AFIT/GST/ENS/89J-2

CSAR AIDE: DESIGN REQUIREMENTS FOR A COMBAT SEARCH AND RESCUE DECISION SUPPORT SYSTEM FOR JOINT RESCUE COORDINATION CENTERS

THESIS

Mark E. Bracich
Captain, USAF

AFIT/GST/ENS/89J-2

Approved for public release; distribution unlimited

AFIT/GST/ENS/89J-2

CSAR AIDE:

DESIGN REQUIREMENTS FOR A COMBAT SEARCH AND RESCUE DECISION SUPPORT SYSTEM FOR JOINT RESCUE COORDINATION CENTERS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirement for the Degree of

Master of Science in Operations Research

Mark E. Bracich, B.S., M.A.M.

Captain, USAF

June 1989

Approved for public release; distribution unlimited

Preface

The purpose of this study was to use an adaptive design methodology to define the requirements for a decision aid for use in the command and control of combat rescue resources by the Joint Rescue Coordination Center.

Many people made this thesis possible. I am indebted to my faculty advisor, Lt. Col. "Skip" Valusek, for his patience and guidance. His vision for the future of supporting command and control decision makers has definitely molded this neophyte analyst's approach to the same. I also wish to thank MAJ Dan Reyen, US Army (Ret), for keeping me on the path of reality when I wanted to wander off into the wilderness of research weirdness. It was also nice to have classmates who stayed as far behind as I did just to make me feel better--thanks guys. My deepest thanks, praise, and love go to my wife [redacted] for her understanding and support, and for being my best friend; to my children [redacted] and [redacted] for their tireless faith in me, and for being an always-present source of joy; and finally, to God for enabling me to do something that, at times, I knew I could not do.

Mark E. Bracich

Table of Contents

List of Figures

List of Tables

AFIT/GST/ENS/89J-2

Abstract

This thesis is an application of a methodolgy being researched at AFIT to define requirements for decision aids. The specific application of interest is Combat Search and Rescue Command and Control at the Joint Rescue Coordination Center.

It covers the current status of information management and control in the JRCC and recommends the development of an integrated decision support system (DSS). Such a system should be designed to aggregate information, provide the user with modeling and "what if" capability, and present data and model results in a manner which facilitates the decision making process.

An adaptive design methodology was used to capture requirements and ensure the design suggested meets JRCC needs. Modifications to the methodology are suggested. The result of this effort may be used as the cornerstone for a Statement Of Need for an automated Decision Support System to aid decision makers in the JRCC. Keywords: [illegible] [illegible], Decision Support Systems, Command and Control Systems, Theses. (SDW)

CSAR AIDE:
DESIGN REQUIREMENTS FOR A COMBAT SEARCH AND RESCUE DECISION SUPPORT SYSTEM FOR JOINT RESCUE COORDINATION CENTERS

I. Introduction & Background

CSAR missions must be successfully conducted to preserve and return to duty critical manpower resources, deny the enemy a source of intelligence, and contribute to the morale and mission motivation of combat forces [from AFM 1-1]. Additionally, CSAR may provide for the safety and protection of U.S. civilians, and (if applicable) designated foreign nationals.

(ALFA:5)

Combat Search and Rescue - The Mission

During the Southeast Asia (SEA) conflict, search and rescue (SAR) operations for American aircrews downed in hostile territory frequently took precedence over other ongoing warfighting activities. The American military (especially the air components of the various services) placed a very high priority on SAR. It often seemed that no price was too high when it came to recovering highly-trained, experienced aircrews and denying the enemy a potentially valuable intelligence source and propaganda tool. Early in the SEA conflict, the US, with virtually uncontested air superiority, realized the value of the search and rescue task force (SARTF)--a conglomeration of rescue helicopters and their fighter escorts, combat air patrol (CAP) packages, forward air controllers (FACs), airborne mission commanders (AMCs), and air-refueling tankers. Sometimes these operations were carefully thought out and executed, but due to the nature of rescue and recovery operations, they often "just happened." It was not out of the question for major air strikes to be postponed and the aircraft diverted to support rescue operations by providing air-to-ground firepower, CAP, or a diversionary air strike to draw enemy forces away from the recovery area (McConnell, 1985:70). While this bolstered aircrew morale, it did little to support what should have been our overall objectives in SEA. With the vast "air armadas" rallying to the rescue, enemy ground forces, now uninhibited by the deadly threat from the third dimension, were free to resupply and maneuver. Often the enemy would set up "flak traps" around a downed flyer, using him as bait to lure other aircraft into a deadly ring of antiaircraft artillery (AAA), surface-to-air missiles (SAMs), and intense small arms fire (Tilford, 1980:1, 42, 65, 67, 88, 92). Usually, our response to this problem was simply to apply more firepower.

Sometimes this "brute force" approach worked--often it did not. There are several cases where an entire rescue helicopter crew (Anderson, 1980:85) or several additional aircraft (Tilford, 1980:118) were lost trying to recover one man in the face of overwhelming enemy air defenses and ground fire. With the advent of more sophisticated enemy air defenses, it became obvious that quite often the SARTF would not be the best course of action.

Enter the "New Age" of Combat Rescue. In the late 1970s and early 1980s the Aerospace Rescue and Recovery Service (ARRS) began moving away from the SARTF concept toward more clandestine operations traditionally under the heading of "special operations." This culminated in the 1983 merger of Rescue and Special Operations under the newly formed Twenty-third Air Force (23AF) of the Military Airlift Command (MAC). From 1983 to 1987 Rescue took the "back seat" to Special Operations, politically and financially. The Aerospace Rescue and Recovery Service lost its "operational" resources and all but disappeared. The name remained only to denote the organization responsible for Rescue Coordination Centers. Several rescue units were closed and funding virtually disappeared for major rescue programs.

The future however, looks brighter; at least from the Rescue viewpoint. It seems that, in the face of concern about the future of USAF's role in providing vertical airlift support for special operations and the war-fighting major commands' reluctance to see rescue capability disappear, Rescue is making a comeback. The current "Concept of Operations for Combat Rescue" according to 23AF (Bridges, 1988), reads like a Special Operations job description.

Table 1.1. Concept of Operations for Combat Rescue

- Long range, clandestine operations
- Hostile airspace penetration
- Precise navigation to avoid threats
- Night/adverse weather
- Low level
- Thorough mission planning
- First pass insertion/extraction
- Search/reception by surface teams

The reader should note that this concept does not involve any "search" by aircraft. The loitering required to search for a downed pilot in most modern scenarios is prohibitive. The mission, however, still requires target (e.g. the downed crew) acquisition and identification. Consequently, the continued use of the term "Combat Search and Rescue (CSAR)" is still warranted, and is used interchangeably with "Combat Rescue" or, simply "Rescue." Distincti s will be made where necessary.

There are, obviously, many different scenarios that might involve the need for Combat Rescue; anything from a small recovery force supporting a quick raid on a relatively low-threat target (e.g. Grenada) to major involvement of several squadrons in a full scale conflict against a powerful adversary.

Command and Control of CSAR

The command and control (C^2) of Rescue resources throughout the spectrum of conflict is a key issue facing military planners today (Ziehm, 1988). This is evidenced by the DoD's pledge to publish joint doctrine on the subject in the near future and the current restructuring of the Rescue community. It seems the marriage of Rescue and Special Operations is ending in divorce, with the "Angel of Mercy" coming away meaner (more "special" capabilities) and richer (by way of fiscal attention) than when the union began in 1983. Closer to the task at hand is the current effort to construct an "automated command and control system" for the RCC as an add-on to MAC's Integrated Planning System (Marsh, 1989; Electronic Systems Division, 1988). Command and control of Combat Rescue is obviously a topic of great concern to many people in DoD. Hopefully, this thesis will provide some ideas to those responsible.

Figures 1.1 and 1.2 give an in-depth look at the current thought on the typical generic CSAR command and control relationships. Figure 1.1 is the structure most likely employed at the theater level while Figure 1.2 shows the relationships between the various components of a Joint Task Force (JTF).

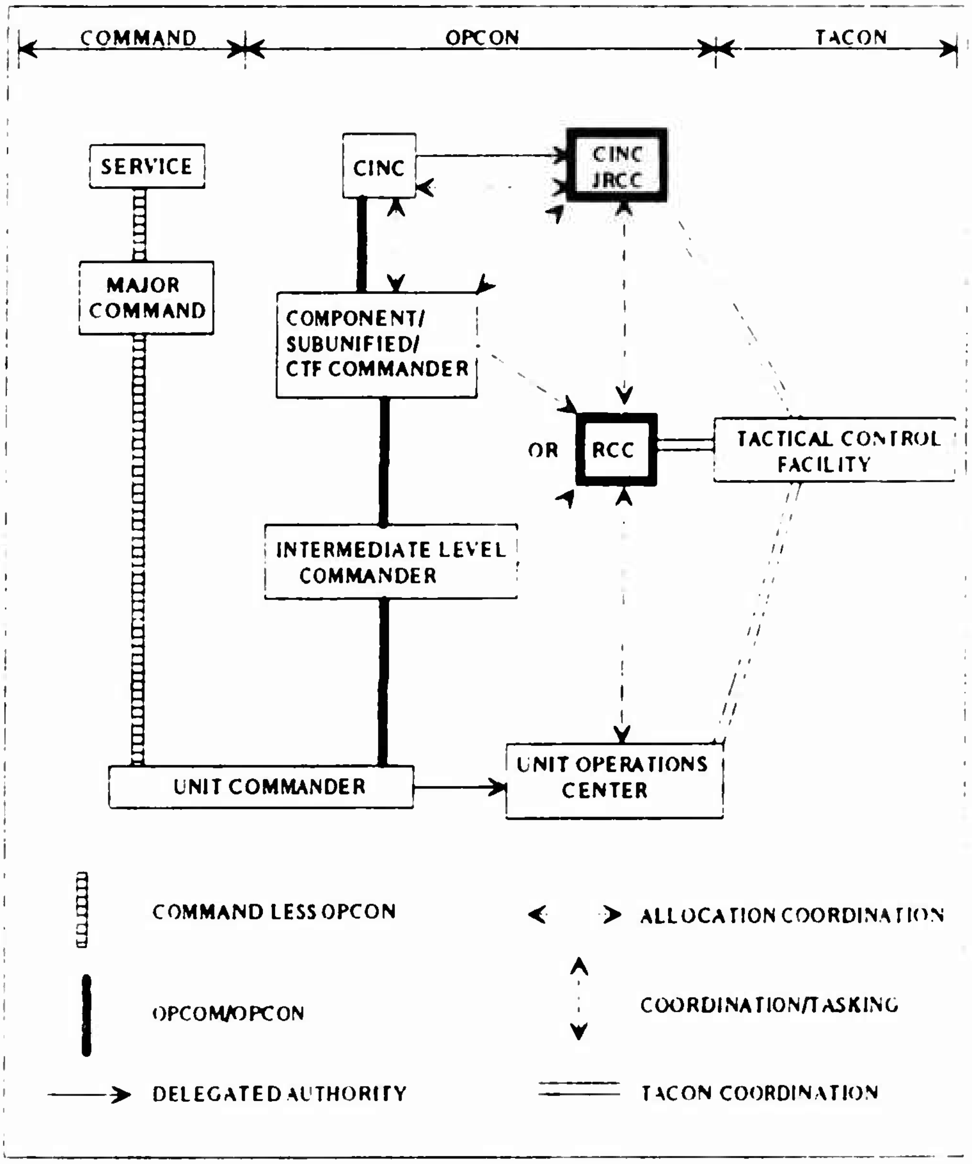
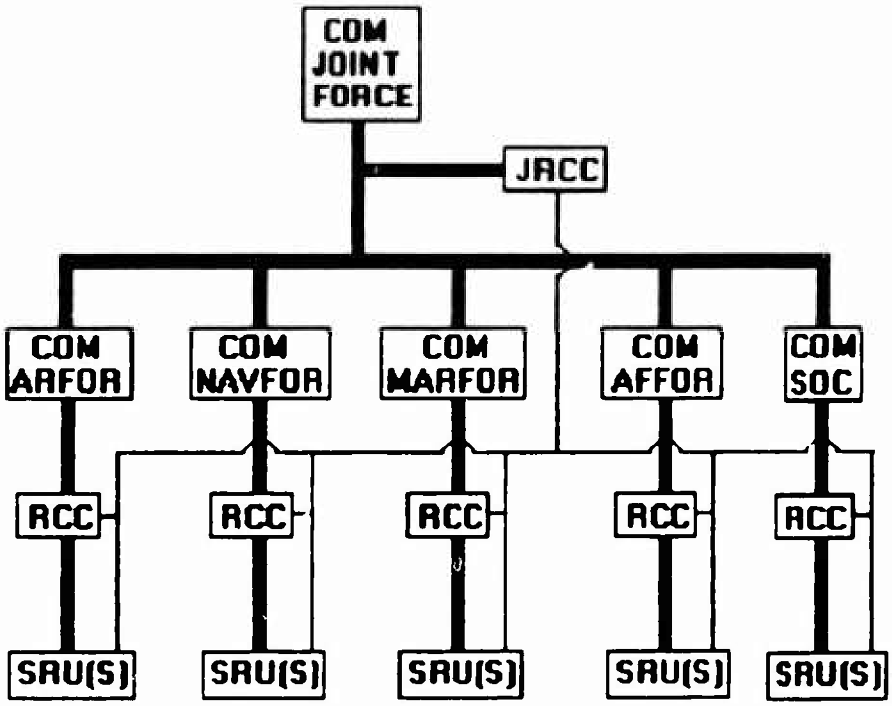


Figure 1.1. JRCC Relationships in the Theater C^2 Structure

Perhaps the best explanation of Figure 1.1 is offered by the accompanying text from Headquarters Military Airlift Command:

> The designated regional commander or theater CINC [Commander-in-Chief] (both the SAR coordinator--SC) establishes a command and control network that can effectively task and control resources allocated to a SAR mission. Policies and procedures for component and sub-unified commanders to provide resources are normally stated in theater directives, service and JCS documents, and the OPLAN/OPORD [Operations PLAN/OPerations ORDer] governing a particular CINC tasking. (Note: When an allied SAR system exists, US command and control arrangements should permit timely integration/coordination with the host.)
>
> COMMAND Command less OPCON [OPerational CONtrol] normally remains with the service of the resource involved. (i.e., for USAF dedicated SAR resources--MAC through 23 AF through commander combat rescue forces.; for fighter support--TAC [Tactical Air Command] through NAF [Numbered Air Force] through deployed wing; etc.)
>
> OPCON The SAR mission coordinator (RCC) normally exercises OPCON of resources assigned for each SAR mission. Control is exercised through the component SAR controller assigned to the RCC. Military commanders may retain control of their forces conducting SAR for their own forces. (i.e., In the case of USAF dedicated SAR resources--SC through SMC through USAF SAR controller through tasked unit commander., where the USAF controller is the MAC provided SAR controller. In the case of USAF fighter support resources--SC through SMC through USAF SAR controller through tasked unit commander., where the USAF controller is the TAF [Tactical Air Forces] provided appropriately qualified fighter liaison officer.)
>
> TACTICAL CONTROL Tactical control of SAR committed resources is normally exercised by the agency responsible for the overall coordination of activities occurring within a designated area (land, sea, or air). Typical tactical control facilities include TACCs [Tactical Air Control Centers] (AFFOR), CAMEs (Army), ATCOs/SOCs (NATO), etc.
>
> NOTE: SARDOs [SAR Duty Officers] and SARLOs [SAR Liaison Officers] enhance the command and control process by providing necessary interface to facilitate rescue mission coordination within the theater/area command and control network and between other Services [respectively]. For instance, within the TACC, the SARDO/SARLO can assist the RCCs and tasked units with tactical clearance coordination as well as keep these agencies informed of on-going or planned air/ground operations which may impact rescue operations.
>
> (Capacik, 1988)

Figure 1.2 illustrates how the JRCC fits into the overall command and control structure of a Joint Task Force (JTF).



LEGEND:

DARK LINE = OPCON IAW JCS PUBS 1 AND 2
(OPCOM IF COM JOINT FORCE IS A CINC)

LIGHT LINE = COORDINATION/REPORTING (RCCs MAY BE COLLOCATED W/ THE JRCC AND SOME PERSONNEL MAY BE "DUAL HATTED" SUCH THAT THE SERVICE RCC MAY NOT BE A SEPARATE ORGANIZATION. THIS WOULD REDUCE MANNING REQUIREMENTS, WHICH MAY BE CRITICAL IN A COMBAT ENVIRONMENT)

SRU = SAR REPORTING UNIT

Figure 1.2. JRCC Relationships in the Joint Task Force C^2 Structure (ALFA, 1988:1-11)

Of special interest to the author was the OPCON, or OPerational CONtrol, of the resources involved. In particular, who is making the key, day-to-day decisions affecting the combat recovery of personnel?

These decisions rest, for the most part, with the Joint Rescue Coordination Center (JRCC).

Although the theater Commander-in-Chief (CINC), or the Joint Forces Commander (JFC), is responsible for setting up his own C^2 network for CSAR, there is little doubt they will use the JRCC in its traditional role as the focal point for all CSAR in their areas of responsibility (ALFA:1-3). To better accomplish this mission, the JRCC is normally co-located with the Tactical Air Control Center (TACC) or, in some cases, with the Joint Operations Center (JOC).

At the JRCC there are people from each service employed as "SAR coordinators" who receive training in the management of SAR efforts. There is no formal training in the management of Combat SAR, although an effort is under way to provide this training at the U.S. Coast Guard's National SAR School (ALFA:3; Mathus, 8/24/88). JRCC personnel also act, in certain situations, as "SAR controllers". Such situations might include insufficient resources or expertise at the component (USAF, Navy, Army, or Marine) RCC, or the combination of the JRCC and component RCCs into a single unit (ALFA:1-11). As the component commanders exercise control of their CSAR forces through component SAR controllers (ALFA:1-4), these coordinators and controllers make many of the day-to-day decisions affecting the Rescue force.

Command and control (C^2) at the JRCC level involves, among other things, gathering and analyzing necessary information, prioritizing targets (i.e. downed aircrews, isolated Special Forces teams, or anyone else who may need rescuing), planning a recovery, coordinating and tasking resources to effect the recovery, monitoring those resources both in the pre-flight and in-flight phases of the mission, and coordinating any additional support required during the mission.

Unfortunately, the military still uses an inefficient and sometimes ineffective approach to conduct the time-sensitive, information-intensive planning and coordination of combat search and rescue missions. If one walks into a combat JRCC today, he will find that information management and control in the JRCC has not changed much since the Vietnam era. Historically, the decisions made in planning for the recovery of a downed pilot in hostile territory have been made based on information gathered from an extensive communications network strewn about in the proverbial smoke-filled rooms with people pouring over volumes of message traffic and intelligence reports. This information has been presented on grease boards, wall maps covered with acetate depicting intelligence estimates of the order of battle (updated manually by intelligence specialists), in regulations, manuals, and a few flowcharts and nomograms. Judgments and decisions that went into planning a recovery have, to a large degree, been based on personal experience. Inadequate intelligence estimates, poor support from and coordination with the rest of the Tactical Air Control Center (TACC), and even the inadvertent omission of a key planning factor or two have sometimes put crewmembers in precarious positions when tasked to execute these plans. Although this approach is not much different than what goes on in the war rooms of other combat planning cells, the CSAR planning process is unique.

What makes CSAR different from any other intense, short-notice planning function? The primary difference lies in the nature of the "target"; usually we find it, strafe it, bomb it, stop it, kill it, or photograph it, but nobody else has to find it, ensure its safety, pick it up, treat its wounds, and bring it home. Another aspect of CSAR is that the rescue force commander doesn't have operational command of many resources that may be necessary to effect a combat recovery. While this poses no problem for a quick, low threat recovery or a deep penetration clandestine rescue, resources for any other type of CSAR mission must be acquired from people who have other concerns as their primary mission, despite their deep interest in and commitment to SAR. Consequently, the planners must have near instant access to information on what resources are available, where they are located and what their status and capabilities are. They must also use these resources effectively to enhance the probability of success within the framework of broader military objectives, and efficiently to prevent wasting valuable planning and mission time or overburdening the TACC operators with unnecessary requests. The concern is maximum force effectiveness with minimum unnecessary expenditure while meeting the objectives of AFM 1-1.

