost data.)

- Resources must be monitored and will have to be reallocated to accommodate a dynamic interconnection topology, I/O and processing overloads, and damaged or failed elements. Functions must be migrated accordingly.
- Fault tolerance of power distribution and control will have to be brought to parity with data distribution and control, perhaps by utilizing a unified approach.
- The system must provide autonomy with respect to unique local functions and cooperation amongst global or system-wide functions.

Operating systems to manage and control digital resources in this environment are only now beginning to be developed and significant research remains to be done.

13.2.3.2 Candidate Approaches

A recent survey of distributed operating system developments indicated that no existing system is designed to operate with stringent real-time constraints. A NASA sponsored effort at CSDL (AIPS) includes the development of an initial, limited implementation and a BMD/SDC effort has evidently been initiated since the above mentioned survey.

For early efforts to develop such a system, it may be possible to adapt an existing transaction-oriented operating system, or alternatively, if Ada is to be used for the implementation language, to make appropriate modifications to an Ada uniprocessor run-time package.

Perhaps the two most important issues in the design of distributed operating systems have to do with:

- Coordinating access to shared objects, (e.g., sensors and weapons) with "fairness" and data consistency.
- Maintaining the consistency of data in the face of user errors, application errors, or partial system failure.

"Locking," "timestamps," "permits," and "tickets" have been proposed in the literature to solve the first problem, and each provides a method for ensuring that "serializability" is preserved. Various implementations of a software-structuring technique called "atomic actions" have been proposed (for non-interruptability) to solve the second problem.

13.2.4 Software Fault Tolerance

13.2.4.1 Issues

While much has been done to improve hardware dependability through fault tolerance, designers of critical software still rely on fault avoidance. Fault avoidance has been achieved for fairly complex systems and applications such as the NASA Apollo and Shuttle software. Validation and verification techniques ranging from module testers to large scale, high fidelity simulations have been used to catch design, specification, and coding errors. Although the payoff in terms of highly reliable software has been great, so has the cost.

This situation in the field of software is analogous to that prevailing in the area of hardware in the early 60's. Life critical digital computers, such as the Apollo guidance computer, were developed at great expense to be very reliable without being fault tolerant. However, it was evident by the late 60's that even this level of hardware reliability (e.g., 10^{-5} failures per hour) needed to be improved by four or five orders of magnitude for some critical applications (e.g., the FAA requires a 10^{-10} failures per hour level for computer systems that provide life-critical fly-by-wire control.) Fault tolerance was the only avenue available to achieve these high levels of dependability.

Similarly, we may not be able to continue to rely on developing very nearly perfect software for future life critical missions. Given the increasing complexity of aerospace applications such as the SDI BM/C^3, these systems will contain latent software faults (design errors). Software fault tolerance may be necessary in order to prevent system failure (and possibly to compound the problem facing "mole" programmers who might attempt to embed traps in software elements).

Though a number of techniques for fault tolerance are in the early stages of development, none has been extensively applied yet on a large scale (hundreds of thousands of lines of code). This technology has not yet reached parity with its hardware counterpart and, as a result, represents a weak link in any fault tolerant system.

13.2.4.2 Candidate Approaches

As pointed out earlier, SDI application characteristics can be used to advantage. Any fault tolerant software approach selected for the BM/C^3 system will benefit from the existence of a relatively compartmentalized architecture. Multiple weapon and sensor systems can be employed for each of several defense layers, with minimum coupling between systems within a layer and between layers. Therefore, from the outset, useful partitioning and dissimilarity provide a degree of protection against single point failures (of hardware or software). If this protection is not considered adequate, a trade-off will be necessary between spending available resources on fault avoidance (i.e., pursuing very reliable software and reliability prediction techniques), or spending it on fault tolerance techniques such as those outlined below.

In addition to the well known and effective use of back-up software (the Shuttle fifth computer is an example), several other techniques are presently under study. Multi-version software is any fault tolerant software in which two or more versions are executed and the results are compared. The types of multi-version software that are most often discussed are N-version programming and Foodtaster. N-version programming consists of N versions of a program (N > 1) which have been independently designed to satisfy a common specification. The results are compared and, based on a majority vote, can identify faults. Foodtaster was originated by Morris and Shephard of the Cranfield Institute of Technology in England. It requires two versions of software and multiple processors. Both software versions execute serially on each processor which doubles the throughput requirement or halves the actual throughput. Fault detection is performed by an estimator that compares all outputs, using extrapolation for this comparison. An added advantage of these approaches may be a reduction in the verification and validation effort required on each version (or increased reliability of each version) due to the comparison of versions during development; i.e., "one hand washes the other."

Another well known technique utilizes recovery blocks. It consists of a program which is provided with an acceptance test, and at least one standby spare. If the primary program passes the acceptance test, the spare program is never executed. This scheme is dependent on the acceptance test being invoked and being correct. The recovery scheme requires backup error recovery if a fault is detected, where the state of the system is saved just prior to the execution of the primary module so that, if the primary module fails, the system state can be restored to allow execution of the alternate module.

Hybrid techniques have been proposed to avoid the disadvantages of multiversion software and recovery blocks, but most still suffer from the need for an acceptance test.

13.2.5 Damage Tolerance

13.2.5.1 Issues

A large class of faults must be dealt with by a dependable BM/C^3 system. Software design errors (discussed above) and low probability hardware faults are particularly troublesome, but some success is being seen in these areas. Similarly, the military has for several years addressed the problem of local damage with degrees of success, e.g.; for systems on tactical fighter aircraft or other weapon platforms.