The JRCC controllers must manage many different resources in many different ways. Resources include both dedicated CSAR resources (primarily tankers, helicopters, and pararescue teams) and non-CSAR resources (e.g. fighters, Forward Air Controllers, and just about everything else). In addition, the coordinators and controllers, and the planning cells that work closely with them, must manage a deluge of information in order to prioritize targets, plan missions, and coordinate, task, and control resources. This information comes in the form of messages, telephone and face-to-face conversations, TACC input, regulations and policy, status boards, radio traffic, and intelligence. There is very little automation in the JRCC, even the most mundane tasks (e.g. typing messages, filling out forms, and chasing down data) must be performed manually. The possibility of an important fact, observation, or insight being overlooked is quite high. The consequences could be disastrous.

With this in mind, the goal of this research was to use the tools of operations research to help the Rescue community do its job better. Personal experience as a Rescue helicopter pilot, exercise planner and controller for several major exercises throughout the Pacific theater, and instructor at the USAF formal school for Combat SAR led the author to a single conclusion: The effective and efficient use of information in the planning and decision processes used in the control and coordination of Rescue forces is <u>the</u> bottleneck in improving the way Rescue conducts business.

With the virtual elimination of budgetary support for anything new in the CSAR arena and the decreased emphasis on planning and exercising CSAR over the past several years, very little, if any, research has been conducted into making this vital mission safer, more efficient, and most importantly, more effective. While there are many factors bearing on this problem (e.g. new aircraft, better avionics, and more effective and efficient force structures) by addressing non-political, non-fiscal aspects of command and control, we can possibly see results much sooner.

Research Problem

The combat experience level of US military professionals is falling drastically. JRCC coordinators and controllers are no exception. There are very few, if any, currently on duty who have ever come close to managing combat rescue resources in a totally realistic environment. In field training exercises (FTXs), where the primary goal of rescue play is aircrew training, if anything happens in the RCC to create an unacceptable aircrew training environment, an "academic situation" is called and the RCC is basically left out of the loop. In command post exercises (CPXs), there are no aircrews. Consequently, the objective is the training and/or evaluation of the JRCC. Granted, the CPX planners do all they can to create a realistic environment, but the situation is very controlled. After-action reports reviewed and written by this author during his exercise planning days in the Pacific theater rarely failed to mention the inadequate capabilities of the RCC to accomplish its mission efficiently and/or effectively. The problem was not, and *is* not the people. They are dedicated, hard-working professionals. The problem is the process. The inability of the human mind to adequately store and *use* all the information required to do the job well is the root of the problem in the JRCC.

The best one can hope for today's controller to do in an environment like the JRCC is to find solutions that are "good enough", but not necessarily optimal. This concept, called "satisficing" by cognitive researcher Herb Simon (Simon, 1969:38), may no longer be appropriate in the rescue and recovery planning, coordination, and control process in the face of today's sophisticated threats and continued fiscal belt-tightening. In many scenarios, the JRCC needs to optimize (assign the best resource to a given mission, given the priority of that mission) to ensure it gives highly trained, valuable soldiers the best chance for survival and success.

The technology currently being used to support this life-and-death decision-making process in the age of TVs that fit on your wrist, cars that talk, and "a PC in every pot" is reminiscent of the old codger who refused to give up his outhouse in favor of "sum new-fangled terlet thang" because, after all, "the outhouse werks, don' it?" It may work, but how well and for how long before things get piled up too high? The influx of information into the JRCC coupled with the decreased time available to make accurate, effective decisions will eventually overwhelm the current system of filefolders, greaseboards, and antiquated communications systems. Today's science and technology offers a cornucopia of algorithms, methods, and systems, both hardware and software, designed to help decision makers make better decisions more efficiently. The problem lies in determining how science and technology can best be applied in this decision-making arena.

Research Objective

The primary objective of this thesis was to determine what tools from the Decision Sciences and Information Engineering disciplines could be integrated into a system and applied to CSAR command and control. Through the process outlined in Chapter 2, the author determined that the JRCC needed a Decision Support System (DSS) to integrate the necessary data and models required to assist the decision makers. The main goal then became to design that system, hereafter referred to as the Combat Search And Rescue Analysis, Integration, and Decision Environment, or CSAR AIDE.

The secondary objective was to investigate advantages and disadvantages encountered by the "user as designer" approach.

Limitations and Assumptions

The second objective was born of necessity in that there are no active combat oriented JRCCs in the continental U.S. The logistics of working with overseas users negated any other approach.

The major assumptions of this thesis are:

1) Rescue will remain a separate mission with responsibility for the accomplishment of that mission at the CINC, or Joint Force Commander, level;

2) CINCs will employ the JRCC in its historical role as the focus for C^2 of CSAR;

3) Controllers in the JRCCs want to do the best job possible.

Scope of Research

As of this writing, the rescue business is once again in a state of major reorganization. This thesis effort was not designed to be a panacea for all of Rescue's command and control problems. The objective is to alert the rescue community, especially the JRCC, to the advantages of modern toiletry and possibly provide decision makers with the design of a system that, if nothing else, can serve as a good basis for a Statement of Need for a command and control decision support system in the JRCC.

Overview

The following chapters will show how this thesis attacked the information-based problem and the decision processes in the JRCC. Chapter II focuses on the methodology used to bound the problem, the approach taken to suggest a solution, and the process by which that solution may come about. Chapter III discusses the requirements determined and proposed design of the DSS resulting from the application of the methodology. In Chapter IV conclusions and recommendations for further research are made. Supplementary, detailed information is found in the appendices.

II. Methodology

"The volume of information that staffs must process has increased many fold since World War II, and the time allowed for decision making has decreased many fold. As a result the requirements on the 'brain capacity' of commanders and staffs have increased vastly. To meet these requirements by simply expanding the administrative apparatus is fundamentally impossible...The only escape from this incompatible situation lies in the extensive application of automation, primarily computers...a 'man-machine' system is more perfect than 'man' alone or 'machine' alone...."

- Soviet General of the Army Shtemenko
(Wohl, 1981: 619-620)

Introduction

If the US is to meet the challenges posed by technology in planning, coordinating, and executing a rescue effort, we must exploit the technology available. The JRCC needs a system or tool that captures and integrates, for the decision maker, the vast amount of information available from a variety of sources. It must present information, options, and "what if" capability in the best possible manner for each decision maker concerned, allowing him to make the best decisions in the time allowed.

The approach most likely to meet these needs is a Decision Support System (DSS). This chapter will explain why. It describes what DSS is and what it is designed to do. Then it explains the requirements determination and adaptive design methodology used to formulate the requirements for CSAR AIDE.

Decision Support Systems

A Decision Support System (DSS) is a "*system* (manual or automated) that *supports* the cognitive *processes* of *judgment* and *choice*" (Valusek, OPER 652:7/11/88 [emphasis added]). Ralph Sprague characterizes DSS as "*interactive* computer based systems, which *help* decision makers utilize *data* and *models* to solve *unstructured* problems" (Sprague & Watson:8). DSS are best suited to unstructured problems where the decision maker (DM) would benefit from the ability to integrate analysis capability (models) and data through a "friendly" and effective Man-Machine Interface (MMI) while maintaining his own cognitive style. In other words, a DSS helps the decision maker in areas where he needs the support of models, algorithms, and databases but doesn't want them to get in the way of the process he goes through to make the decision. The purpose of a DSS is not to automate the decision process (for that would make it an "expert system") nor is it to impose a sequence of analysis on the user (Sprague and Watson, 1986:48). The purpose of a DSS is to *help* the decision maker use his own decision processes more efficiently and more effectively.

A DSS has three components: the database, the model base, and the Man-Machine Interface (MMI). These are merely technical terms for ways to manage, analyze, and interact with information. Databases give the user access to data and ways to manipulate the data to provide *information* (useful data). Models provide the user methods to analyze the data which can give the user *insight* into relationships between the data and how best to *use* the information.

The MMI allows the user to *interact* with the data and information in a way that is comfortable to the user.

While a working knowledge of databases is commonplace today, familiarization among decision makers with models and how to use them is usually limited to those having to wade through the output of the operations research branch or, if one is high enough up in the food chain, listening to a tailored briefing of the results. With a user-friendly MMI we can "get operations research to the end user" (Valusek, OPER 652:8/4/88). In fact, a well-designed DSS means that "end users" can be members of the lower orders in that food chain. No longer is scientific analysis limited to major long-term studies and quick-and-dirty responses to high-level taskings. DSS gives the middle and lower-echelon decision makers access to models and techniques that can help them be more effective in their day-to-day jobs. DSS, when coupled with a user-centered design methodology, is how analysts will get operations research off the shelf and into the hands of those who can use it to make daily decisions.

Concepts of Adaptive Design

> *The way of designing a DSS is different from that of a transaction processing system. A fundamental assumption in the traditional "life cycle" approach is that the requirements can be determined prior to the start of the design and development process. However, . . . DSS designers literally "cannot get to first base" because the decision maker or user cannot define the functional requirements of the DSS in advance. Also, as an inherent part of the DSS design and implementation process, the user and designer will "learn" about the decision task and environment, thereby identifying new and unanticipated functional requirements.*
>
> *(Alavi & Napier, 1984:21)*

Adaptive design is an approach to system design. It denotes an evolutionary process by which a system is developed to meet user needs as perceptions of the problem and its solution change over time. Adaptive design differs from traditional (or "life-cycle") design in that it is iterative. That means the user does not have to state all end requirements up-front, "freeze" requirements, and then live with whatever the builder delivers at a later date. On the contrary, the adaptive approach allows the user to be much more active in the evolution of the system and what it will and should eventually do, adjusting his "requirements" as his perceptions change.

User-Designer-Builder. This thesis defined player roles in terms of applying adaptive design to efforts designed to meet the needs of those operational military decision makers who do not possess an over-abundance of spare time and whose budgets are limited to providing them the capability to maintain the *status quo*. There are a few underlying assumptions beneath this particular assignment of roles. First, the user is a very busy person. While he may be quite capable of performing his own requirements determination, he just doesn't have the time. Second, as is more often the case, the user is *not* capable of performing that determination without some help. He may know "there must be a better way" but he probably doesn't know how to express those needs. Third, a designer is available.

There are three key players in adaptive design: the user, the designer, and the builder. The user is the decision maker the DSS is designed to support. The adaptive design methodology being researched at the Air Force Institute of Technology (AFIT) strives to ensure the design process has a minimal disruptive impact on the user. Because the user's time is considered a critical resource and must be divided among the processes of Information Requirements Determination (IRD, discussed below), development, and evaluation, the process employs a designer who works at the user's convenience under the assumption that he will have a maximum of three one hour sessions with the user in order to bound the problem (Valusek, 1988:107).

The designer is someone who can speak the languages of both the user and the builder. His education must include database fundamentals, analytical modeling techniques, and computer capabilities and MMI. His job is to accurately translate the user's perception of need into a requirements statement that is easily understood by the builder.

The builder is a computer scientist who accomplishes a technical analysis of the requirements, transforms the user's design into database, model, and interface technical design specifications, and builds the DSS based on the evolving needs of the user coupled with available technology.

Determination of Requirements. Valusek and Fryback "use 'information requirements determination' (IRD) to refer to the early process of developing a descriptive list of candidate requirements, detailing those requirements as much as possible over time, and then gaining an idea of their relative importance. [They] feel the label 'information requirements analysis' (IRA) refers to a later process of winnowing, reconciling, transforming and fully detailing the set of candidate requirements into a specification for a viable system" (Valusek and Fryback, 1985:107).

Information Requirements Determination (IRD). In adaptive design, IRD is accomplished by the user (or user and designer) gaining a thorough insight into exactly what the problem is, what its bounds are, and what processes are used to solve the problem. The method used to accomplish this in this research is concept mapping.

Concept Mapping. The theory behind concept mapping lies in education research. According to researchers Novak and Gowan:

> "Concept maps are intended to represent meaningful relationships between concepts in the form of propositions. *Propositions* are two or more concept labels linked by words in a semantic unit. In its simplest form, a concept map would be just two concepts connected by a linking word to form a proposition" (Novak & Gowan, 1984:15).

Figure 2.1 illustrates a simple concept map of "aircraft". Note that each propositional statement that includes the concept helps to increase the understanding of that concept.

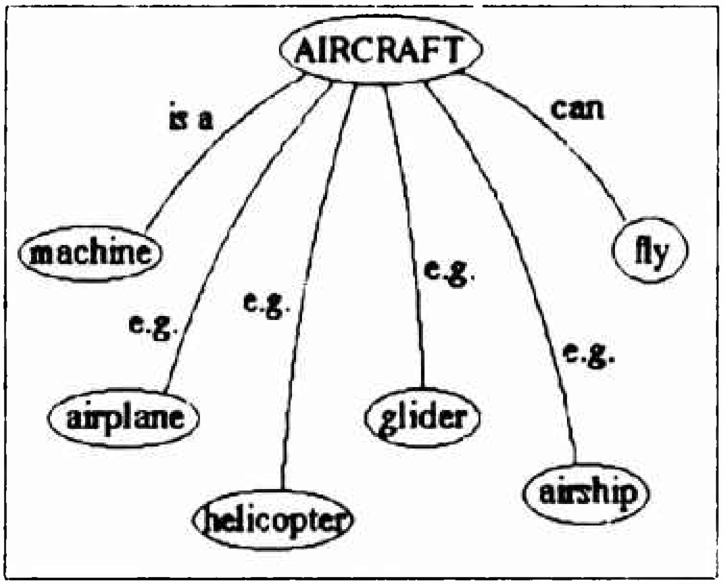


Figure 2.1. Simple concept map

Concept maps are not only a "knowledge representation" scheme, like so many other methods we read about in education and artificial intelligence literature where the objective is to illustrate what is known about a particular concept, but they also represent a powerful technique which serves as an easy-to-use-and-understand "knowledge acquisition" tool (McFarren, 1988:88). "Concept maps present . . . information in the same manner that man stores information in his brain thus making it easier for others to understand his cognitive process" (McFarren, 1988:13).

Concept maps made of or by different individuals concerning the same problem or process can yield significantly different relationships and concepts. Even maps of the same individual can change over time as the individual's perception changes or as he becomes more familiar with the subject (McFarren:101). Granted, this is a very superficial treatment of the power of concept mapping, but the simplicity of it will present itself shortly.

Concept mapping was used to bound and structure the problem, and to determine where to begin designing the system. It was also used as a means to gain insight into the decision maker's thought and decision processes.

Information Requirements Analysis (IRA). This stage of adaptive design is where the designer and builder turn the user's needs into an actual technical design specification--one key area at a time. The user need not be involved in the technical detail of IRA.

Design vs. Implementation. The reader should note that there are actually two "design" processes being conducted (Valusek, personal discussions: 4/89). The user design is geared to illustrate his actual requirements, while the builder design reflects what is currently "do-able". Figure 2.2 illustrates this relationship.

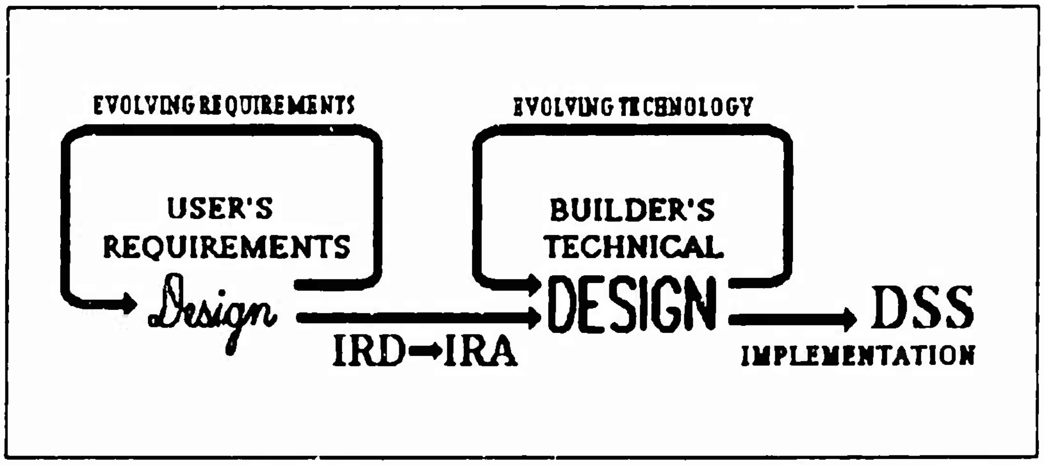


Figure 2.2. User *"Design"* vs. Builder "DESIGN"

Figure 2.3 shows the evolution of the desired capability disregarding technology available (represented by the storyboard to be discussed later) and evolution of the actual DSS implemented. The DSS evolves by applying technology and new understandings of user needs to expand or improve the current system.

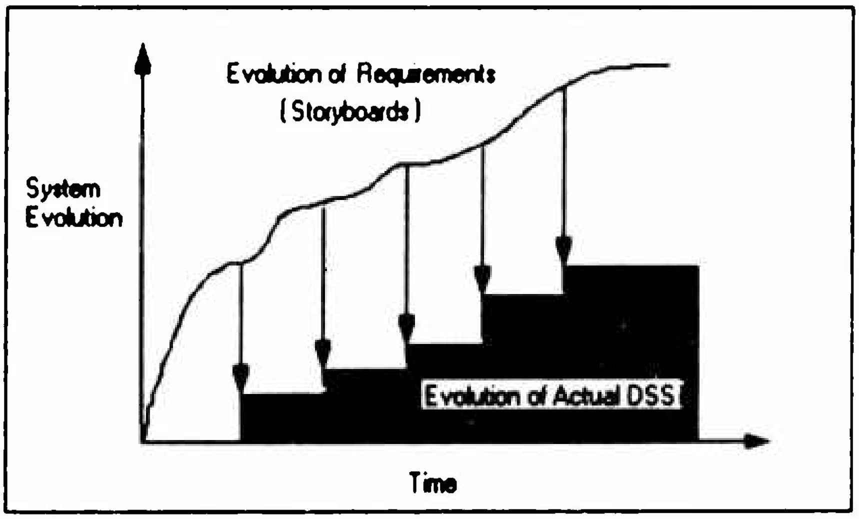


Figure 2.3. Adaptive design (Valusek, 1988:110)

It is very important not to let technology bound the actual user requirements. The objective of adaptive design is to state the user's needs and let technology meet those needs in the actual implementation of the DSS as it is capable to do so. Another benefit of this approach is that the user gets a usable (if not yet complete) and useful system much sooner than he would using the traditional "life-cycle" design methodology. The process used to apply adaptive design in this effort borrowed heavily from McFarren's Problem Definition Process to structure the user design process.

<u>The Requirements *Design* Process</u>

The process used to establish the requirements design of CSAR AIDE is based on McFarren's Problem Definition Process (PDP). PDP is "a strategy for determining the initial and iterative design requirements of a DSS based on user needs" (McFarren, 1987:88). Where adaptive design is a concept, PDP is a tool to execute that concept and is used to develop "the actual design specifications of the prototype DSS and [aid] its expansion into the full system" (McFarren, 1987:94).

PDP was chosen because it offers a very structured method of performing the "front end" of the adaptive design of a DSS which is by nature a rather unstructured process. In addition, McFarren claims, and this research confirms, that PDP can be applied to the "entire range of problems, from structured to unstructured" (McFarren, 1987:92).

Chapter 5 of McFarren's thesis provides a very detailed treatment of the Problem Definition Process and the theory behind it. The main points are summarized in Appendix A.

Where this approach differs from PDP is in the involvement of the builder. The approach taken in this thesis is designed to minimize the impact of the design process on the user in terms of both time and money. The objective, as illustrated in Figure 2.2, is for the designer to ensure the user has a good basic set of requirements (in the form of a graphic design, discussed below) before involving the builder. This eliminates some of the historic problems associated with the builder accomplishing the IRD, a "user-centered" function, from the builder's perspective. PDP brings the builder in at the beginning of the effort, therefore some adjustment to the process is required.

The steps of the methodology used in this thesis are summarized in Table 2.1. A concept map of the process is given in Appendix B.

Table 2.1. Methodology

0. The Flag.
1. Problem Description.
    a. Problem Definition.
    b. Task Analysis.
2. Graphic Design.
    a. The Feature Chart.
    b. The Storyboard.
3. Design Evaluation.

A detailed discussion of each step follows:

Step 0. The Flag. This is the indicator that alerts the DM that a problem or opportunity exists. It may be a gut feeling or an actual event (McFarren, 1987:98). McFarren says that the decision maker (DM) "may recognize that the problem is suitable for a DSS" (McFarren, 1987:98). Such clues as a highly unstructured problem or the requirement for dynamic visual displays, heavy interaction with the user, and models requiring evolutionary development may point to the applicability of DSS (Hippenmeyer and Valusek, 1989:16). However, in order to avoid the pitfall of walking around with a hammer and looking at everything as if it were a nail, this research suggests that this is a valid strategy with which to approach *any* tough decision making problem. If a valid solution method presents itself during one of the early stages, what has the DM lost?