The SDI, however, poses a new combination of problems that includes all of the old ones plus a nuclear environment and offensive scenario that can cause massive damage on a global scale. It is likely that an enemy attack would be preceded by a precursor defense suppression effort that could include direct interception of space-based SDI elements and nuclear detonations to disturb the communications environment and damage or destroy susceptible SDI system components. In addition, the possible use of salvage fusing on the enemy's part, where their RV's are designed to self-detonate when intercepted, will increase the possibility of an intense radiation environment for the SDI system. This will include EMP, radiation, thermal shock, and blast effects.

This damage will occur nearly simultaneously among elements such as computer systems which, unfortunately, will probably be at their peak workload. Traditional fault detection, identification, and recovery techniques usually have been built on the premise that one random fault will be "cleaned-up" before another occurs, where clean-up times may be fractions of a second. Nearly simultaneous events (e.g., milleseconds apart) will cause havoc unless special techniques are devised.

In addition, past approaches to damage, which assumed the ability to reconfigure around the locally affected system, may now be faced with massive global damage, which will make system reconstitution in a timely and graceful manner much more difficult.

Providing the damage tolerance necessary to maintain an effective SDI system, particularly the BM/C^3 functions, is the most challenging issue discussed in this section. Very little attention has been paid to these problems by the fault tolerant computer community and, accordingly, basic research and applications-specific development is needed.

13.2.5.2 Candidate Approaches

Survivability at the highest, "systems of systems" level can be provided to a degree by combinations of platform shielding, stealth, active protection (e.g., evasion, weapon carrying escorts), and proliferation of elements such as space-based platforms.

At the system level, it may be possible to extend the approaches developed for more local damage protection; e.g., physical shielding, replication, compartmentalization, dispersion of elements, etc. The notion of functional survivability is of particular importance here; e.g., via excess capacity from similar and dissimilar elements and the reconfigurability to utilize it, rather than individual system survivability techniques.

Degrees of automation can expedite damage assessment, reconfiguration, and system reconstitution, where it will be necessary to blend the human's ability to react to unanticipated situations and the computer's ability to diagnose and respond to the situation quickly, accurately, and comprehensively without the possibility of procedural errors.

A number of comprehensive programs (SDI and non-SDI) are presently underway to develop radiation hard electronic components for space-based military assets. The approaches include development of less radiation vulnerable technologies such as GaAs, circumvention (e.g., hibernation and restart techniques), shielding, and even reversion to older technologies at the cost of power, speed, and density. Sections 11 and 12 discuss radiation hardness issues in more detail.

Single event upsets that are caused by background cosmic radiation can be handled as a beneficial by-product of fault tolerant hardware techniques that employ replicated versions and voting to mask errors. Higher levels of radiation that are caused by nuclear detonations would affect all replications and defeat such a system.

13.2.6 Evaluation and Verification

13.2.6.1 Issues

The substantially improved dependability which fault tolerance provides is inevitably purchased at the cost of system complexity. The system designer's problem is exacerbated by the fact that there is still little practical experience in the design of fault tolerant systems, and what experience there is indicates that intuition alone is often insufficient. The design process should encompass the total system; i.e., the hardware, operating system, application software, and operational procedures. The latter, for instance, has been a major source of computer downtime, yet it continues to receive little attention.

Quantitative evaluation tools must be developed to augment the design engineer's experience during the iterative BM/C^3 development process by measuring the degree to which alternative design choices satisfy dependability, performance, and cost criteria. Complex models will be necessary to determine the sensitivity of the system measures of effectiveness to each of these design alternatives and, accordingly, to allow intelligent trade-offs and design decisions to be made. These measures of effectiveness and design evaluation criteria must also be identified.

Similarly, verification tools and techniques will be relied upon to a degree never before required. As has been observed by SDI critics, complete end-to-end verification of an SDI system in a realistic, fully stressed operational environment will not be possible. While this will not be the first instance where such was the case (strategic offensive weapons, lunar landing, etc.), the number of system elements that must be coordinated and the possibility of an extremely hostile nuclear environment compound the problem for the SDI.

The critics also encourage the image of a monolithic software program of 10 million or more lines of code which must operate flawlessly on first use, despite inadequate testing and "unknowable" requirements, or the entire defensive system will be rendered useless.

Although these characterizations are exaggerated, they are the dimensions to the challenge that will require special attention. The system must be available when required during a lifetime in the 20 year range, and dependable during the short (hours) time involved in a major exchange.

In general, our ability to evaluate, verify, and modify today's systems has not kept pace with present design complexity and implementation technology. An environment must be developed and its fidelity verified in order to evaluate the effectiveness of fault detection, identification, and recovery techniques, and to build credibility with the public, the operators, and the enemy. Performance evaluation technology must also be refined. The focus of computer system performance evaluation has been on transaction-type systems rather than real-time applications, and the bulk of the information available is addressed to systems of this type.

Modeling, simulation, and emulation will be heavily depended upon for both dependability and performance evaluation, particularly for the dimensions that cannot be tested operationally (conflict scale workload involving a saturation attack in a nuclear disturbed environment). High fidelity, observability, and computational tractability will be required, as will degrees of flexibility to respond gracefully to changes (ours and theirs).

13.2.6.2 Candidate Approaches

Although evaluation and verification of the complex SDI "systems of systems" is an imposing challenge, steps can be taken to make the problems more tractable.

Again, the potential to compartmentalize SDI architectures can be used to advantage. Battle Management and other functions can be implemented with a number of relatively uncoupled systems and software programs of individually modest size. This relieves cost and complexity (scaling will be more nearly arithmetic with size than exponential), allows a modular approach to evaluation and testing, limits the detrimental effects of system or code modifications, and adds to system robustness.

Evolving analytical adjuncts to the verification and validation process can be used to augment the more familiar life and stress testing. The use of Markov dependability models is an example. The mathematical states of a model correspond to various operational states of the system; for instance, normal operation and various degraded modes of operation (and performance). The transition probabilities determine how the operational state of the system evolves in time as failures occur, or as decisions with respect to the presence or absence of failures cause the system configuration to be altered. Given the inherent link between failures and degraded performance in a fault tolerant system, a Markov model can also be used to evaluate performance on a probabilistic basis as a function of time. Thus, an evaluation of the probability of a minimum throughput level over an entire mission, and the likelihood of success of performing certain functions during the mission are both possible. This subject, known as "performability," has been investigated on a rather limited basis in the technical community.

The potential availability of relatively abundant digital processing hardware resources should allow for the inclusion of built-in aids to the debugging and testing process. For example, the ability to selectively single-step, read/record, monitor/trace, or update the state of the machine(s) is particularly important where multiple copies must be synchronized for fault tolerant implementations. On-line error detection and logging is useful to store data following abnormal behavior. The simulations that previously provided these functions are no longer practical because of the speed and complexity of today's architectures; therefore, these capabilities must be built into special test devices or the target machines themselves to provide run time visibility.

Additional aids to the software/system developer are becoming available, including interface and consistency checkers, interrupt and timing analyzers, standards enforcers, structure and flow analyzers, and higher-order languages. Independent verification and validation, rapid software prototyping, and greater reuse of already validated software modules are also becoming more common practices. Eventually, computer-aided design techniques may automate the software development process from requirements specification to verified code.

Analytical proofs-of-correctness represent a very long-term approach to software verification and validation. Substantial research is necessary if this technique is to become useful on a practical scale (i.e., for large, complex programs).

It is expected that modular, distributed information processing systems will allow more graceful growth, change, and technology upgrading of individual elements over the long life of the system without requiring wholesale regression testing and revalidation of all the elements; that is, by encouraging compartmental and/or cumulative approaches to reclaim the very high degree of trustworthiness required.

The historically successfuly incremental process of final system validation, where increasing proportions of the actual system are added to increasingly realistic testing scenarios and environments, is particularly compatible with the planned evolutionary development and deployment of an SDI system. The surveillance mode will be continuously in operation and other functions such as tracking boosters and reconfiguring fictitiously damaged communications networks can be routinely and non-provocatively exercised. Simulations of unanticipated scenarios are regularly used by the military and NASA, where antagonistic "Red teams" confront the system with contingencies. This operational readiness testing must include continued "flexing" of online and spare system elements to simulate stressed workloads and to expose latent failures and allow timely maintenance.

The cumulative effect of these exhaustive evaluation and testing sequences has provided confidence in life critical systems such as Apollo and the Shuttle in the past, and promise to meet even the challenges of the SDI.

13.3 RECOMMENDED RESEARCH PLAN

Task 0 - BM/C^3 Problem Characterication (Near Term, High Priority)

The techniques utilized to provide a dependable BM/C^3 system will be affected by higher-level SDI architectural design choices such as the degree of human control, the degree of autonomy versus functional integration and sharing of resources, and the degree of dissimilarity between multiple sensor or weapon systems and phenomenologies. Sensitivity analyses should be performed to identify the dependencies that exist between fault tolerance constraints and these SDI architectural choices in order to list attributes of an SDI architecture that would be helpful to a BM/C^3 system, and to identify architectural characteristics that would be particularly troublesome. In addition, sensitivities to potential changes in system design, defense strategy, and enemy threat should be identified.

Several strawman SDI system architectural constructs should be baselined, representative of the spectrum of design directions that can be anticipated today, to serve as an initial point of departure for these sensitivity analyses (e.g., near-term, ground-based, KEW only, limited defense versus long-term, ground and space-based, DEW and KEW, "splendid" defense).

In addition, representative BM/C^3 algorithmic characterizations (processing type, required data integrity, degree of parallelism possible, etc.) and performance levels (CPU throughput, I/O bandwidth, memory size, response times, etc.) must be postulated.

These strawmen architectures and BM/C^3 problem characterizations need only be initial-iteration approximations and should draw heavily on work already accomplished by industrial contractors during the architecture "horserace," the SDIO Pilot Architecture Study, and the ongoing BMD/SDC efforts.

The deliverable, in addition to the sensitivity analyses mentioned above, would include quantified requirements for digital resources, without which much of the meaningful fault tolerant computer system architectural research cannot proceed. This will provide an in-house (BMD) basis for conducting research and calibrating contractor performance.

Task 1 - Coverage, Connectivity, and Control (Long Term, High Priority)

The more sophisticated approaches to processor/memory coverage, which usually employ tight synchronization and voting for error masking, have come to grips with most of the stressing requirements (source congruency, Byzantine failures, etc.). However, they are complex and may be inefficient in their use of hardware resources.

The objective of this first of two sub-tasks would be to translate, through fundamental innovation, today's hard earned, in-depth understanding of the problems associated with providing comprehensive coverage into significantly simpler realizations which retain compliance with the most stressing requirements.

The principal trade-offs will most likely be between the present brute force replication that takes advantage of the regularization of VLSI circuit technology, and the less-hardware-intensive use of coding techniques which to date have required more complex custom logic. Though a hybrid of the above is likely, entirely new approaches would be welcome.

If the replication approach is pursued, it is expected that more effective techniques for trading performance for fault tolerance in real time will be required. If the coding approach is selected, it would be conceptually advantageous to employ unified, end-to-end techniques that encompass processor, memory, and communications elements.