Nothing. What has he gained? Hopefully, he has a better insight into his problem.

Step 1. Problem Description. "The first stage in designing a DSS is to identify the key decisions" (McFarren, 1987:98). Again, this is the first stage in *any* decision-making problem. This design methodology simply represents an approach to finding the proper tools to deal with unstructured or semi-structured problems. Concept mapping is used to define the problem and analyze the necessary tasks (McFarren, 1987:98-100). Since data analysis is a builder function, and the desire is for the user to have "done all his homework" before involving the builder, the data analysis (step 2c of PDP) is not accomplished as part of the user's design process.

1a. Problem Definition. The user, working with (or as) the designer, constructs a concept map (or a few iterations thereof) based on his understanding of the problem. The problem is defined by showing the key concepts involved and how they link together. The resulting concept map is a graphical representation of the problem space and its limits (McFarren, 1987:100-101) and can be used to relate the problem and needs to the builder when the time comes. From this point on, this map will be called the "domain map".

1b. Task Analysis. Here the user/designer constructs another set of concept maps, now concentrating on the user's perception of the decision process used to solve the problem. This is called the "process map". For highly structured tasks, the map might be reduced to a sequential listing or a flow chart of tasks required to solve the problem.

A "key decision element in a decision or problem solving process which is a feasible starting point from which to build a DSS" is called a "kernel" of the problem (McFarren, 1987:4). Using concept mapping and working closely with the user, the designer can extract a "well defined problem space, a sufficiently described decision process and a means to identify kernels" (McFarren, 1987:13).

At this point, an informal search of the process map is conducted for "candidate" kernels, those from which a DSS could be built. Sometimes the kernels can be recognized based on consistencies or inconsistencies noted in concept maps made by different DMs or by the same DM over time (McFarren, 1987:146). The designer then lists all possible candidate kernels and their role in the decision process (McFarren, 1987:101-103).

Step 2. Graphic Design. Feature charts and storyboards are graphic tools for accomplishing Sprague and Carlson's ROMC approach to identifying the necessary capabilities of a DSS (McFarren, 1987:105). The ROMC approach provides a guideline for focusing design efforts on "*[R]epresentations* to help conceptualize and communicate the problem or decision situation, *[O]perations* to analyze and manipulate those representations" (Sprague and Carlson, 1982:96) by "support[ing] intelligence [information gathering], design [option generation], and choice [decision-making] activities" (Sprague and Carlson, 1982:117-118), "*[M]emory aids* to assist the user in linking the representations and operations, and *[C]ontrol mechanisms* to handle and use the entire system" (Sprague and Carlson, 1982:96). ROMC is used to identify the required components, characteristics, and capabilities of the system (Sprague and Carlson, 1982:116). The ROMC checklist provides a method to ensure the needs of the user are related to the builder. Feature charts and storyboards provide a graphic representation of ROMC (McFarren, 1987:106).

The Storyboard. The storyboard details each component of the overall system design in the form of screen displays. Each "frame" of the storyboard is a snapshot of the DSS screen during a particular phase of the problem (McFarren, 1987:107). This snapshot of the Man-Machine Interface design shows the input/output format (in a very general sense), necessary information, and desired controls for the operation(s) shown (McFarren, 1987:108). Each storyboard frame also provides a forum for the user/designer to describe the underlying operations that would take place (McFarren, 1987:108). Each frame of the storyboard "capture[s] the decision process and help[s] to identify where in that process models appropriately fit" (Valusek, 1988:111).

The user/designer team can begin building the storyboard as soon as the process concept map is drawn and the task analysis has started to identify kernels. Each kernel found in the task analysis represents a possible frame or series of frames. The designer must strive to design screen displays that reflect what the user wants to see and be able to do to perform the given task.

Most of the DSS research community agree that animated storyboards are the ideal. Andriole and Fletcher investigated this with convincing conclusions. However, it is recognized that there will always be a need for a paper format. Valusek recommends the format shown in Figure 2.3.

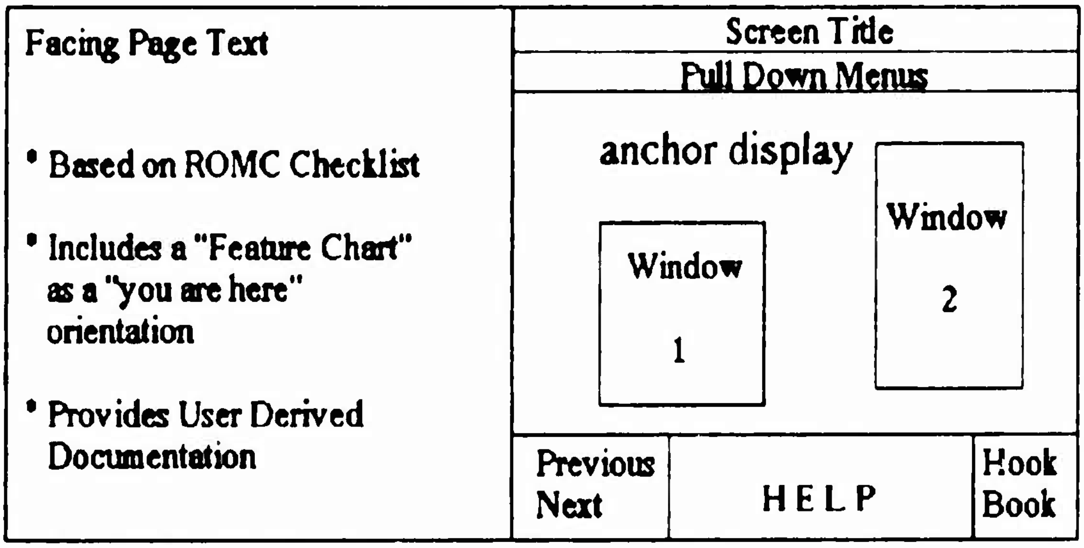


Figure 2.4. Sample Storyboard Layout (Valusek, 1988:108; Hippenmeyer and Valusek, 1989:12)

<u>The Feature Chart</u>. Seagle and Belardo give very good, specific guidance on the construction of feature charts as a builder's tool to convey his information requirements analysis to the user and designer and ensure he understands the user's needs (Seagle and Belardo, 1986:19). "The feature chart shows the features of the system with which the user interacts" (Seagle and Belardo, 1986:13). A feature chart is a graphic illustration and, since it shows the connectivity of the individual frames of the storyboard as well as the models and databases the user can access, it is a map of the overall design of the DSS. Its level of detail is defined such that

nothing is shown that the user doesn't interact with (Seagle and Belardo, 1986:19).

But which comes first, the feature chart or the storyboard? McFarren's thesis concentrated on the use of concept mapping, therefore he did not discuss the rest of the design process in great detail. It is the opinion of the author that, if a knowledgeable designer is involved, the process can work either way. Hopefully, the user and designer could develop both simultaneously, with each depiction feeding the other and growing accordingly. If the designer has well-defined kernels and is not quite sure how they will interact it would be wise to focus on developing frames for the storyboard. This could serve to strengthen the understanding that was missing in the process map. Putting a simplified feature chart first might be the course to take if the structure of the system is rather obvious. While the literature on feature charts classifies them as a builder's tool, the author often found the feature chart invaluable as a user/designer tool in deciding which kernel to storyboard next.

Step 3. Design Evaluation. The ongoing evaluation of both the design and the prototype, as well as the actual system, is very important to the success of adaptive design in meeting user needs. Such evaluations ensure "continued evolution of the system" (Valusek, 1988:107). The concern here, of course, is the evaluation of the user's design. It is very important to note, however, that at this stage of the design process the evaluation criteria are used to dynamically guide the designer's creative efforts rather than serve as a "static" checklist. There are times when a static evaluation is possible and advisable (before passing the design requirements to the builder, for example).

What to Evaluate. In the evaluation of the overall design, the user/designer team focuses on the graphic system representations--the feature chart and the storyboard--to ensure the proposed system will meet the user's needs. But what exactly are they looking for?

The basis for the evaluation criteria chosen for this effort was Sprague and Carlson's "4Ps" (Productivity, Process, Perception, and Product) approach, coupled with additional guidelines (Knittle, *et al.*, 1986) that focus on the development of the Man-Machine Interface. These criteria, included as Appendix C, provide the user/designer with a checklist of *what* their concerns should be in their design.

Several of the criteria cannot be measured quantitatively when "evaluating" a paper design, but the user can give good approximations on most of them such as "this is better than how I do it now" or "I can do that a lot better myself."

How to Evaluate. The method used to perform continuous evaluation was to keep a copy of the checklists that appear in Appendix C close at hand for easy referral. This served as a "sanity check" as the design process continued. When new insight presented itself, the author used the "evaluation" guidelines to see how best to attack the problem of including the information in the existing storyboard or designing a new frame. Often the evaluation criteria would trigger an idea or thought that was not readily implementable.

Consequently, some way of capturing the qualitative judgments about and impressions of the design, based on the evaluation criteria, was needed.

Valusek recommends event logging by the user in the form of an ever-present "hook book" designed to capture the user's thoughts on-the-fly as he works with the system (Valusek, 1988:109). The hook book is simply a series of cards or entries in a log that contain the date (for sorting chronologically), a label (for sorting by task), a brief description of the user's idea for improving the system, and a description of the actual circumstances which gave rise to the idea (Valusek, 1988:109). This too may seem more appropriate with a prototype or an actual DSS, but it has proven quite useful with animated storyboards.

In fact, event logging works quite well in the development of the paper products as well. While 3x5 cards are the suggested media for event logging, the author found that, while using Windows software to draw the storyboard and compose this thesis, he had access to a very useful feature. In the Notepad function of Microsoft Windows there is a ".LOG" feature which stamps the date and time after the last entry in a text file and allows the user to make his new entry while it is fresh in his mind. The current hook book for CSAR AIDE is included as Appendix H.

Iteration. The term "iteration" is used very loosely here. The timing of the iterative process was driven by information. As new information became available to the author, it would force an evaluation of the current design to correct previous mistakes or misconceptions. Each time a major revision took place based on "designer enlightenment" an "iteration" was accomplished.

Evaluation of the graphic design will undoubtedly lead to modifications of the feature chart and the storyboard. This may even offer additional insight and affect the understanding of the problem and the decision making process (especially when an evaluation is forced by new information), thus affecting the concept maps created in step 1. Each "design-evaluation" iteration brings the user closer to providing the builder with a set of requirements for a truly useful Decision Support System. When to bring the builder in is largely a question of economics. If the user and builder feel their design can be translated into a useful prototype (they must define *"useful"*) at a reasonable cost, then it's time to talk to the builder.

The designer must ensure the concept maps, feature charts, and storyboards all reflect the same "meaning" when he passes the requirements to the builder. His main concern is that the problem described by the user is the same problem being worked by the builder.

Summary

This chapter covered the highlights of Decision Support Systems, adaptive design, and a research methodology to support the user's side of adaptive design. Also covered was the difference between IRD and IRA. Concept mapping and storyboarding were presented as a viable method to perform IRD.

Chapter III will detail the application of this methodology to the problem of command and control decision making in the Joint Rescue Coordination Center.

III. Application to the JRCC Problem Domain

> *"...the increase in battlefield information rate brought about by modern weapons, sensors, and tactics requires selective but extensive application of automation to assist commanders and their staffs in reaching timely and appropriate decisions."*
>
> *(Wohl, 1981:618)*

> *"CSAR should be a well-planned effort so that the proper resources are placed into action to rescue our personnel."*
>
> *(Wilson, 1988:8)*

Introduction and Overview

This chapter deals with the application of adaptive design to meet the needs of the JRCC. As the main objective of this thesis was to establish the basis for a set of user-oriented design requirements, only the aspects of *"Design"* as illustrated in Figure 2.2 are addressed. The chapter will not cover the entire history of the five iterations of the user design process the author went through to get the current product. However, it will cover the flag, the lessons learned about adaptive design in performing the iterations, and the current iteration of graphic design and evaluation. This is accomplished using the format of the process outlined in Table 2.1.

Step 0. The Flag

The indicator that alerted the author to the existence of a problem in the RCC was his own frustration in dealing with RCCs as an exercise planner and controller, and as a rescue helicopter pilot flying simulated combat missions controlled and coordinated by the JRCC.

Several agencies were contacted when this research began in May 1988 and, while they expressed interest and provided valuable data, none had the resources or expertise to actively participate in this project as the user. Consequently, the author pursued the issue as the user/designer to illustrate the advantages to be gained by such an approach. This will be covered in more detail in Chapter IV.

Step 1. Problem Description

The first step was to use concept mapping to identify the key decisions to be made. The only RCCs tasked with the mission of becoming the JRCC during combat are located overseas. There are no stateside RCC coordinators whose primary mission is training for combat, hence there are no "expert" combat JRCC personnel readily accessible. Consequently, the author used his own experience as a rescue helicopter pilot, tactics officer, and exercise planner and controller, and personal interviews with past and present RCC controllers and rescue planners, coupled with his research into regulations, manuals, and training curriculum to model his perception of the problem.

1a. Problem Definition - Bounding the Problem. The first concept map of the JRCC's responsibilities and concept of operations (Figure 3.1) illustrates the diversity and complexity of the JRCC problem domain. The second concept map created (Figure 3.2) limited the problem domain to the planning and conduct of actual CSAR missions.

Initial Concept Map: JRCC Responsibilities. To illustrate where the JRCC fits into the overall command and control structure, refer again to Figures 2.1 and 2.2. The author's concept map of the JRCC's responsibilities and concept of operations (Figure 3.1) shows the importance of the decisions made by the JRCC in the overall framework of the theater's command and control structure.

This map took only two iterations to produce. The first map did not account for the importance of the component RCCs and their relationship to the JRCC; a major oversight on the part of the author. This was critical in that the first storyboard frames were based on that first map. The lesson learned: do not "jump the gun" spending a lot of time on storyboarding until you are reasonably sure you have the problem defined correctly. This is the reason the first iteration was quickly replaced.

Figure 3.1 shows the JRCC, as the single manager and focal point for all CSAR within the Joint Force Area of Operations (or Theater), is responsible for making recommendations to the Joint Force Commander (or CINC) on the apportionment of CSAR resources. The JRCC planners must also concern themselves with maximum economy of force in publishing their daily employment plan. In addition, they must provide assistance to and often accomplish the functions of the component (or service) RCCs.

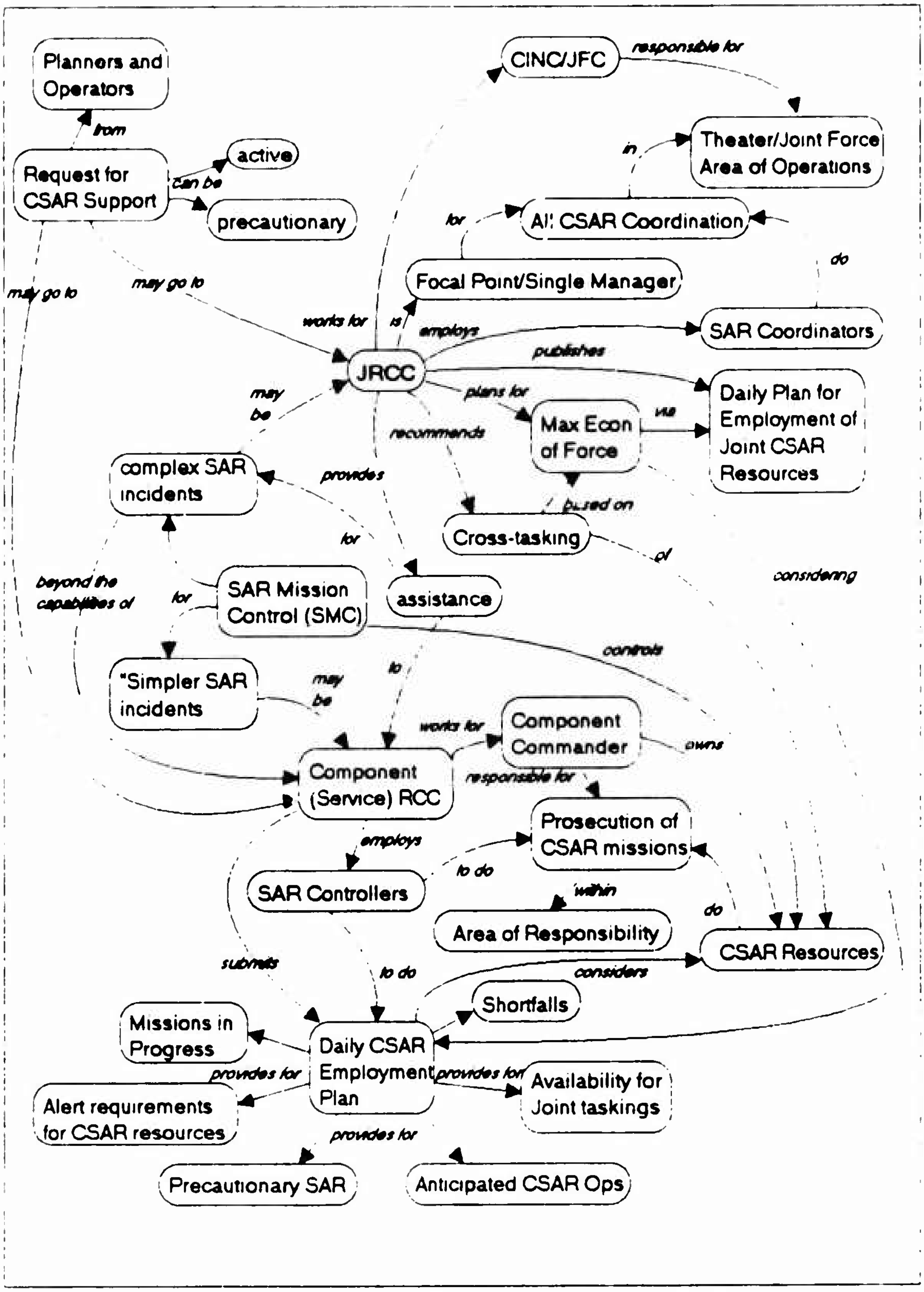


Figure 3.1. Concept Map of JRCC Responsibilities and Concept of Operations

Domain Map: CSAR Mission. Figure 3.2 is the author's latest concept map of his understanding of the command and control of a CSAR mission. Conflicting sources (e.g., ALFA and the National SAR School *vis-a-vis* Roark and WESTPAC RCC) generated an interesting question while the author was trying to bound this problem: How does one design a system for all JRCCs when each has its own forms, charts, and way of conducting business?

The idea was to come up with a product that, if implemented, could be used to train RCC personnel before they assumed their duties and then be used by them in the performance of those duties, enabling them to become more efficient and more effective at a faster rate than they would or could under the current system.

The author decided (after losing count long after nine iterations of domain maps) that the best course of action was to categorize the tasks under the major "function" headings shown in bold outline in Figure 3.2. These functions fall in line with the proposed training syllabus for CSAR at the Coast Guard's National SAR School (National Search and Rescue School, 1987).

Those functions that do not pertain to the actual conduct of CSAR are missing. The JRCC does many things to support the CINC or JFC (see Figure 3.1); these "upward functions" (e.g. advising the CINC on force apportionment) are not covered in this effort, but would obviously be a welcome addition to the DSS. The emphasis here is on the JRCC's functions as they pertain to an actual CSAR mission.

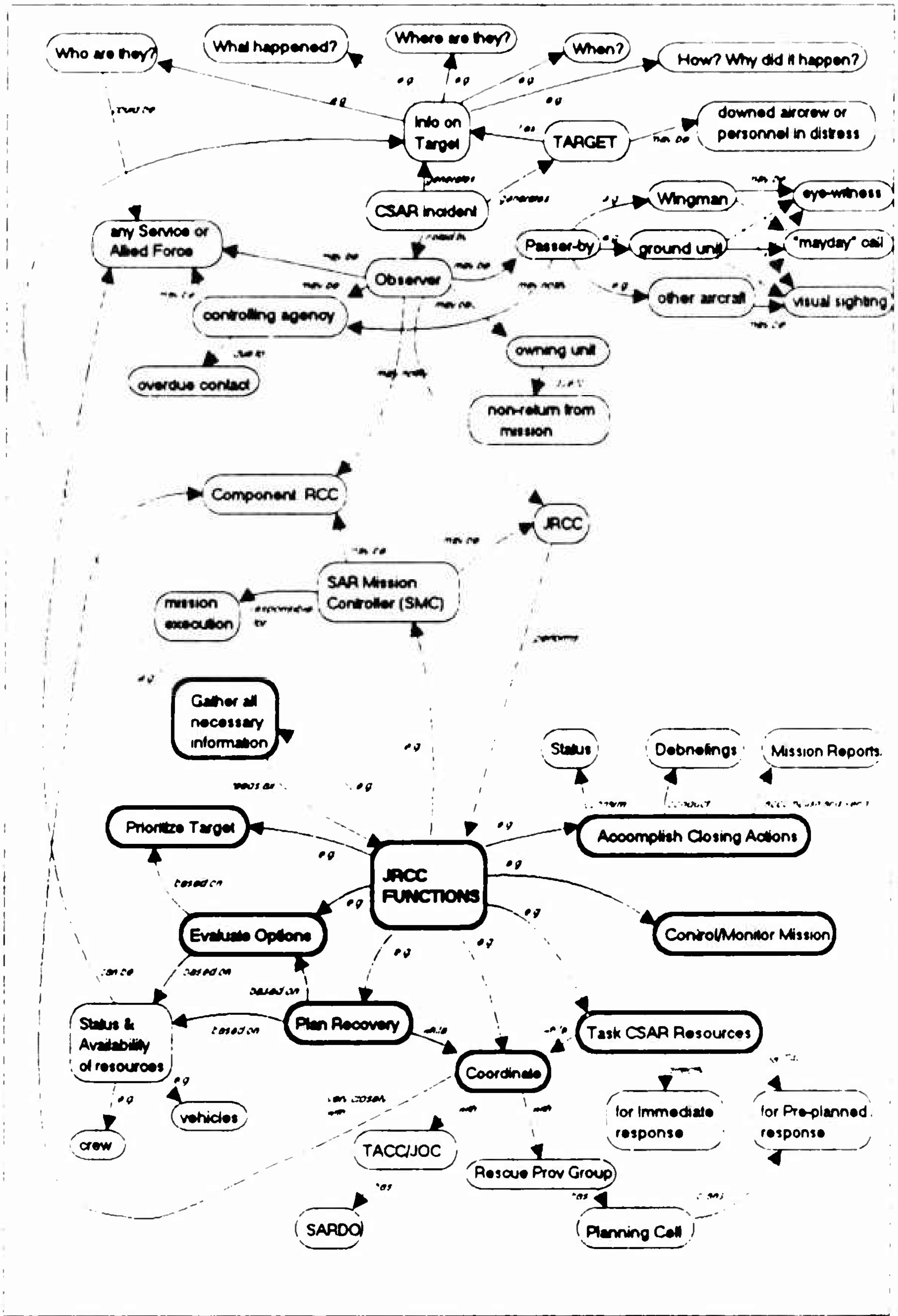


Figure 3.2. JRCC Domain Concept Map

Note that the CSAR incident eventually filters to the JRCC via one route or another, sometimes simply to be monitored, other times to be planned, coordinated, and controlled by the JRCC. Whatever the case, the map shows the diversity and intensity of activity in the JRCC.

Looking at the map from the viewpoint of one wishing to assist the decision makers in making timely, effective decisions based on accurate, up-to-date, complete (i.e. including all essential factors bearing on the problem) information, automation is the first obvious step. The need for a system to aggregate the information and present it in a logical manner could be met by a sophisticated Management Information System (MIS), however there are several areas of the concept map which point to the need for something more. This will be addressed in detail within the context of each specific function the JRCC must perform.

1b. Task Analysis - Identification of Key Judgments and Decisions. The focus here was to look at the individual functions of the JRCC as illustrated in the domain map (Figure 3.2) to find out what tasks the JRCC has to perform to accomplish its mission.

Process Maps: JRCC Functions. To gain a better understanding of the tasks involved in each of the JRCC functions, a decision process concept map was created for each function. These maps (and the storyboard made from them) are based on material collected from the US Coast Guard National SAR School, ALFA, HQ ARRS, the Western Pacific RCC, and the combined RCC located at OSAN AB

Korea, as well as personal experience and correspondence with RCC controllers (Roark, 1989). The process maps appear in Appendix D.

Each of these functions, itself a kernel, was examined for subkernels which can be considered candidates to expand into the prototype. A complete list of candidate kernels is contained in Appendix E.

Gathering Information. This obviously represents the most "data intensive" task accomplished by the JRCC. The process map (Figure D.1) shows the majority of information revolves around two areas: 1) the target and its environment, and; 2) CSAR resources. This information is used in some way in the performance of all other functions.

Prioritizing the Target(s). This process map (Figure D.2) was constructed based on the author's own perception of important factors to consider, coupled with other factors that could come into play. This concept map has not been validated yet as the author has not interviewed anyone who ever had to do such prioritizations and no source document could be found.

Evaluating Options. The evaluation of options must include the decision whether to plan and execute a recovery or wait for further information. If a search is required, what is the appropriate method: air or ground? Which methods and tactics are appropriate for this mission based on the threat level? Which resources are appropriate? Figure D.3 illustrates these relationships.

Planning the Recovery. This function overlaps the evaluation of options in that as the planner evaluates his options, he begins to formulate a plan. However there are a few specific actions to be taken, the most significant being turning the evaluated options into a set of valid assumptions and possible resources to task (Figure D.4).

Coordinate with Other Agencies. This is one of the most important functions of the JRCC (Figure D.5). The JRCC, through effective coordination, ensures mission accomplishment and minimizes unnecessary resource expenditure and duplication of effort. The key to effective coordination is efficient communications.

Tasking CSAR Resources. JRCC tasks CSAR resources through a very formalized process. For this reason, the author found it unnecessary to construct process concept maps for this function. Instead, the decision flow charts used by the JRCC to accomplish this function are included in Figure D.6. Tasking takes place through the network shown in the previous process map (Figure D.5).

Controlling/Monitoring the Mission. This is the "slow" part of a given mission as far as the JRCC is concerned (Figure D.7). If the mission requires clandestine operations, the JRCC may know nothing until the recovery is completed and the resources reach safety. If the mission is overt, JRCC follows the mission using the Mission Monitor & Flight Following Wall Chart, noting launch, rendezvous, and recovery times and locations. Arrangements are made with medical authorities to provide for inbound casualties. Any significant deviations from the original SAR plan are analyzed and requests for additional support are passed to the appropriate agency.

Accomplishing Closing Actions. Closing actions (Figure D.8) include conducting debriefings, confirming resource status, and preparing mission documentation.

Identification of Candidate Kernels. Appendix E is the aggregated task analysis list covering the tasks required of JRCC personnel. The list is not complete, nor are the proposed aids and/or methodologies the answer. However, this is the kind of product a user should expect to see from the designer very soon after working on the process maps as the data and models are starting to be formulated. This list provides the kernels which serve as the starting points (or goals, depending on the kernel) for designing frames of the storyboard.

Step 2. Graphic Design - The User's Requirements Statement

This is what user design is all about: describing a system, in graphic terms, that will truly meet the user's needs. What follows is a description of the fifth and final iteration of the design requirements for CSAR AIDE to come under the auspices of this thesis. The author is not so presumptuous as to think this is what the user would want to give the builder. On the contrary, this product must be evaluated by the actual users of the system, and this may take several more iterations. Keep in mind, that as there were no users available, this design is based on the *author's* perceptions based on his experience and study of the problem.

The Storyboard. As soon as the process maps began taking shape, and the task analysis had started to identify kernels, the storyboard was forming in the designer's mind. The process of drawing the frames actually helped the author make links between kernels that were not obvious from the concept maps. This lead to increased understanding of the decision process and modification of the appropriate process map(s).

The current iteration of the storyboard is included as Appendix G. Each storyboard is accompanied by an explanation based on Valusek's layout illustrated in Figure 2.4. Users of Microsoft Windows 2.x (Microsoft, 1988) will recognize the format of the frames. The Windows format was used because of its simplicity and controllability.

The Feature Chart. In this effort, the author chose to develop the storyboard and feature chart at the same time. After developing a frame for the storyboard, it would be added to the feature chart. This enabled him to keep a better grip on how the system was evolving and which direction to take next, which was necessary because he initially wanted to present an animated storyboard. The advantage of this approach is that the designer can show the user how the system "feels" during a decision making session, even if it's only for a small part of the overall process. Developing separate frames from different parts of the process concept map with no links between them does not give the user much sense of the "system", instead he sees separate little software packages each doing its own thing.

Of course, the feature chart was not limited to those frames already developed. Sometimes, just for a change of pace, the author "brainstormed" the feature chart to develop or include new features. The feature chart is displayed in Appendix F.

Step 3. Design Evaluation - The Driver for Continued Evolution

Using Sprague and Carlson's Productivity, Process, Perception, and Product measures, coupled with Knittle's MMI criteria, as the basis of the evaluation criteria led to many improvements in the graphic design. The size of the JRCC problem domain prohibited the author from ever finishing a "complete" set of storyboard frames. But then, this is the strong point of adaptive design: "Start small and grow". Several kernels, or key processes, went through three to five iterations.

As mentioned in Chapter II, the evaluation of the user's design in this effort was a *dynamic* process whereby the criteria served as a guideline for development more than as a checklist for evaluation. Consequently, any static evaluation by the author would be the mere affirmation of characteristics already designed into the system or included in the hook book (Appendix H). Therefore, what is required at this point is user involvement. If CSAR AIDE is ever to be used for anything, a bonafide user must evaluate not only the design, but the very basis of the design--the concept maps. If the user finds the concept maps adequately reflect his perceptions of the problem domain and the solution process, then the storyboard can serve as the basis for continued evolution. If not, the methodology presented can provide a direction to proceed.

Summary

This chapter covered the application of an adaptive design methodology to the JRCC problem domain. The methodology discussed in Chapter II and used to design CSAR AIDE worked rather well in identifying kernels and developing the graphic system representations. The methodology and the results of applying it as presented in this chapter and the appendices give the user a good basis for determining his requirements.

Chapter IV summarizes the results of this thesis. It covers the lessons learned about command and control decision aiding and adaptive design. It also discusses the future of CSAR AIDE.

IV. Conclusions & Recommendations

Introduction

This thesis addressed the application of an adaptive design methodology to develop a Decision Support System for use in the JRCC. Chapter I covered the Combat SAR mission and the command and control of that mission. It also discussed information management and decision making in the JRCC and the requirement for a Decision Support System. Chapter II explained what a Decision Support System is and how the adaptive design methodology was used to establish the requirements design of the DSS. In Chapter III the author discussed the results of the application of the process detailed in the preceding chapter. After evaluation and validation by the user, Chapter III, in conjunction with Appendices D, E, F, G, and H, could be used as a statement of need to relay the JRCC's requirements to a system builder. This chapter will cover the lessons learned and offer suggestions regarding command and control decision aiding, the adaptive design methodology, and the CSAR AIDE decision support system.

Military Command and Control Decision Aiding

Command and control is *not*, as one might come to believe reading the professional literature on the subject, simply communications. On the contrary, C^2 is "the exercise of authority and direction by a properly designated commander over assigned forces in the accomplishment of the mission" (Office of the Joint Chiefs of Staff, 1979:74).

The first lesson learned in this area was that many decision makers today do not view their decision making effectiveness as a separate, addressable issue. Effectiveness is looked at as a training problem, *not* an environmental problem. The focus of a great majority of "command and control systems" to date has been on the management of data, report generation, and communications connectivity, *not* on supporting the decision process. Military professionals must understand the immense importance *judgments* and *decisions* play in "the exercise of authority and direction". If we define *judgment* as the formation of an opinion, and *decision* as the commitment of resources, their importance is obvious. In order to make effective decisions in the employment of their forces, decision makers must make valid judgments about the state of their environment. They make these judgments and decisions based on data, which must be transformed into *information*, and an *analysis* of that information. While much of the military command and control decision process is viewed as an art, we cannot close our eyes to the advantages to be gained by applying science to the process. The application of scientific methods of analysis in a decision aid can allow the decision maker to concentrate on applying his skills to the *art* of decision making rather than the science of data aggregation, information analysis, and alternative evaluation.

Numbers on the battlefield are important. Technology on the battlefield is important. But ineffective judgments and decisions made in the employment of those forces can render the numbers helpless and the technology useless. The potential force multiplication offered by well-designed, well-built command and control decision aids that address decision maker effectiveness will, in the future, be as important as bullets and beans.

The second lesson learned in this area was that the Air Force must establish a clearing house for C^2 decision aids. The Decision Aids Working Group, a subpanel of the Joint Director of Laboratories Technology Panel on Command, Control, and Communications, has taken the lead in this area by working with the Army's Command and Control Microcomputer User's Group (C^2 MUG) at Ft. Leavenworth. Currently, the scope of their effort only includes the creation of a database for known decision aid efforts. The Air Force and its sister services need to empower a joint version of such a group with the management authority required to oversee the development, procurement, and support of military C^2 decision aids. The lack of a system, as it now stands, offers no controls over duplication of effort and no central repository for information on ongoing decision aiding efforts.

This problem was highlighted by the author's discovery, in April 1989, that the problem of information management and command and control of combat rescue in the JRCC was, in fact, also recognized and being addressed by HQ ARRS and Twenty-third Air Force (23AF).

Unfortunately, when the author began this effort in May 1988, he was unable to find anyone who was seriously interested in working with him on the problem. Although several organizations (USSOCOM/JSAG, USCENTCOM/J3, 23AF/XP, ALFA, and the USCG National SAR

School) did express interest in and provide valuable information for this project, none had the expertise or resources to actively participate in the effort as the user. The individuals contacted at Headquarters Aerospace Rescue and Recovery Service (HQ ARRS) when research began in May 1988 were unaware of any ongoing efforts .. this area or any interest therein. At the time, their focus was strictly peacetime SAR. Perhaps due to the impending reorganization of rescue, but unknown to the author, and apparently to the individuals contacted, the seeds of discontent with the current *modus operandi* of combat RCCs had, in fact, begun to take firm root within the rescue command structure.

ARRS, RCC, and 23AF personnel are now working diligently to ride the coattails of MAC's C^2 Information Processing System (IPS) by tailoring the features of that system to meet their needs. The 360-page System Specification which "specifies the performance, design, development, and test requirements" for IPS (Electronic Systems Division, 1988) contains not a single reference to search and rescue (other than defining "ARRS" in the glossary of terms). The current scrambling is an effort to establish requirements for CSAR C^2 that can be "tacked on" to this major effort. Unfortunately, while the IPS will address many of the problems RCC currently has with the gathering, aggregation, and display of information, it does not address decision aiding. IPS is a Management Information System (MIS), *not* a Decision Support System (DSS). Its focus is efficiency, *not* effectiveness. The JRCC needs a system which will provide *both.*

Given that the IPS isn't scheduled to be deployed until 1994 ("Air Force Selects...", 1989:51), perhaps this thesis can either spawn an interim system or be used to establish additional requirements for the IPS add-on. Ideally, the DSS would be implemented on workstations in the JRCC, and be fed by the MAC IPS and other Management Information Systems.

Adaptive Design

An adaptive design methodology was used to establish the requirements for CSAR AIDE. The approach is adaptive because it adjusts the design to the changing perceptions and increasing understanding of the user. Adaptive design is the most likely way to get operations research out of the hip pocket of the three- and four-star generals and into the hands of decision makers in the trenches. The adaptive approach, by being very responsive to user needs, helps the user overcome the inherent distrust many operators feel toward operations research. But for adaptive design to work, one or both of two things need to occur. First, we need sophisticated, easy-to-use development software to facilitate the "user as designer" approach. This must be coupled with an education program, perhaps as part of the Professional Military Education program, where potential users (decision makers) are educated to the capabilities and limitations of modeling and command and control decision aids. Secondly, more designers, such as graduates of AFIT's Strategic and Tactical Sciences program, need to be assigned to positions where they will be used as designers.

Role of the Players. There are many approaches with which to apply adaptive design. The key difference lies in the involvement of the various players and the roles they play at different stages of the process. The question of who plays what role can be answered by addressing three factors: education, time, and money.

Education, as used here, is defined as an awareness of exactly what it takes to perform an assigned role and the capabilities to carry it out. The user is, of course "educated" in his decision process (hopefully). The builder is "educated" in the technical aspects of system design and construction. What is needed then is someone to act as the designer who is "educated" in translating the perceived needs of the user into an accurate requirements statement easily understood by the builder.

Ideally, the designer could be a user who is educated as a designer. In the military, however, this is very rare due to the user's academic background or the operational demands on his time. A builder who is educated as a designer is often available in the form of experienced government contractors, but this is almost always expensive and time-consuming. The expense of involving an educated builder from the beginning of a project, perhaps even before the user really knows what he wants, is often prohibitive, especially for small, specific purposes. Often users would rather "do it the hard way" than get involved in a contracting paperwork nightmare.

As already hinted above, just who fills the role of the "educated designer" is largely dependent on the other two factors: time (who has the time to do it?) and money (what does the user want to pay for?). AFIT graduate students educated in this capacity are a resource users need to be made aware of.

The actual process or methodology used to apply adaptive design is dependent on what roles have been dictated by the situation. Due to the geographical separation between the user and the designer, the secondary objective of this thesis was to investigate the advantages and disadvantages of the "user as designer" approach to adaptive design, with the author posing as the user.

The main advantage to this approach was the feeling of control over the process and the design itself. This was offset, however, by the amount of time required to accomplish the work. If the user's time is as valuable as we think, then the user as designer approach will not succeed until sophisticated, user-friendly software development tools are readily available. Object-oriented software that allows the user to quickly create the screen displays he wants to see will be a great boon to user design.

In the future, as more users become computer literate, and as more sophisticated user-friendly development tools become available, the user will be successful in being his own designer. For the time being, however, it is suggested that he at least review the literature on adaptive design contained in the bibliography of this thesis. It would be advisable for him to involve someone who can act as a designer by contacting either the analysis shop supporting his command or AFIT.

If the builder is acting as the designer, as is usually the case, then the approach proposed by Hippenmeyer and Valusek is suggested. They provide a methodology (outlined in Appendix I) based on a combination of "specific mechanisms" of research in "adaptive design" and "rapid prototype" methodology from the *builder's* perspective using the framework of Pressman's "six stage iterative approach to rapid prototyping" (Hippenmeyer and Valusek, 1989:14). They tout their approach as an "integrated development scheme" using the tools of adaptive design (Hippenmeyer and Valusek, 1989:14).

Structure of the Process. A concerted effort to form a structure for the adaptive design process should be undertaken. This structure would define the tasks and functions to be performed by the various players at each stage of the design process. Such a structure is required if the methodology is ever going to be "sold" to DoD and contractors as a way of conducting the business of designing and procuring decision aids. With some modification, McFarren's PDP offers a good starting point.

The Future of CSAR AIDE

While JCS Pub 0-2 tasks each military service to provide forces capable of performing CSAR in support of its own operations, only the Air Force trains regularly in all aspects of Combat Search and Rescue, yet one of the primary missions of the Coast Guard is SAR, and the Navy and Army--with at least as many aircraft-- have as much of a need for CSAR as the Air Force. In fact, recent developments have placed the Navy in the position to be chief proponent of CSAR. They are leading the effort to publish joint doctrine on the subject (JCS Pub 3-50, draft due out in June 1990). The Army, due to Initiative 17, may eventually be responsible for all special operations vertical airlift, and many CSAR scenarios are considered "special operations." Consequently, this thesis investigated all four of these services to gain an adequate understanding of the current state of the art in CSAR decision aiding and how it is evolving. The result of that investigation was dismal. The issue is just beginning to be addressed in the MAC IPS add-on effort.

Today's technology has increased the burden on the Joint Rescue Coordination Center by multiplying the amount of information available in a given time and decreasing the amount of time available for planning a rescue mission in the face of a sophisticated, well-developed threat. This environment will quickly render useless the JRCC's current system of file folders and greaseboards. Fortunately, technology also offers many avenues for information management and analysis to support decision making.

Today, we find many decision makers throughout industry, government, and the military relying heavily on decision aids of all kinds--from simple tabulated data and graphs to elaborate customized computer software. The "life-or-death" importance of decisions made by JRCC controllers warrant the scientific analysis community's utmost effort to provide these decision makers with the best decision aids possible. CSAR AIDE is an effort in that direction.

User Involvement. As the author, despite his intense study of the subject, dares not claim to be a fully-qualified expert RCC controller, there is cause for justifiable caution in "buying" his design for CSAR AIDE. User involvement is required if CSAR AIDE is to be taken any further.