Simplification of connectivity and control techniques should be a beneficial by-product of the above efforts. For example, much of the control complexity required in present implementations is a reflection of the fundamental complexity of the approaches taken for fault tolerance. As an aside, any fault tolerent system, a separate and parallel effort should be directed at developing a distributed operating system that meets the stringent control requirements outlined in paragraph 13.2.3.1.

As a second subtask, a base of experience must be built on the application of fault tolerance techniques to parallel processing architectures.

Several demanding BM/C^3 application functions may lend themselves to parallel processing (e.g., multisensor data correlation, optimized resource allocation/target assignment, etc.). Adding fault tolerance to existing parallel processing architectures as an applique is not likely to be successful. Therefore, architectures that combine significant parallelism and fault tolerance

need to be developed and evaluated against stressing sample BM/C^3 functions, both for special (e.g., array) and general-purpose processing.
This investigation should consider the possible utility of: the inherent redundancy in parallelism; of specialized co-processors; and of developing an integrated signal and general purpose processing architecture.

The most promising approaches in both sub-task areas should be selected for prototype evaluation and proof-of-concept demonstration in an SDI BM/C^3 test bed environment. For instance, a comprehensive body of empirical data should be gathered on the actual coverage levels provided by the various techniques suggested. The results of these efforts must eventually be unified into a coherent system-wide digital architecture.

Task 2 - Software Fault Tolerance (Long Term, Medium Priority)
The first of two sub-tasks involves: formulation of metrics for software reliability measurement; classification of the effects and severity of software failures, including high workload impacts; and assessment of the resource requirements for various software fault tolerance techniques, including their impacts on computer system architectures and vice versa. A possible adjunct effort to this would extend software reliability (errors remaining) prediction/evaluation techniques to an operational environment context rather than the present software development life cycle context.

The second effort would be directed at investigating fundamentally new approaches to software fault tolerence and the application of several best-candidate techniques (e.g., n-version programming and new innovations) to selected BM/C^3 critical core software elements. Proof-of-concept demonstrations would be performed in concert with appropriate prototype fault tolerant computer system development (see Task #4), perhaps in the context of test bed experiments and "fly-offs."

The investigations should consider the impacts of scaling these relatively small sized (code-wise) software experiments to very large application programs. The benefits of utilizing attached co-processors should also be assessed.

Task 3 - Damage Tolerance (Long Term, High Priority)

A survey should be conducted of recent programs that have had to deal with global damage tolerance (e.g., WWMCCS, Naval Carrier Task Force, GPS, etc.), and research initiated to extend this limited experience base to a selected strawman SDI BM/C^3 context.

The investigation should include traditional approaches such as physical shielding, dispersion, excess capacity and the reconfigurability to utilize it, and functional survivability through connectivity to dissimilar but functionally equivalent elements. Sensitivity analysis should be conducted to identify the architectural impacts of various techniques at all SDI system levels.

Special attention should be paid to massive damage events which require techniques for multiple, simultaneous failure recovery. Also, the practicality of utilizing computer aids, such as expert systems, to expedite comprehensive damage assessment, diagnosis, and reconfiguration should be assessed.

Task 4 - Evaluation and Verification (Near Term, High Priority)

The major objective of the first of two sub-tasks would be to provide models, simulations, and emulations that are tractable, yet retain the important characteristics and dimensions needed to realistically represent the complex systems and operational scenarios being examined.

Evaluating the effectiveness of alternative fault detection, isolation, and recovery techniques requires exhaustive fault modeling and means of observing system response in detail. Models must include the effects of failures and damage, the reconfiguration strategies, the frequency of false alarms, and their effects on the overall system functions. Random and stressed-state fault injection tools must be developed, as well as test case generation aids. These fault simulators should be extended to the context of the BM/C^3 problem so that actual system configurations, algorithms, and application code can be exercised.

Tools and techniques (e.g., Markov analyses) must be developed, and their fidelity validated, that integrate dependability analysis with performance analysis to obtain measures of system effectiveness; i.e., performance in the face of errors and failures. The models and analyses must represent all system elements, including the interconnection devices and media. Exercises with these tools should include representative workloads, overloads, and fault/damage scenarios.

The second sub-task involves selection of several fault tolerant computer system prototypes for architectural evaluation and experimental validation and to gain early experience with the models, simulations, and testing techniques required. This should also include stress testing of various configurations with algorithmic overloads and injection of random states and faults as was done earlier with models (above).

Beneficial by-products of this activity would include system-level experience with BM/C^3 application algorithms, distributed control, and local networks. In addition, experience with a test bed that incorporates a number of architectures and techniques will highlight the difficulties involved with providing connectivity for heterogeneous system elements; which is a desirable design attribute.

All of these tasks either involve development of tools and techniques that would contribute to a BM/C^3 technology evaluation and verification facility or require such a facility for their implementation/demonstration. Therefore, the development of a test bed should receive very high priority. It can be used to evaluate a broad range of technology alternatives, ensuring that sources of innovation are not closed off prematurely.

13.4 CONCLUDING REMARKS

Despite the technology challenges outlined here, there is reason to be optimistic. The SDI implementation and deployment timescale of 15-25 years is in our favor. Successful, complex computer systems such as the AT&T ESS were completed nearly 20 years ago. In addition, the relatively mature and rapidly advancing digital technology state-of-the-art must be compared to other enabling SDI technologies to maintain perspective. Directed energy weapon brightness for lethality, midcourse discrimination phenomenology, and physical survivability of space-based assets are each enormous challenges; even potential showstoppers. They give the BM/C^3 world breathing room to work their problems.

We have, therefore, taken an aggressive view of the present and anticipated digital technology track that the United States is following. The "going-in" position is that the higher-level SDI or BM/C^3 architectural decisions need not and should not be driven by fault tolerant digital processing hardware or software concerns, although damage tolerance will significantly impact these decisions. Full advantage should be taken of the degrees of forgiveness provided by the inherent robustness of the SDI "system of systems".