A user, or preferably a group of users, should evaluate the design of CSAR AIDE using not only the criteria set forth in this thesis, but also the common sense that comes from doing the job day after day. Before the design can proceed, the work accomplished to this point must be validated. The system must gain credibility if it is to gain acceptance.

Development of a Prototype. The design of CSAR AIDE, once validated, should be developed into a working prototype which can be further evaluated and tested in real situations. Two immediate applications would then be feasible. First, the system could be used during a major SAR exercise. This would surely generate improvements in the design. Secondly, if the prototype were deemed successful, it could be employed at the USCG National SAR School for the training of combat RCC controllers.

The author conducted an extensive software investigation to this end and found the Actor Windows programming language to be most promising for the MS-DOS environment.

Continued Expansion of Kernels. The storyboard and feature chart should continue to evolve to include the remaining kernels. The concept maps and the resulting task analysis (Appendices D and E) provide the designer a place to start.

Construction of Supporting Model Base. Several models are referenced in the storyboard. Unfortunately, except for the small "toy" systems built by the author for various classes in Operations Research, none are operational. The model builder will have to start from scratch as there as no models used for CSAR mentioned in the JCS or USAF Studies and Analysis model catalogs (Office of the Joint Chiefs of Staff, 1988; Air Force Center for Studies and Analysis, 1988).

One possible model not included in the storyboard is an evaluation scheme by which to monitor the effectiveness of the decisions aided by the proposed system *vis-a-vis* those courses of action taken without the use of CSAR AIDE.

This "self-analysis" module, once incorporated into the DSS, would track results of missions in which the controller used the analysis capability of the DSS and those missions in which no analysis was performed by the DSS, and make use of various decision analysis and multi-criteria utility assessment techniques to determine the effectiveness of the system. Most, if not all, of the inputs to this model would come from the JRCC's mission status/results reports incorporated into the DSS. The output would be used as another tool to evaluate the DSS, hopefully catching any flaws in its design.

Concluding Remarks

Decision Support Systems brought about through a user-centered adaptive design methodology give decision makers the capability to add new insight into their decision processes thereby allowing them to make better decisions. The design requirements presented here in the form of CSAR AIDE can indeed serve as the cornerstone for a Statement of Need for decision support in the JRCC.

Appendix A: McFarren's Problem Definition Process (PDP)
(McFarren, 1987:Ch. 5)

Step 1. The Flag.

- indicator that alerts the DM that a problem exists
- may be a gut feeling or an actual event
- problem suitable for DSS?

Step 2. Problem Description.

- identify the key decisions
- uses concept mapping to define the problem, analyze the necessary tasks, and analyze the data required to perform those tasks
- may take several iterations

2a. Problem Definition.

- user constructs a concept map based on his understanding of the problem.
- problem is defined by showing the key concepts involved and how they link together
- user and the builder use this map to 'negotiate the meaning' in the map
- resulting concept map is a graphical repesentation of the problem space and its limits

2b. Task Analysis.

- user constructs another set of concept maps concentrating on his perception of the decision process used to solve the problem.
  - -- called the "process map"
  - -- for highly structured tasks, the map might be reduced to a sequential listing of tasks required to solve the problem
- user and builder perform another negotiation of meaning
- conduct an informal search of the process map for kernels from which a DSS could be built.
- designer lists all possible kernels and their role in the overall decision process.

2c. Data Analysis.

- the process map is used to establish the input and output specifications of the system
  - -- the type, format, and form of information required for each event/process in a decision element is identified
  - -- these are added to the process map

-- the same is then done for the output requirements
- designer compiles a complete set of data requirements which form the beginning of the system requirements

Step 3. Feature Charts & Storyboard.

- application of Sprague and Carlson's ROMC approach to identifying the necessary capabilities of a DSS.
- clearer, more flexible desciption of system
- Feature charts and storyboards provide a graphic representation of ROMC.

3a. The Feature Chart.

- provides the "process" aspect of the DSS
- shows the features of the system with which the user interacts
- negotiable graphic illustration and a map of the overall design of the DSS.

3b. The Storyboard.

- details each component of the feature chart
- shows the input/output format, necessary information, and desired controls for the operation(s) shown
- provides a forum for the user/designer to describe the underlying operations that would take place.

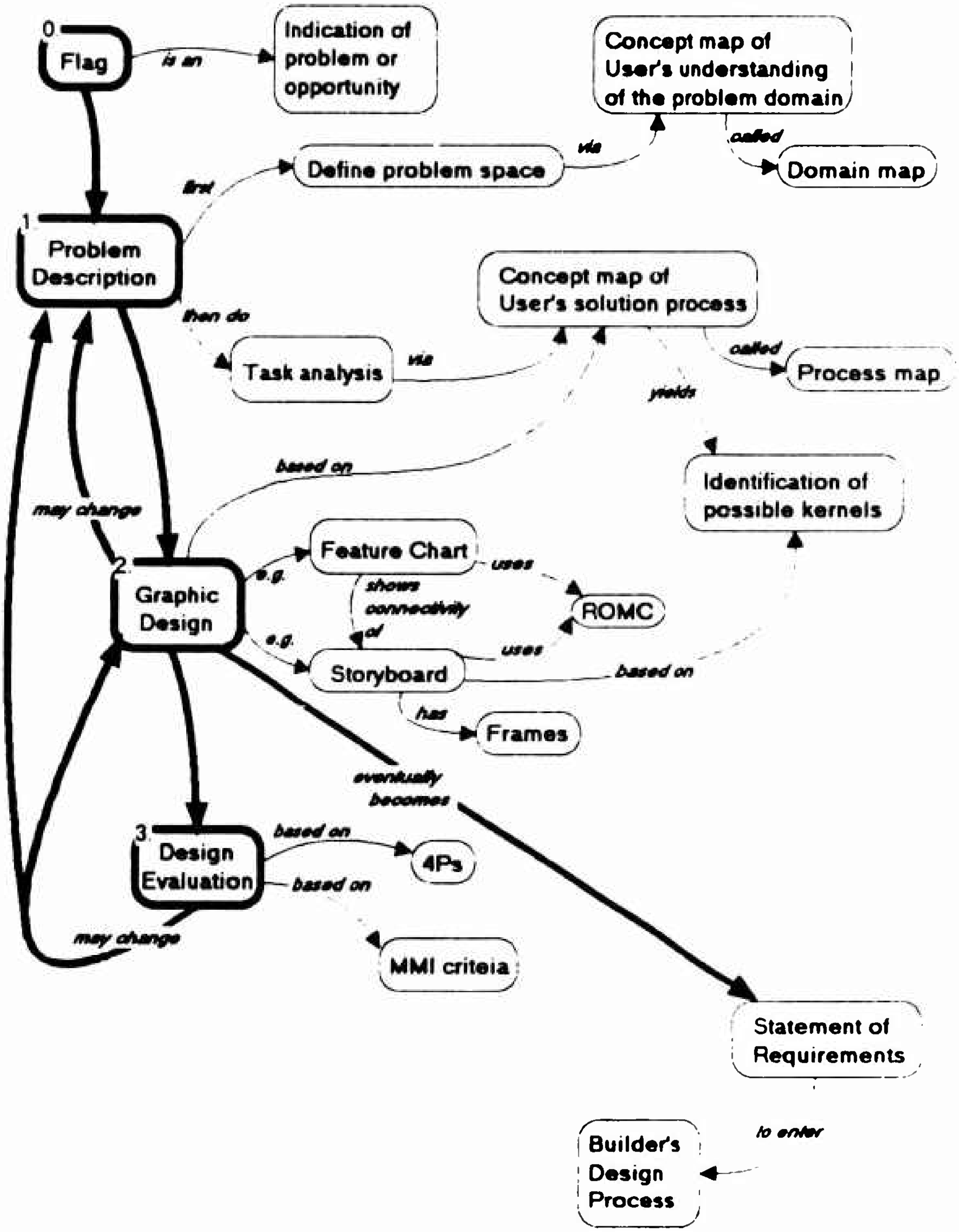
Step 4. Evaluation.

- recommends looking into a few approaches
  -- Sprague and Carlson's "4Ps" (Productivity, Process, Perception, and Product)
  -- Adelman's 3 interfaces (DSS/user; user & DSS/ organization; Organization/environment)

Step 5. Kernel Development.

- developing separate kernels and groups of kernels at different rates may be necessary based on the understanding of the problem and the technology available to support kernel development. This is especially true of large, complex DSS supporting large problems with many kernels.
- Small problem: can work on all kernels simultaneously.

Appendix B: Concept Map of Methodology Used



Appendix C: Evaluation Criteria

Part 1. The "4Ps" Approach to DSS Evaluation (Sprague & Carlson, 1982)

Productivity Measures (impact on decisions)

1. Time to reach a decision
2. Cost of making a decision
3. Results of the decision
4. Cost of implementing the system

Process Measures (impact on decision making)

1. Number of alternatives examined
2. Number of analyses done
3. Number of participants in the decision making
4. Time horizon of the decision
5. Amount of data used
6. Time spent in each phase of decision making
7. Time lines of the decision

Perception Measures (impact on decision maker)

1. Control of the decision making process
2. Usefulness of the DSS
3. Ease of use
4. Understanding of the problem
5. Ease of "selling" the decision
6. Conviction that the decision is correct

Product Measures (DSS technical merit)

1. Response time
2. Availability
3. Mean time to failure
4. Development costs
5. Operating costs
6. Maintenance costs
7. Education costs
8. Data aquisition costs

Part 2. Guidelines for MMI Evaluation (Knittle et al., 1986)

1. minimize worker effort

- user should only be required to perform work "which is essential and cannot be performed by the system"
- simplification of input (minimum keystrokes or use of pointing devices, voice/equipment recognition, synonyms)
- work done in the past should not be repeated
- min data entry redundancy (one entry updates all occurance of that item)
- operating system does as much as necessary to free worker
- recovery capabilities, audit trails, production and operations statistics, file protection, backup, error diagnostics and transaction logging.
- on-line help immediately available such that the user does not have to refer to a "manual" to solve a problem
- imbedded menus for users who have trouble with what he sees
- specific instructions on how to correct an error
- use understandable system messages (self-explanatory, relevant, specific, timely, helpful)
- all work should be capable of being performed on-line.
- screen layout match as much as possible the written data entry form

2. minimize worker memorization

- requires less training
- point and shoot
- "small number of steps and a small amount of key information, patterns to all techniques, and prompting the actions"
- "learning the system" should be "an incrementally extensible and hierarchical process" where the user is not required to learn anything non-essential to the task
- working with a small part of system gets results

- "layered interface": as user gets more proficient, system introduces new commands
- system "instructions or communications should be in task-related natural language"
- use consistent terminology in the user's vocabulary
- user commands should be simple and natural (and not confusing)

3. minimize worker frustration

- min response time, notify user when operation will take longer than 15 seconds, use audible signal and abort capability
- permit experienced users to bypass menus/prompts/other guidance techniques (linear structure for function selection screen presentation)
- if interrupted, provide review of last few functions
- choose help level, have hot key for help,
- layered help: 1=what to do now or options; 2=more detail on what user may do and any limitations on the user at this point; 3=cross references to commands or activities related to this help statement; 4=references to the user's manual or system documentation for extensive off-line investigation
- help function that shows hierarchical structure of the program in graphical form (avail hot)
- terminate or interrupt any activity at any time
- self-configuring, self verifying installation (fully useable system at startup)
- feedback for any action for which the results are not obvious
- tolerance for user errors (confirmation)

4. maximize use of habit patterns

- keep track of user patterns for various tasks, adjust to user needs and preferences

- "muscle memory" look at a certain spot and expect to see a specific bit of information (put info where user expects to see it)

- use of a backup key

- max use of single programming language

5. maximize tolerance for human differences

- again, track user profiles and respond accordingly when a given user logs on
  - covers both type of menu/prompts and input/output formats

- visual and audible "attention-getting methods"

- support procedural and non-procedural approaches (np=supporting statements in any order, referring to variables not yet defined)

6. maximize tolerance for environmental change

- support changes in hardware/software or changes in application as a result of new functional requirements

- "flexibility and expandability" (reconfiguration/reverification)

- transportable between various computers

- data tuning (optimum data storage organization) and data migration (moving frequently accessed data to storage locations where it can be quickly accessed should be automatic

7. notify users of problems promptly

- as soon as it occurs and tell what will happen if not corrected
  - audible warning for those who don't look at the screen much (experienced ones)
  - explicit confirmation of actions that could have serious consequences

- what's required to rectify the error?

- notify when file capacity is filled to within a certain percentage so the user can take appropriate action

8. maximize user control of tasks

- "user control the flow and sequence of work to the extent possible when there are no sequence-dependent activities"
- modify priorities of processing
- synonym table (already mentioned) to include user-defined terms
- macro definition capability
- default options for various tasks

9. maximize task support

- provide complete support for all tasks to get the job done
- provide communications capability with other sources/sinks of information

Appendix D: Process Maps of JRCC Functions

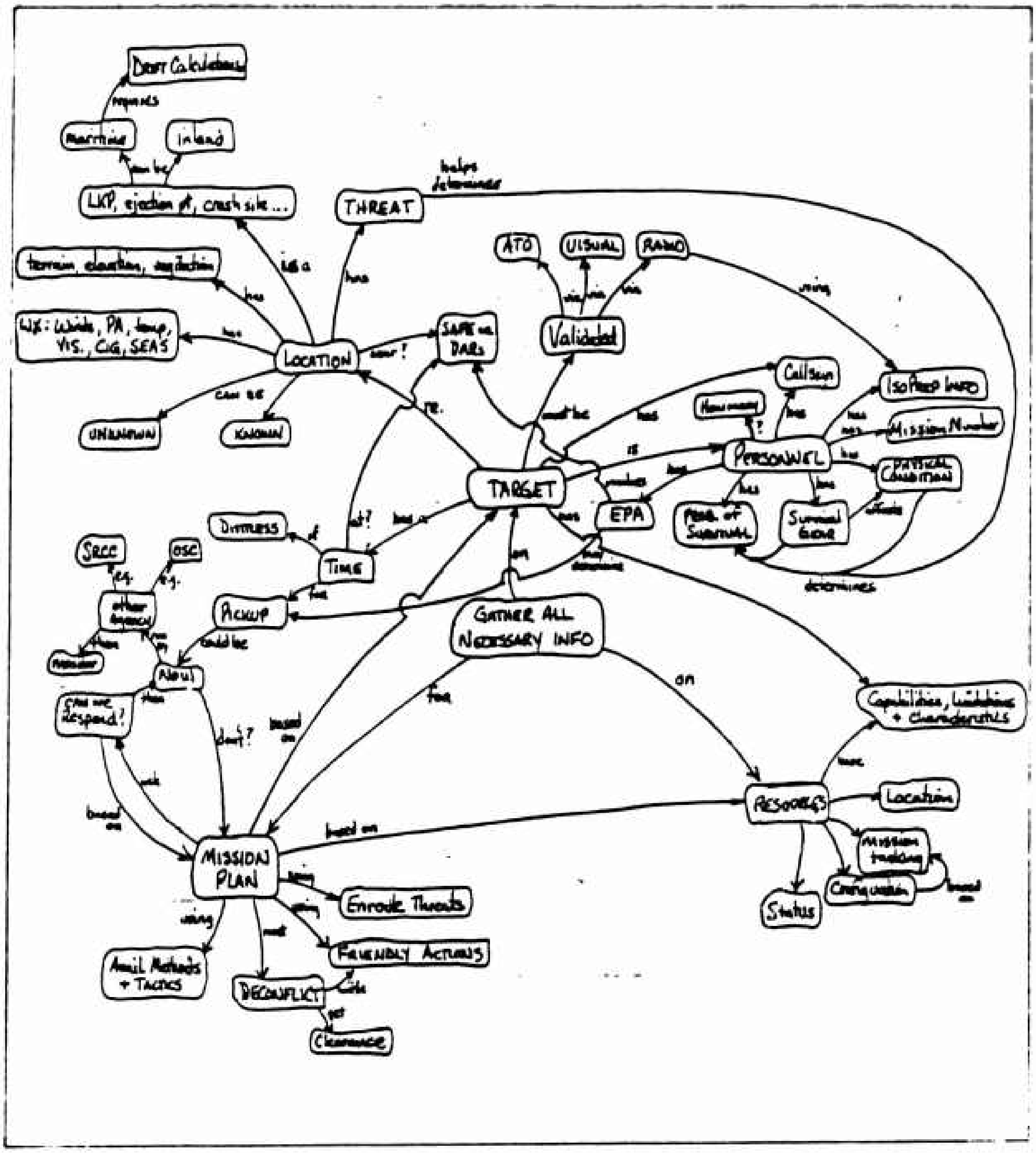


Figure D.1. Gathering Information

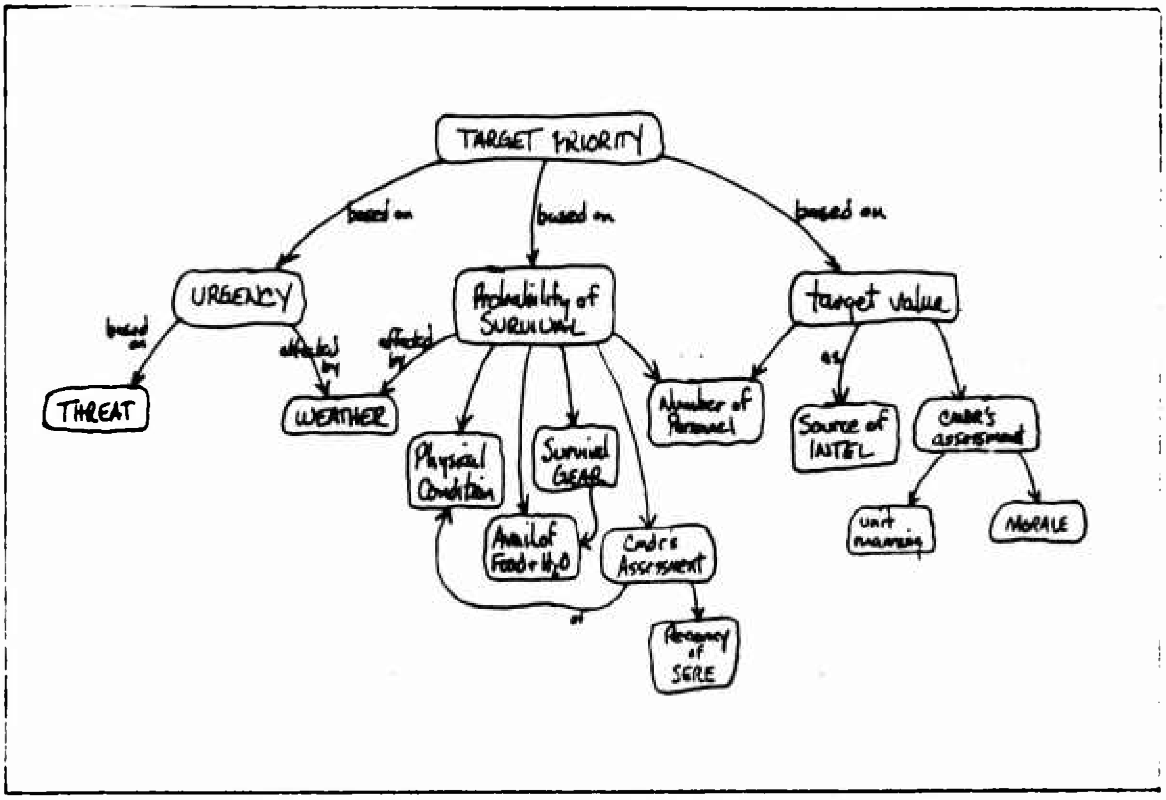


Figure D.2. Prioritizing the Target(s)

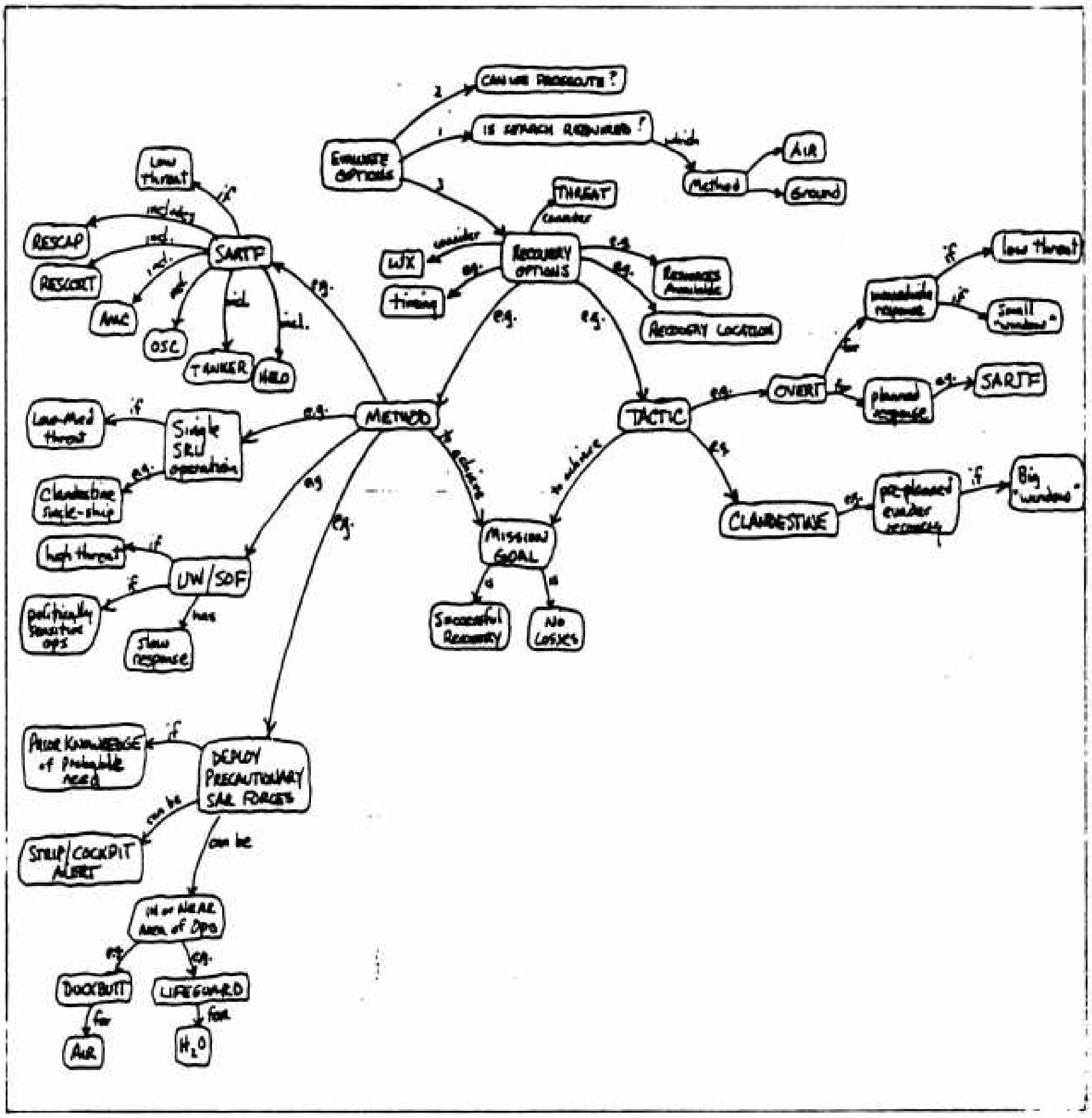


Figure D.3. Evaluating Options

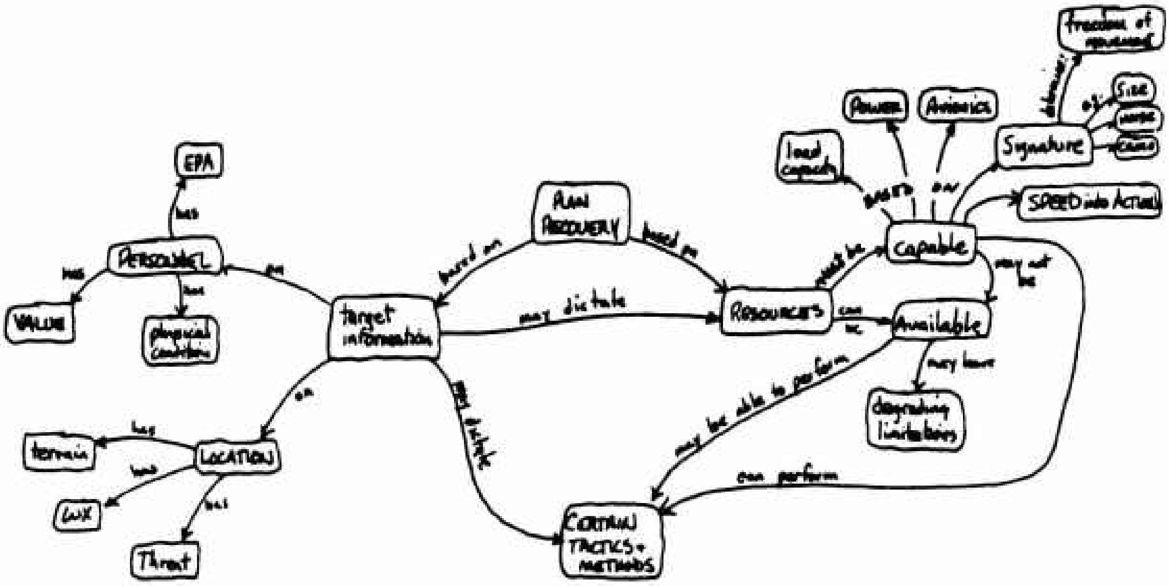


Figure D.4. Planning the Recovery

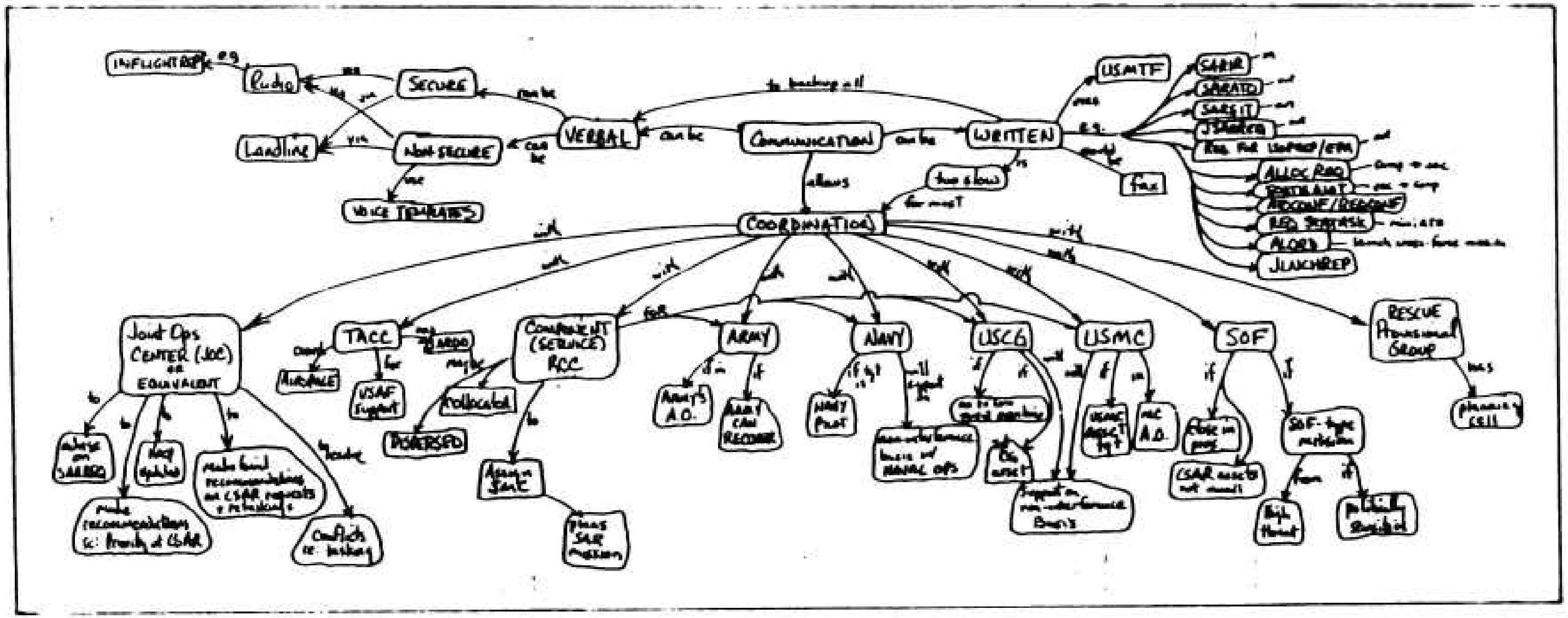


Figure D.5. Coordinate with Other Agencies

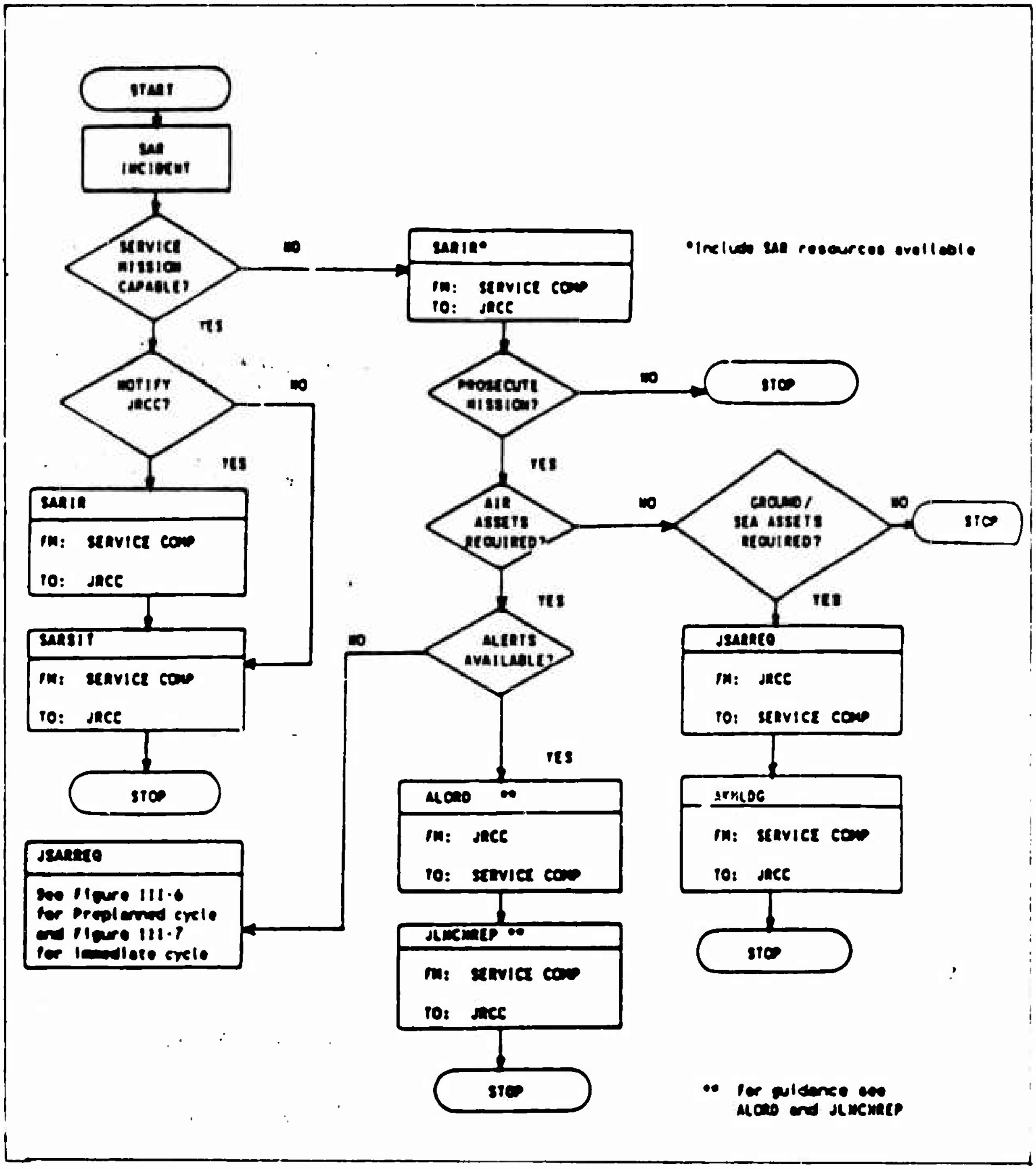


Figure D.6a. Tasking CSAR Resources - SAR Decision Tree (ALFA, 1989:2-15)

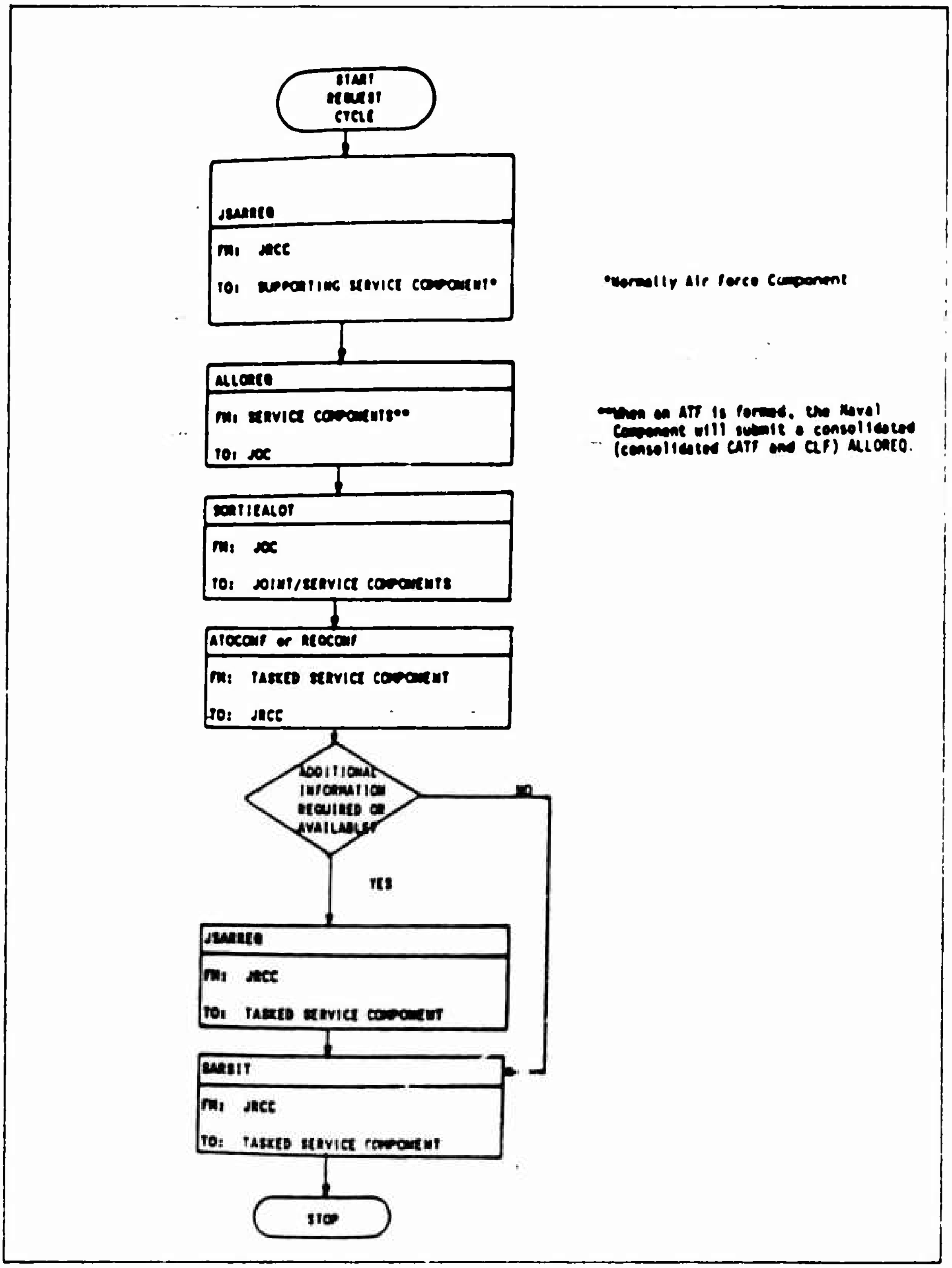


Figure D.6b. Tasking CSAR Resources - Preplanned SAR Request Cycle (ALFA, 1989:2-16)

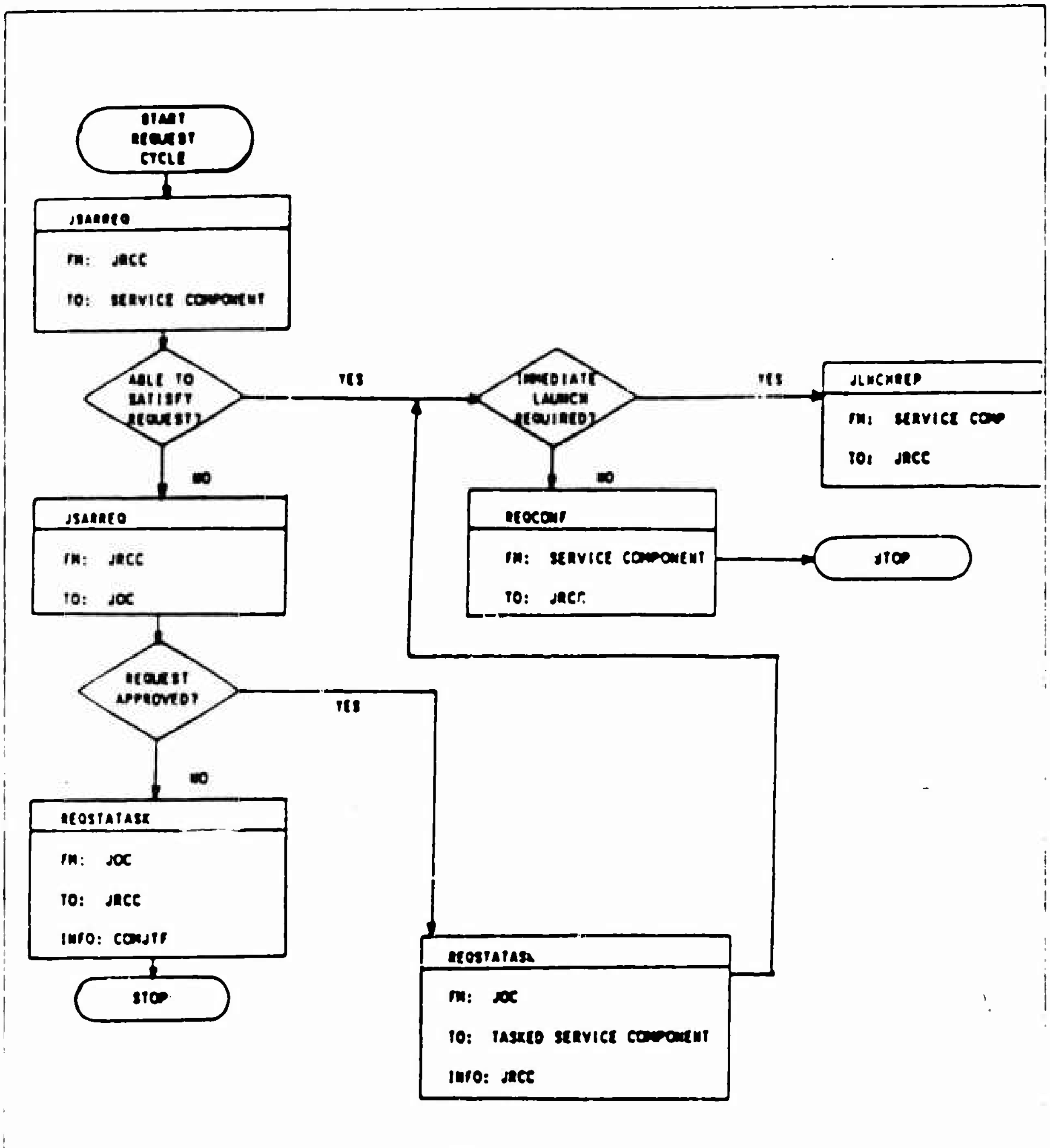


Figure D.6c. Tasking CSAR Resources - Immediate SAR Request Cycle
(ALFA, 1989:2-17)