Given the enormous advantage the U.S. enjoys over the Soviet Union in computer technology, it should be looked upon as a potential force multiplier and source of relieving other SDI technology/phenomenology problem areas.

13.5 ACKNOWLEDGEMENTS

The author wishes to acknowledge the contributions of Dr. Doyle Thomas of the sponsoring agency, Mr. William McDonald and Mr. Charles Cooper of SDC, and the other panelists. He appreciates the help provided by Dr. Herbert Hecht, Dr. Daniel Siewiorek, and by his colleagues at the C.S. Draper Laboratory; Dr. John Deyst, Dr. Jay Lala, Mr. Philip Felleman, Dr. Bart DeWolf, and Dr. Richard Schabowsky, Jr.

APPENDIX - RELIABLE SOFTWARE FOR STRATEGIC DEFENSE - ISSUES AND DIRECTIONS

Dr. Herbert Hecht

SOHAR, Inc.

While topics of software reliability and fault tolerance have received increasing attention in recent years, there are still a number of unresolved fundamental issues that impede a systematic approach to the development and validation of highly reliable software. Decision making in the areas of system structure and resource allocation could be greatly improved by availability of creditable data on

- Frequency and effects of software failures
- Methodology and resource requirements for software fault tolerance.

As some examples of benefits that can be obtained by R&D activities in this area, the following are discussed below:

1. Standards for software reliability measurement
2. Classification of the effects and severity of software failures
3. Measurement of workload effects on software reliability
4. Effects of system architecture on software fault tolerance
5. Evaluation of resource requirements for various fault tolerance techniques.

The objective, current status, and recommended tasks for each of these topics are discussed below.

1. STANDARDS FOR SOFTWARE RELIABILITY MEASUREMENT

1.1 OBJECTIVE

The primary objective of this effort is to permit utilization of reliability experience across projects. This requires that software reliability attributes be quantified and expressed in a uniform manner. A secondary objective is to permit project managers to state software reliability requirements in terms consistent with system reliability requirements and that permit monitoring during the development phase and at acceptance.

A meaningful measure must support the calculation of a mission reliability or availability. For hardware the failure rate is an accepted measure of this type. A failure rate (based on computer execution time) can also be generated for software. Other possible measures are the mean number of executions of a program or the mean number of I/O operations performed prior to failure.

1.2 CURRENT STATUS

The IEEE Computer Society has had a standards project (P982) on Software Reliability Measurement since 1982, but has been unable to reach consensus on a standard. The primary difficulty is that many participants in the project equate reliability measurement with reliability prediction. Thus, the requirement for a simple quantitative metric gets intertwined with research into software quality factors (use of HOLs, requirements consistency, etc.). It is not likely that this effort will result in a useful product for several more years.

The reliability metric being developed under STARS is also likely to be software quality oriented. The June 85 draft of the data collection forms document shows the reliability metric to be derived from accuracy, anomaly management, and simplicity checklists. These measures are of some interest during in-progress reviews, but they bear no relation to the system reliability requirements or to reasonable acceptance criteria for reliability.

1.3 RECOMMENDED TASKS

1. Survey of existing data on software reliability measurement
2. Evaluation of existing measures against the requirements of BMD
3. Application of one or more candidate measures in pilot projects
4. Formulation of a recommended software reliability metric to SDIO.

Expected project duration: 15 months

Expected resources: 1.5 professional labor years

2. CLASSIFICATION OF FAILURE EFFECTS AND SEVERITY

2.1 OBJECTIVE

The objective of this effort is to permit focusing the effort for failure prevention and fault tolerance on areas which pose the greatest threat to reliable functioning of BMD software. Many software failures produce only negligible effects on the primary service performed by the computer. In one of the very few studies of the severity of software failures, the NASA shuttle office found that only 2 percent were critical and that only one-third produced more than minor effects. By isolating critical failures, resources for software reliability improvement can be directed more effectively.

2.2 CURRENT STATUS

The most widely used classification of software faults consider the fault mechanism (logic, computational, data format, etc.). This is useful for the selection of development practices and tools but has no bearing on the effect of the failure (e.g., stopping a computer, causing improper I/O operations, or failure to insert a form-feed). No specific work in this field is currently being carried out by DoD. In the NASA report cited above, the severity classification is a byproduct.

2.3 RECOMMENDED TASKS

1. Survey of existing data on effects and severity of software failures
2. Evaluation of measures for use in ballistic missile defense
3. Application of one or more candidate classifications in a pilot project
4. Formulation of recommendations in this area for SDIO.

Expected project duration: 18 months

Expected resources: 2 professional labor years

3. MEASUREMENT OF WORKLOAD EFFECTS ON SOFTWARE RELIABILITY

3.1 OBJECTIVE

The objective of this effort is to determine the adverse effect on software reliability caused by high computer workloads. During an actual engagement, the battle management computers are expected to be highly loaded, and that is the time when failures can be least tolerated. The quantification of workload effects is necessary in order to provide rational guidelines for performance margins in the battle management computers.

3.2 CURRENT STATUS

Work by Iyer and Rosetti at Stanford University and by Siewiorek and Castillo at CMU has pointed to significant increases in the software failure rate during periods when computers were operating at a high workload. In the Stanford studies, increases in the failure rate up to two orders of magnitude were reported

3.3 RECOMMENDED TASKS

1. Establish guidelines for a pilot study in a computer environment representative of battle management
2. Identify a site for the pilot study and provide suitable instrumentation
3. Conduct the study and report results
4. Formulate recommendations for performance margins for battle management computers based on the workload effects.

Expected project duration: 24 months

Expected resourcs: 3 professional labor years

4. EFFECTS OF SYSTEM ARCHITECTURE ON SOFTWARE FAULT TOLERANCE

4.1 OBJECTIVE

The objective of this study is to provide a systematic basis for considering software fault tolerance requirements during the selection of the top and intermediate levels of the computer system architecture. Although the "Software First" approach to computer system selection is gradually becoming accepted, there is little appreciation of the impact of computer architecture on software fault tolerance. Some software fault tolerance techniques require simultaneous operation of several loosely coupled (asynchronous) computers; others do not work well with geographically dispersed nodes. Most software fault tolerance techniques impose a considerable performance penalty, yet they must be implemented if reliability goals are to be met. Thus a study of architecture effects on software reliability is necessary to provide a rational guideline for an integrated hardware/software design.

4.2 CURRENT STATUS

While research into individual methods of software fault tolerance is being sponsored by RADC/COE, there is no effort underway to investigate such major questions as:

- Whether there shall be a central "Fault Tolerance Engine" to which the individual programs are hitched or whether each program shall be responsible for its own fault tolerance
- The optimum scope (extent of code protected) for each fault tolerance technique
- The benefits of "Lifeboat Computers" for combined hardware/software fault tolerance.

4.3 RECOMMENDED TASKS

1. Survey of currently applied software fault tolerance measures and the computers in which they reside
2. Evaluation of the suitability of computer architectures for various fault tolerance techniques
3. Recommendations for computer architectures and associated software fault tolerance techniques for battle management

Expected project duration: 18 months

Expected resources: 2.5 professional labor years

5. RESOURCE REQUIREMENTS FOR SOFTWARE FAULT TOLERANCE

5.1 OBJECTIVE

The objective of this effort is to provide a basis for estimating and evaluating of the resource requirements of various software fault tolerance techniques. At present, fault tolerance techniques are selected primarily on the basis of their projected ability to recover from failures. However, most fault tolerance techniques add considerably to the development cost and schedule, and many involve a very sizeable performance penalty. Memory requirements are also impacted when software fault tolerance is used. Both qualitative and quantitative resource requirements need to be established in order to permit cost effective selection of techniques.

5.2 CURRENT STATUS

As far as is known, there are no past or current efforts in this area.

5.3 RECOMMENDED TASKS

1. Establish qualitative resource requirements for each of the following techniques
   - Robust programming
   - Fault containment
   - Recovery blocks
   - N-version programming.
2. Identify applicable current uses of each of the techniques and collect data on resource usage
3. Evaluate the data obtained in the light of battle management software, and develop a resource estimation algorithm.
4. Establish a data base for collection of resource expenditure data for verification and improvement of the algorithm

Expected project duration: 18 months

Expected resources: 2.5 professional labor years

APPENDIX A-

SUMMARY TABULATION OF RESULTS FROM THE SDI LARGE SCALE SYSTEMS TECHNOLOGY STUDY

TECHNOLOGY	ISSUES	ALTERNATIVE APPROACHES	RECOMMENDATIONS
System Theory	1. Determine how to interconnect the major system components to achieve maximum effectiveness of the overall SDI system and find the best interconnection of elemental systems, within each component to reliably maintain the function of the component within the SDI battle environment.	1. Component interconnection network must be configured and dynamically reconfigured so as to optimize the overall performance criteria.	1. Efforts be made to more precisely identify and model functional requirements of system components to optimize networking.
	2. Develop a family of high-fidelity models to represent important component-level and element-level features and aggregate and/or decompose system features to obtain simplified models to allow effective analytical design procedures.	2. Decide on purposes of model and system features relevant to model purposes. Capture relevant system features in math-model format. Develop simplified models as surrogates of raw model. Use spatial, functional, and temporal aggregation. Validate the model via simulation.	2. Efforts toward system-theory/input-output modeling of sensor, computing, data management, BM information, BM decision, and weapon components. Effort to find ways of simplifying these models. Identify global and decentralized criteria. Develop SDI model/control simulation facility.
	3. a. **Identification of** battle-**management controll**ables. b. **Determination of** performance criteria to judge merits of control manipulations. c. Controllable cost-effectiveness measures. d. Design of control algorithms and information facilities for real-time control decisions. e. Tests to verify control performance and cost effectiveness.	3. a. & b. Approach from close familiarity with hardware and functions performed. c. Use some form of simulation based on simplified system models. d. Use a family of system models involving control design concepts from large-scale system theory. e. Verify via detailed simulation and tests.	3. Identify candidate battle-management controllables and associated overhead cost of controlling. Develop ways of modeling controllables in format of control theory and decentralized control design procedures that lead to fault-tolerant/adaptive performance.
Artificial Intelligence	1. **Fundamental** reasoning and planning **methodologies** for dynamic processes **based on** uncertain information.	1. In order for AI to have a significant impact on SDI BM/C^3 problems, research must be performed and proof-of-principle applications must be developed that take a new view of "real-world" problems. In particular, rather than simplifying the problems to remove the noise, uncertainty, imprecision, and blemishes inherent in real problems, these aspects must be treated directly in solving a set of scalable problems that are microcosoms of the SDI Battle Management problems.	1. Fund basic research in reasoning and planning in real world situations that include dynamics, uncertainty, noise, and distributed resources.
	2. Distributed, Cooperative Problem Solving		2. Focus and motivate the research with a set of scalable problems that stress real world issues, in order to evolve effective, distributed reasoning and planning capabilities.