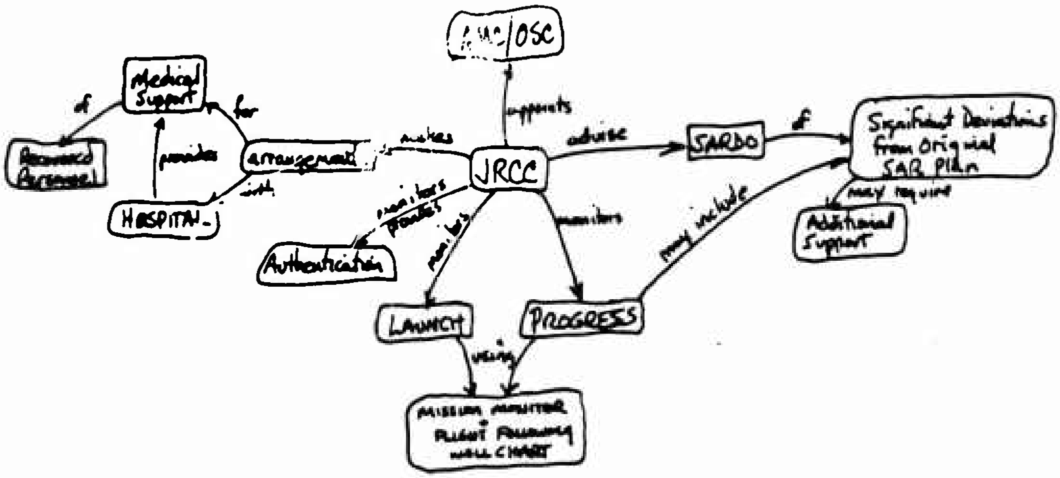


Figure D.7. Controlling/Monitoring the Mission

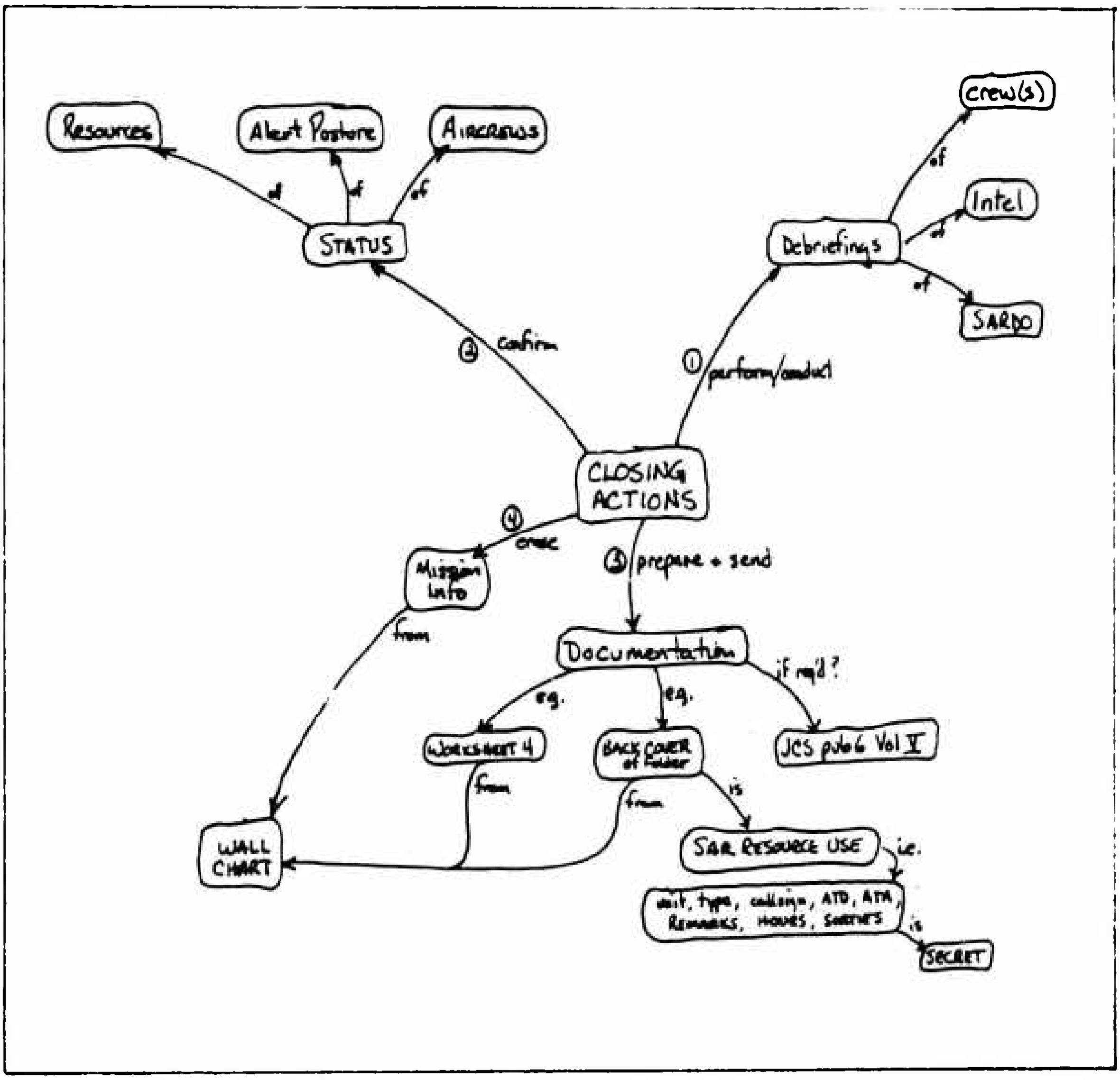


Figure D.8. Accomplishing Closing Actions

Appendix E: Task Analysis Candidate Kernels List

NOTE: Entries preceded by an asterisk (*) are addressed by the current iteration of the storyboard.

Gather Information

* Task: Calculate datum, drift, coverage, and search area.
  Current Method: National SAR Manual, USCG Manuals
  Proposed Aid/Methodology: Algorithm
  Source: same as above

* Task: Plot target location on map
  Current Method: on acetated wall chart
  Proposed Aid/Methodology: automatically placed on map display when location typed in input display
  Source: target database

* Task: Determine pressure altitude and temperature at the site.
  Current Method: overflights or, if not given by OSC, table lookups based on wx reports, and rules of thumb
  Proposed Aid/Methodology: Database lookup, possibly coupled with expert system for hueristics.
  Source: weather manuals, helicopter ops manuals (MACR 55-54)

* Task: Gather information on terrain, elevation, vegetation
  Current Method: overflights, map study
  Proposed Aid/Methodology: digitized map display with underlying database
  Source: DMA, CIA, DIA

Task: Gather info on weather--winds, PA, temp, visibility, ceiling, seas
  Current Method: overflights, weather reports via phone
  Proposed Aid/Methodology: data link with weather unit
  Source: weather unit

* Task: Gather info on LKP, ejection point, crash site
  Current Method: wingman, reporting agency, passer-by
  Proposed Aid/Methodology: enter in database, shown on map in any coordinate system
  Source: same as above

* Task: Determine threat level
  Current Method: use of threat classification aid (flow chart) based on intel briefs, OB plots, debriefs and various threat criteria
  Proposed Aid/Methodology: knowledge system
  Source: same as above

* Task: View and understand enroute and target area threat
  Current Method: intel briefs, OB plots, debriefs
  Proposed Aid/Methodology: dynamic link with intel OB system overlay on map display
  Source: same as above

Task: Know location and characteristics of DARs and SAFEs
  Current Method: unknown
  Proposed Aid/Methodology: database
  Source: DIA

Task: break down ATO
  Current Method: manual search of ATO for CSAR resources
  Proposed Aid/Methodology: "automated" ATO database
  Source: none

* Task: Validate target
  Current Method: visual or radio by someone in the target area
  Proposed Aid/Methodology: include in target database
  Source: same as above.

* Task: Gather ISOPREP and EPA information
  Current Method: call or send message to unit intel section
  Proposed Aid/Methodology: secure database
  Source: ISOPREP and EPA cards

* Task: Determine physical condition of target personnel
  Current Method: actual report or best guess
  Proposed Aid/Methodology: model based on conditions of distress and historical data (probably not very useful)
  Source: medical studies?

* Task: Decide if immediate response is feasible or recommended
  Current Method: rules based on resources available, threat level, and contact with the target
  Proposed Aid/Methodology: Expert System to make recommendations based on resources database and information input about threat and contact
  Source: interview experts

Task: Decide whether to start planning or send the mission to Prov Group for planning and coordination
  Current Method: rules based on info available, most likely mission type, and current workload
  Proposed Aid/Methodology: Expert System to make recommendations
  Source: interview experts

Task: determine estimated target position (if unknown)
Current Method: best guess based on rules of thumb and EPA
Proposed Aid/Methodology: Knowledge System based on rules, coupled with historical data from USCG SAR School and SERE, linked with EPA
Source: rules-experts; historical data-USCG SAR School and SERE; EPA database (see above)

* Task: Determine number of personnel to recover
Current Method: actual reports, flight plans, pax rosters, unit alpha rosters
Proposed Aid/Methodology: if unknown, database on vehicle (aircraft and ship) capabilities and payloads
Source: "Jane's" Books

Task: Determine who (JRCC or RCC) acts as SMC
Current Method: based on whether or not RCC can handle the mission
Proposed Aid/Methodology: none
Source: n/a

Prioritize Target

* Task: Assign target priority
Current Method: hueristics, CINC/JFC/Prov Group guidance
Proposed Aid/Methodology: interactive knowledge system
Source: interview experts, historical accounts of guidance

Task: Estimate probability of survival
Current Method: not done overtly; figured into priority
Proposed Aid/Methodology: model based on historical data (SERE)
Source: SERE schools, MAC pararescue study

Task: Estimate target value
Current Method: not done overtly; figured into priority
Proposed Aid/Methodology: MCDM model using AHP
Source: user's preferences based on ?

Task: Estimate Urgency
Current Method: hueristics based on weather, physical condition, and threat (probability of survival)
Proposed Aid/Methodology: MCDM model using AHP
Source: user's preferences based on above info

Evaluate Options

Task: Decide which resources are available
Current Method: ATO and resource status board
Proposed Aid/Methodology: automated ATO and resources database
Source: data on resources

Task: Decide which resources are appropriate
Current Method: satisficing based on priority and availability
Proposed Aid/Methodology: Expert System to recommend resources
Source: interview experts, data on resources

Task: Establish recovery time and location
Current Method: EPA or resource/target availability
Proposed Aid/Methodology: recommendation by expert system based on scan of nearest DARs and SAFEs, analysis of enemy OB in nearby area, analysis of terrain, and rules of movement
Source: interview experts, SERE manuals

* Task: Determine most appropriate CSAR method
Current Method: rules
Proposed Aid/Methodology: interactive knowledge system
Source: same as above

* Task: Determine most appropriate CSAR tactic
Current Method: rules
Proposed Aid/Methodology: interactive knowledge system
Source: same as above

Task: Evaluate alternatives
Current Method: probably rule-based (recognition-primed decision making) therefore not much evaluation of alternatives
Proposed Aid/Methodology: quick response simulation model
Source: n/a

Plan Recovery

Task: Determine friendly actions affecting recovery
Current Method: ATO, mission briefs, intel
Proposed Aid/Methodology: automated ATO, "newsclipping" service
Source: unknown

Task: Determine enemy actions affecting recovery
Current Method: ATO, mission briefs, intel
Proposed Aid/Methodology: automated ATO, "newsclipping" service
Source: unknown

Task: Decide which resource to task
Current Method: satisficing
Proposed Aid/Methodology: MCDM model to suggest optimum resource for a given target.
Source: data on resources

Task: Determine configuration of resources
Current Method: standard loads/configuration, or "take what you get"
Proposed Aid/Methodology: database
Source: WWMCCS ?

Coordinate Resources

Task: Request resources
Current Method: manually type and send message to appropriate agency
Proposed Aid/Methodology: automate using word processor with message templates
Source: USMTF

Task: Communicate with SARDO, CSAR units, SRCCs, host nation
Current Method: phone, message
Proposed Aid/Methodology: comm net including modem, fax
Source: LAN ?

Task CSAR Resources

Task: If immediate launch, execute scramble checklist
Current Method: manually run checklist
Proposed Aid/Methodology: once determined, automated checklist will execute messages, notify SARDO, and prompt user for actions
Source: scramble checklist

Task: Appoint On-Scene Commander (OSC) and Airborne Mission Commander (AMC)
Current Method: Who is in the area? manual search of ATO
Proposed Aid/Methodology: automated ATO + database on who is qualified
Source: ATO + unit info ?

Task: Decide to launch resources
Current Method: target and resources ready, success likely
Proposed Aid/Methodology: notification window pops up when certain conditions are met
Source: resource database, target database

Task: Send appropriate messages
Current Method: follow SAR Decision trees
Proposed Aid/Methodology: Knowledge System and word processor with templates
Source: SAR Decision trees and USMTF

Control/Monitor Mission

* Task: Assign frequencies
Current Method: manual search of SPINs, ATO, and log
Proposed Aid/Methodology: database of available/assigned freqs
Source: same as above

* Task: monitor mission launches/progress
  Current Method: Mission Monitor & Flight Following wall chart
  Proposed Aid/Methodology: automate
  Source: same as above

<u>Closing Actions</u>

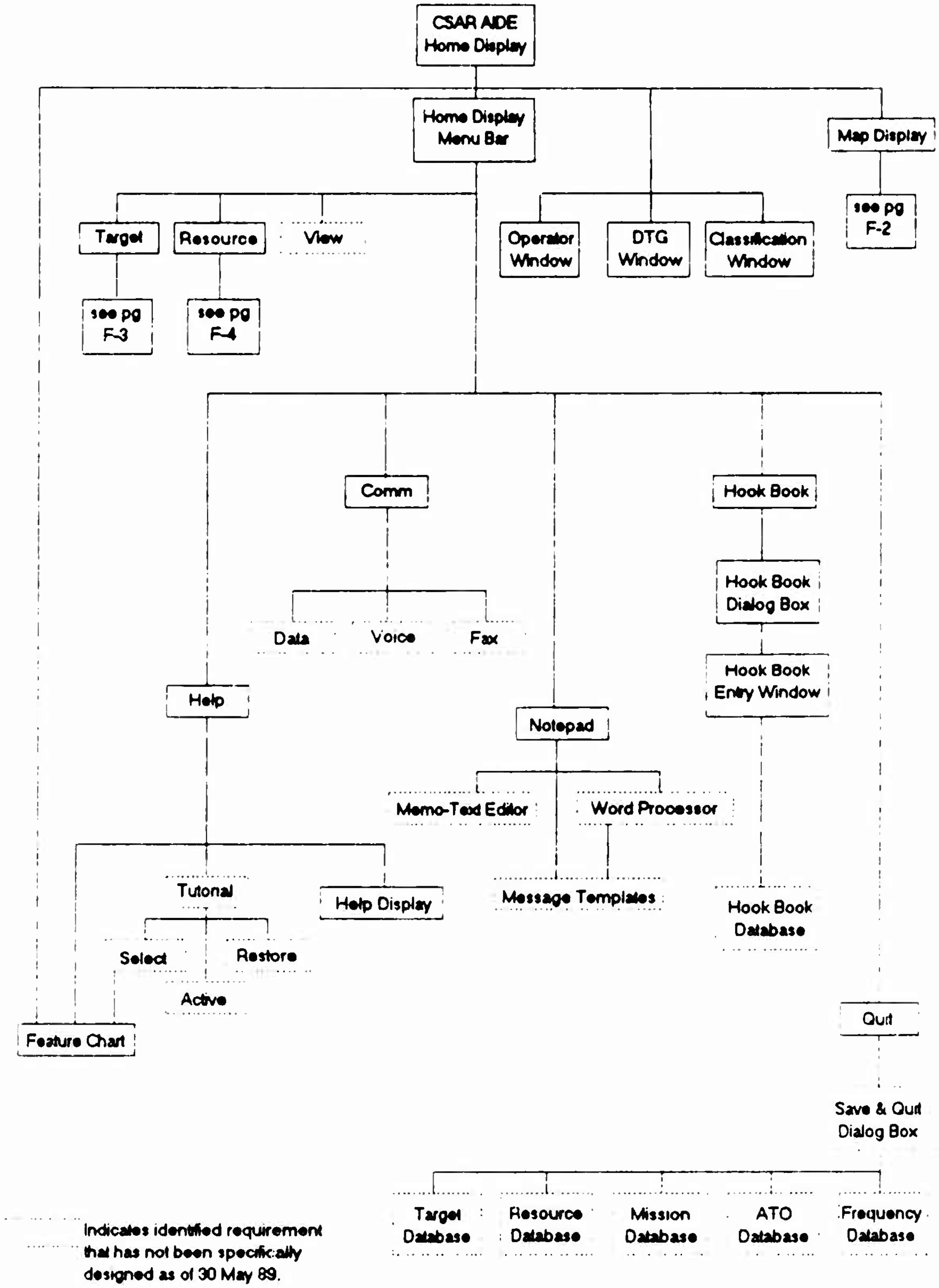
Task: Confirm status of resources, aircrews, and alert
  Current Method: phone call to/from unit
  Proposed Aid/Methodology: datalink with unit ops center
  Source: unit ops center

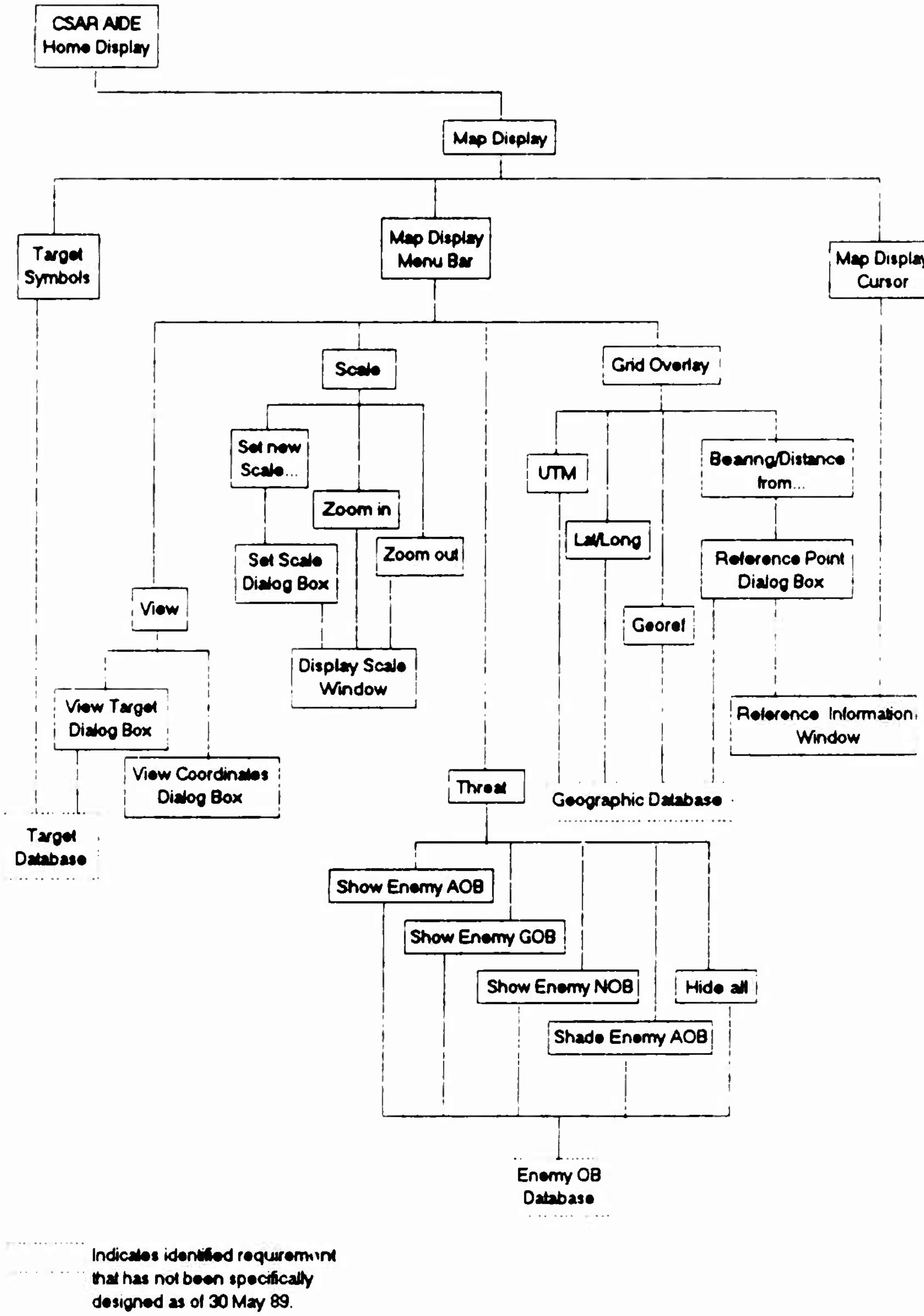
Task: debrief crews, intel, SARDO
  Current Method: verbal with forms to fill out
  Proposed Aid/Methodology: automated templates to enter data in appropriate database or send to appropriate agency.
  Source: checklists ?