TECHNOLOGY	ISSUES	ALTERNATIVE APPROACHES	RECOMMENDATIONS
Artificial Intelligence (Continued)	3. Development and use of Specialized Computational Systems.		3. Accelerate and motivate the development and use of specialized computational architectures.
Control Theory	1. Decoupling 2. Reliability 3. Security	Ad Hoc Engineering ● Hybrid State Spaces ● Hybrid Time Domains ● Hybrid Control Spaces and Policies ● High-Dimensional State Spaces & Multimodeling ● Decentralized/Distributed Control ● Hierarchical Control ● Team Theory ● Control Algorithms - Explicit Adaptivity - Implicit Adaptivity ● Control Processor Architectures ● Enhanced System Qualities Functionally Structured Distribution	Seek innovative approaches to the solution of large scale and distributed control process problems through the open solicitation of research proposals. Invite quantitative comparison of traditional approaches to large scale and distributed control process theory with more innovative approaches, such as the Functionally Structured Distribution through solicited research.
Estimation/ Decision Theory	1. Criteria to provide executive command authority with reliable information to initiate, interrupt, or terminate the operation of the SDI system.	1. During boost and post boost, coalignment of attitude sensors for the BSTS and SSTS vehicles will have to be that of the weapons platforms.	1. BM/C^3 research is essential in resource, that is, intercept weapons, allocation in an optimal way.
	2. Means of determining and maintaining the "health" of the BM/C^3 system, such as fail safe, fail determination, and other means, for all essential elements such as sensors, data processors, communication links, intercept systems, etc.	2. SSTS will "act" in a totally responsible manner under human authority during a battle.	2. Research is necessary into the development and definition of robust SSTS BM/C^3 functions.
	3. Raising to a much higher level of development the techniques for optimal resource allocation.	3. Levels of initiation of SDI engagement, withdrawal, and conclusion will be defined and executed on the BM/C^3 platforms.	3. This research must include different elements of each SSTS BM/C^3 platform such as sensors, attitude references, control devices, data processors, etc., and fail soft techniques must be developed in research.
		4. The same or similar BM/C^3 approaches will be implemented in the latter threat target.	4. All of the above must also be BM/C^3 research for the terminal battle management platforms.

TECHNOLOGY	ISSUES	ALTERNATIVE APPROACHES	RECOMMENDATIONS
Data Management	Within the SDI environment where: a. Data resides at distributed sites b. Many of the sites are constantly moving c. The sites face a hostile environment d. Update and retrieval rates are high e. Performance, reliability, and security are critical. The primary issues of data management are:		
	1. Data Allocation	1. In a highly dynamic network, such as in the SDI case, the assignment of database fragments to network nodes must be dynamic.	1. Algorithms for the dynamic allocation of data in a highly dynamic network need to be developed.
	2. Data Consistency	2. Two alternative architectural approaches to enhanced availability and performance are: o Multiple local databases, coordinated by higher level BM/C^3 functions o Single distributed database, with emphasis on availability and performance.	2. Database approaches that trade absolute data consistency for enhanced availability and performance need to be investigated.
	3. Performance	3. Performance requirements must be determined. System functionality as well as its performance must be verified. Experimentation is critical. Certain data-intensive functions may demand the use of special-purpose hardware and/or software.	3. Competing SDI architectures need to be refined to the extent that reasonably tight and credible bounds on performance requirements can be deduced. Performance verification methodologies and tools need to be developed and should incorporate experimentation to demonstrate performance on actual hardware and software where feasible. Methods for improving performance in the context of the data-intensive SDI BM/C^3 functions need investigation.
	4. Security	4. Two alternative approaches to security are multilevel security and system high operation. The 1982 Air Force Summer Study identified three near-term approaches to multilevel secure data management. The approaches were based on the logical, physical, and cryptographic separation	4. Multilevel security versus system high operation need to be analyzed for trade-offs. The applicability of the approaches to multilevel secure data management that were proposed in the 1982 Air Force Summer Study needs to be examined.

TECHNOLOGY	ISSUES	ALTERNATIVE APPROACHES	RECOMMENDATIONS
Data Management (Continued)		of data of different security levels, respectively. The Study also pointed out that the inference of classified data from unclassified data is a very difficult problem that will eventually have to be addressed.	The threat that inference poses in the SDI environment needs evaluation.
Networks	1. End-to-End Error Control	1. a. Datagram Service b. Virtual Circuit	Initial Analysis (provide reasonable approximation) • Baseline assumptions must be made to provide a realistic contex for networking research • Communications requirements must be determined • Communications architectures must be derived • Communications network subproblem should be defined representative of the overall problem. Develop Algorithms/Protocols • End-to-End error control protocols must be designed. • Static and dynamic routing algorithms developed for comparison, and use of flooding for key message classes investigated. • Mathematical algorithms derived for flow control then related to parameters for protocols. • Topology update algorithms developed and optimized • Develop interface with ARC to compute message arrival rates and network topology for input to communications network simulation. Tested Evaluation
	2. Routing	2. a. Optimal quasistatic routing b. Optimal dynamic routing c. Limited flooding for high priority messages under conditions of stress	
	3. Flow Contol	3. a. End-to-End windows b. Link-to-Link windows c. Windows computed by combined routing and flow control algorithms	
	4. Failure Recovery and Topology Updating	4. a. ARPANET algorithm b. Shortest Path Topology Algorithm	

TECHNOLOGY	ISSUES	ALTERNATIVE APPROACHES	RECOMMENDATIONS
Communication	1. Network Security	1. Security requirements analyzed and security architecture determined on basis of requirements. Techniques for remote key distribution and network access control applied. Integration of security mechanisms into communications protocols.	1. Establish security requirements and policy based on defined threat scenarios and identified communities of interest within the system. Use this to design the network security system.
	2. Coding and Waveforms	2. Investigate: ● Waveforms for modulation ● EDAC coding to maximize throughput ● Data compression techniques ● Data reconstruction techniques ● Approaches to message piecing	2. Determine a family of waveforms, including sphere of application, to modulate expected signal types. Define EDAC codes and investigate data compression, reconstruction, and message piecing.
	3. Nuclear Effects and Hardening Design	3. Postulate expected nuclear scenarios and standardize possibilities. Access data bases of atmospheric models and apply to SDI Analyze SDI radio and SATCOM links to develop anti-scintillation/anti-jam criteria. Develop criteria for expected EMP and direct attack threats for application to designs.	3. Develop a range of nuclear environments for postulated engagement scenarios. Analyze communications links within them to establish performance/design requirements. Established architectures then subjected to nuclear effects analysis. Antenna hardening technology should be developed.
	4. Defense/Offense C^3 Network Coordination	4. Requirements for defense/offense coordination must be established early. Appropriate interface points to the WWMCCS and MEECN networks identified if needed.	4. Establish requirements for defense/offense BM/C^3 coordination.
Computer Systems	1. Development of computer systems architectures to melt performance requirements.	1. Design to strawman problem based on allocation of system requirements on platform basis. A design that skews processing requirements allocation to ground segments. Approaches to the following subissues: ● Throughput needs ● Memory Size	1. High performance space architecture ● Study of parallel/distributed architectures ● Special function processors ● Data flow, control flow ● Nonvolatile memory ● Computing requirements ● Simulation Facility

TECHNOLOGY	ISSUES	ALTERNATIVE APPROACHES	RECOMMENDATIONS
Computer Systems (Continued		● Algorithm definition ● Software design specification ● Coordination of software/sizing/ algorithms ● Automated software development and test tools	
	2. Reliability and fault tolerance to meet system life and critical battle period reliability.	2. High reliability components and reduced component count through VLSI. Improved reliability prediction methods. Reliable component manufacturing, packaging, assembly, and test processes. Design to reduce susceptibility and test and redundancy to detect and recover from errors.	2. Fault tolerant technology Reliable processes
	3. Interconnecting networks at several levels.	3. Definition and design of interplatform communication processing. Development of high-performance automatic computer interconnect networks.	3. Communications software Multiple-fault tolerant networks
	4. Design flexibility to accommodate changes in threat, technology, and treaties	4. Automated tools and methodology to reduce the design cycle for both hardware and software. Use of general purpose computer architectures to the greatest extent consistent with performance.	4. Reduction of design cycle, hardware/ software via automation techniques. Flexible, high performance processing elements.
	5. Coordination with other SDI processing requirements to ensure compatibility and commonality.	5. Subsystem interfaces designed for compatibility and efficiency.	5. Coordinated analysis of platform functions.
	6. Radiation hardening	6. Design considerations include VLSI, shielding weight, radiation effects, lack of commercial interest. Special memory protection, software recovery techniques and nonvolatile memories.	6. Leverage VHSIC and radiation hardening technology research.
	7. Security from unauthorized access or denial of use due to intent or accidental interference.	7. Research in techniques for distributed computer security including verification simulation techniques.	7. Distributed computer security techniques for study.

TECHNOLOGY	ISSUES	ALTERNATIVE APPROACHES	RECOMMENDATIONS
Fault Tolerance	1. Coverage	1. ● Non-collaborative (temporal checks; diagnostics) ● Collaborative (redundancy codes; replication, comparison/voting).	1, 2 & 3. ● Attempt, through innovation, to simplify present computer system fault detection, identification, and recovery techniques while complying with the most stressing failure modes.
	2. Connectivity	2. Topologies for intra-node connection ● Bus ● Ring ● Mesh ● Tree ● Full cross-strapping.	● Develop distributed operating systems that are designed to operate with strigent real-time constraints. ● Develop architectures that combine parallelism and fault tolerance.
	3. Control	3. ● Adaption of transaction-oriented operating system. ● Modification of the Ada uniprocessor runtime package.	● Evaluate promising candidates (above) against stressing fault scenerios and BM/C^3 workloads.
	4. Software Fault Tolerance	4. ● Utilize inherently robust SDI structure. ● N-version programming ● Recovery blocks ● Other (Back-up S/W, robust programming).	4. ● Formulate metrics for software reliability measurement and techniques for reliability prediction; classify the effects and severity of software failures; and assess the resource requirements of software fault tolerant techniques ● Investigate fundamentally new approaches; apply best candidates to critical-core BM/C^3 software and demonstrate with fault tolerant hardware.
	5. Damage Tolerance	5. ● Shielding, stealth, evasion, self-defense, proliferation, replication, disperrsion, dissimilarity. ● Automation for damage assessment, diagnosis, and reconstitution. ● Radiation hardening.	5. ● Develop techniques for handling multiple simultaneous failures. ● Support architectural impact (sensitivity) analyses.
	6. Evaluation and Verification	6. ● Compartmented SDI system. Uncoupled segments of modest individual size. ● Flexible models, simulations, and analytical tools. ● Potentially abundant processing resources for built-in test aids.	6. Develop basic environment to provide computationally tractable models and simulations that still realistically represent complex, fault tolerant systems and operational scenarios.

TECHNOLOGY	ISSUES	ALTERNATIVE APPROACHES	RECOMMENDATIONS
Fault Tolerance (Continued)		● Incremental/cumulative system validation.	Utilize several fault tolerant computer system prototypes and fault scenarios to gain early experience with the models, simulations, and testing techniques required to evaluate the effectiveness of fault detection, identification, and recovery techniques.