Task: Prepare documentation
  Current Method: manually typed
  Proposed Aid/Methodology: word processor with formatted templates automatically filled in as much as possible by database, reviewed by user before automatically sent to proper agency
  Source: messages, documentation

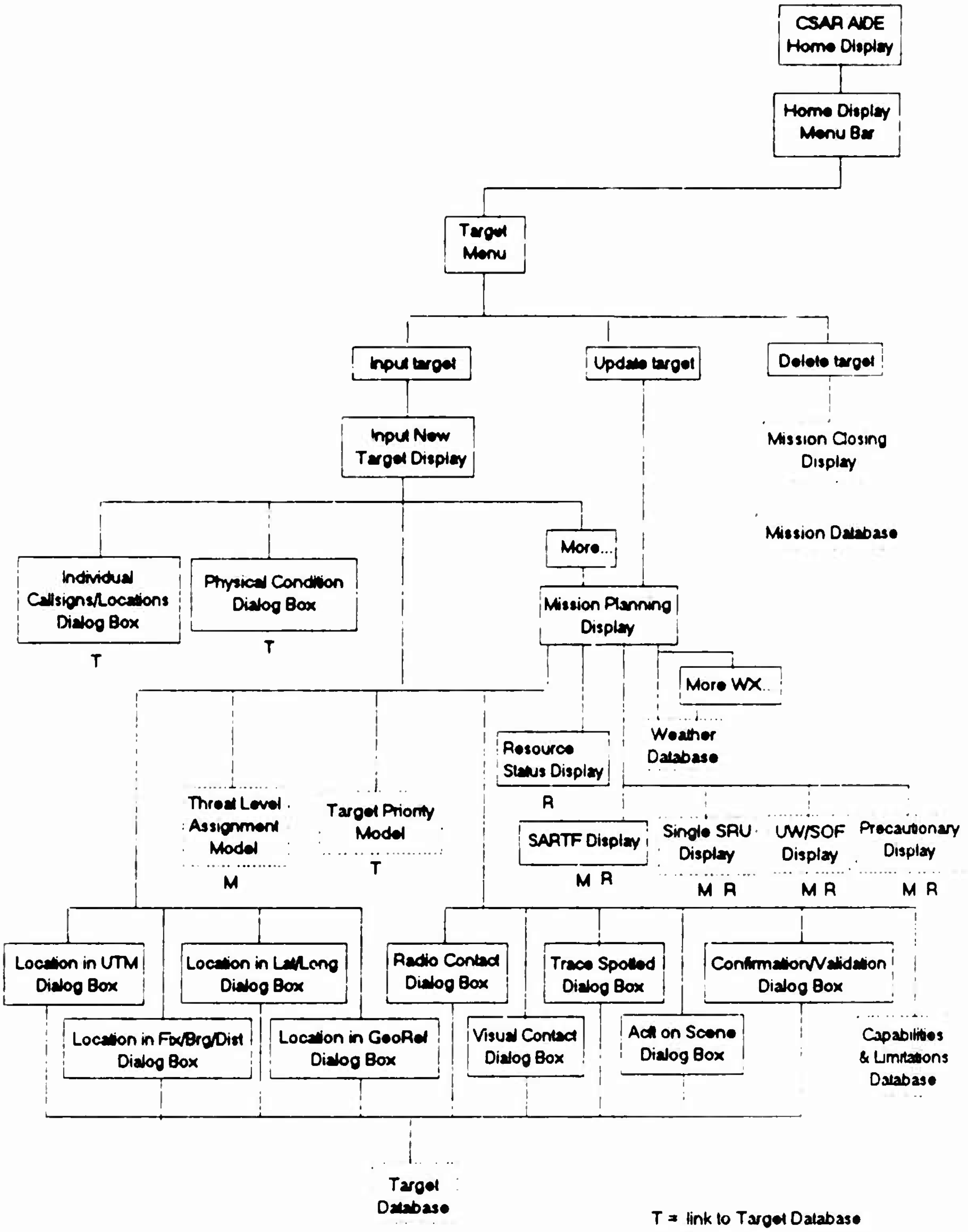
Task: Analysis of decision
  Current Method: not done
  Proposed Aid/Methodology: cost-effectiveness model
  Source: ?

Appendix F: CSAR AIDE Feature Chart



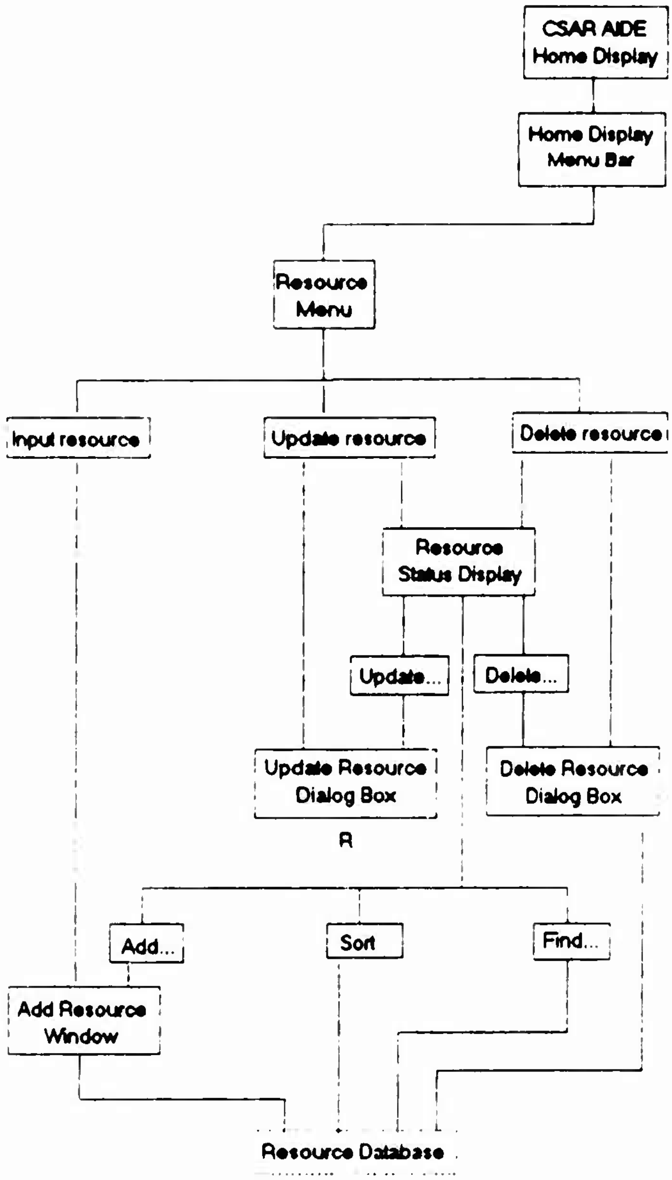


CSAR AIDE Home Display
Map Display
Target Symbols
Map Display Menu Bar
Map Display Cursor
Scale
Grid Overlay
Set new Scale...
Zoom in
Zoom out
Set Scale Dialog Box
Display Scale Window
View
View Target Dialog Box
View Coordinates Dialog Box
Target Database
UTM
Lat/Long
Georef
Bearing/Distance from...
Reference Point Dialog Box
Reference Information Window
Threat
Geographic Database
Show Enemy AOB
Show Enemy GOB
Show Enemy NOB
Hide all
Shade Enemy AOB
Enemy OB Database
Indicates identified requirement that has not been specifically designed as of 30 May 89.



T = link to Target Database
M= link to Mission Database
R = link to Resource Database

Indicates identified requirement that has not been specifically designed as of 30 May 89



R = link to Resource Database

Indicates identified requirement that has not been specifically designed as of 30 May 89.

Appendix G: CSAR AIDE Storyboard

Introduction

This appendix contains the current iteration of the design for CSAR AIDE. The storyboard contained in this appendix uses the Microsoft Windows Man-Machine Interface, therefore an explanation of Windows terms and control mechanisms are included. The storyboard format will then be explained, followed by the storyboard itself. It is important to realize that the designer does *not* have to design the storyboard frames using computer software. These frames were initially hand-sketched and were not converted to the enclosed format until the final thesis draft. The designer should use whatever method fits his creativity and meets the demands of the user.

The Windows Man-Machine Interface

The Microsoft Windows version 2.1 user interface was chosen for this storyboard because it offered a ready-made, easy-to-understand-and-use control mechanism. This section contains a streamlined glossary of terms and explanation of procedures taken from the Windows Users Guide (Microsoft Corp., 1988), Running Windows (Andrews and Stinson, 1986), and the Actor Language Manual (Whitewater Group, 1989).

Glossary of Windows Terms

Active. Describes the window or icon to which the next operation will apply.

Check box. A small square box that appears in the dialog box and that can be turned on or off. Check boxes generally represent multiple options the user can set.

Child window. Used to split the parent window's work area into smaller pieces, which can serve separate functions and be managed separately.

Click. To press and release the mouse button quickly.

Command button. A large rectangular button (this storyboard may use ovals) that appears in a dialog box and carries out or cancels a command.

Appendix G: CSAR AIDE Storyboard

Introduction

This appendix contains the current iteration of the design for CSAR AIDE. The storyboard contained in this appendix uses the Microsoft Windows Man-Machine Interface, therefore an explanation of Windows terms and control mechanisms are included. The storyboard format will then be explained, followed by the storyboard itself. It is important to realize that the designer does *not* have to design the storyboard frames using computer software. These frames were initially hand-sketched and were not converted to the enclosed format until the final thesis draft. The designer should use whatever method fits his creativity and meets the demands of the user.

The Windows Man-Machine Interface

The Microsoft Windows version 2.1 user interface was chosen for this storyboard because it offered a ready-made, easy-to-understand-and-use control mechanism. This section contains a streamlined glossary of terms and explanation of procedures taken from the Windows Users Guide (Microsoft Corp., 1988), Running Windows (Andrews and Stinson, 1986), and the Actor Language Manual (Whitewater Group, 1989).

Glossary of Windows Terms

Active. Describes the window or icon to which the next operation will apply.

Check box. A small square box that appears in the dialog box and that can be turned on or off. Check boxes generally represent multiple options the user can set.

Child window. Used to split the parent window's work area into smaller pieces, which can serve separate functions and be managed separately.

Click. To press and release the mouse button quickly.

Command button. A large rectangular button (this storyboard may use ovals) that appears in a dialog box and carries out or cancels a command.

Control menu. The menu that appears on every window. Icons and some dialog boxes also have a Control menu. It contains movement, sizing, and closing commands. It is accessed by clicking on the Control menu box.

Control menu box. The small box to the far left in the Title bar that allows access to the control menu. Double-clicking on the box will close the window.

Dialog box. A rectangular box that appears when the system needs additional information from the user.

Double-click. To rapidly press and release the mouse button twice without moving the mouse. This action carries out a command.

Drag. To press and hold down the mouse button while moving the mouse.

Grayed. Describes a command or option that is listed in a menu or dialog box but cannot be chosen or selected. The command or option appears in gray type.

Highlighted. Indicates that the object is selected and will be affected by the user's next action. A highlighted object will appear in reverse video.

Icon. A small symbol that represents an application that is running in memory. The user can enlarge an icon to a window when he wants to use that application.

Insertion point. The place where text will be inserted when the user types. The insertion point usually appears as a flashing vertical line in an application's window or dialog box. The typed text will appear to the left of the insertion point, which is pushed to the right as the user types.

List box. A box within a dialog box that lists all items that a command could affect.

Maximize box. The small box containing an up arrow that is located at the right of the title bar. Mouse users can click the Maximize box to enlarge the window to its maximum size.

Menu. A group listing of available commands. Menu names appear in the menu bar. Use a command from the menu by selecting the menu, then choosing the command.

Menu bar. The horizontal bar that lists the names of the window's menus. The menu bar appears below the window's title bar.

Minimize box. The small box containing a down arrow that is located at the right of the title bar. Mouse users can click the Minimize box to reduce the window to an icon.

Option button. A small round button that appears in a dialog box and selects an option when set. Within a group of related option buttons, the user may make only one selection.

Parent window. A window that exercises some special control over some other window or windows, each of which is referred to as its child window.

Point. To move the pointer on the screen until it rests on the item desired.

Pointer. A small symbol that appears if a mouse or other pointing device is installed. The pointer indicates which area of the screen will be affected when the user clicks the mouse button. The pointer is usually shaped like an arrow but changes shape during certain tasks.

Popup window A window that when called appears to lie on top of another window. Popups can be moved and sized but cannot be maximized or made into an icon.

Restore box. A small box containing up and down arrows that appears at the right of the title bar after a window has been maximized. Mouse users can click on the restore box to return the window to its previous size.

Scroll. To move text or graphics up or down, or left or right, to see the parts of the file that cannot be seen in the window.

Scroll bar. A bar that appears at the right side and/or bottom of some windows and in some dialog boxes. the scroll bar contains a scroll arrow at either end and a scroll box (or "thumb") that moves within the scroll bar to reflect the position within the file.

Select. To indicate the item that the next command will affect.

Shortcut key. A special key or key sequence available for some commands that the user can press to execute the command without first selecting a menu (also called "accelerator keys").

Text box. A box in a dialog box in which the user types information needed to carry out a command. The text box may be blank when the dialog box appears or it may contain text if there is a default option or if the user has selected something applicable to that command.

Title bar. The horizontal bar across the top of each window that contains the name of the window and/or file in that window. The title bar also contains the Control menu box and the Maximize and Minimize boxes or the Minimize and Restore boxes.

Window. A rectangular area on the screen which contains a particular application.

Work area. The area of a window that displays the information contained in the file (also called "client area").

Window Operations

Choosing Commands

To choose a command from a menu:

1) Click the menu you want to select.
2) Click the command you want to choose.

To choose a command from the Control menu:

1) Click the Control-menu box (upper left corner of window).
2) Click the command you want to choose.

Canceling Menus

To cancel a menu click an area outside the menu.

Using Dialog Boxes

A dialog box appears when the system needs information from the user to carry out a command. Dialog boxes contain different kinds of items, depending on what information is required:

- Type text in a text box.
- Select one item from a list box.
- Choose one or more boxes from a group of check boxes.
- Choose one button from a group of option buttons.
- Command buttons carry out commands.

To select an item from a dialog box, click the item.

Sizing Windows

To change a windows size:

1) Point to a window border--either an edge or a corner.
2) Drag the border to a new location.
3) Release the mouse button.

Moving Windows

To move a window:

1) Point to the window's title bar.
2) Drag the window to the new location.
3) Release the mouse button.

Moving Icons

To move an icon:

1) Point to the icon.
2) Drag the icon to a new location.
3) Release the mouse button.

Shrinking Windows to Icons

To shrink a window to an icon click the Minimize box (the "downward arrow") in the upper-right corner of the window.

Enlarging Windows to Maximum Size

To enlarge a window to its maximum size click the Maximize box (the "upward arrow") in the upper-right corner of the window.

Restoring Windows

To restore a window to its previous size, click the Restore box (the "up-and-down arrows") in the upper-right corner of the window.

Restoring Icons

To restore an icon to its previous window size, double-click the icon.

Storyboard Format

Each frame of the accompanying storyboard will follow the basic format shown in Figure 2.4, with the text description and ROMC checklist on the left and the graphic depiction of the actual frame on the facing page to the right. The documentation is included in the description and ROMC checklist and is, therefore, not contained in its own section. The reader is referred to Appendix F for the location of each frame in the feature chart. The text will conform to the following outline:

Description

Purpose. What is the reason this frame exists? What decision process or activity does it support?

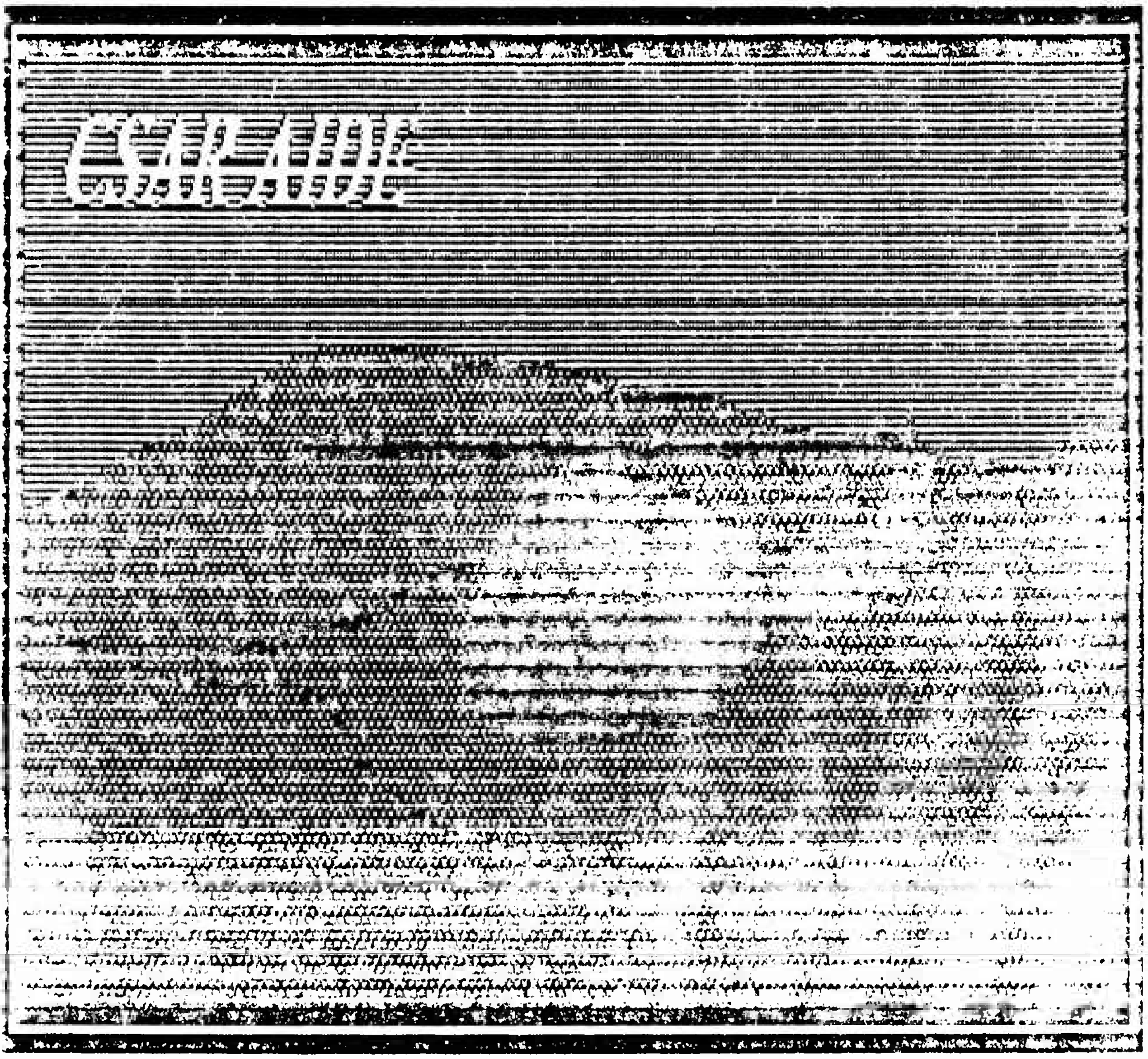
ROMC Checklist

Representations. This section will describe the purpose and use of specific representations depicted as part of the MMI (e.g., maps, charts, data entry forms, and reports).

Operations. Describes what operations the user can perform to support the intelligence, design, and choice phases of decision making.

Memory aids. Describes the supporting databases, files, links, triggers, profile defaults and other aids used in the frame.

Control mechanisms. The control design is taken directly from Microsoft Windows version 2.1. Each window has the same basic control mechanisms customized to meet the needs of that particular window. The basics of this control mechanism are covered on the next page, while in subsequent frames, only exceptions to the norm will be noted. It is assumed the user's computer is equipped with a mouse (if not, the system does have keyboard commands also, but they will not be covered here; see Windows User's Guide).



Combat Search And Rescue

Analysis, Integration, & Decision Environment

Home Display

Description

The Home Display is the main window of CSAR AIDE.

Purpose. Serves as the anchor for all other displays in the DSS. It can be placed (sized or iconned) into the background (taking the rest of the displays with it) to allow the user to use the Windows environment to access applications outside of CSAR AIDE. As it is always present (as long as the CSAR AIDE program is open), it offers controls the user may need during any phase of operations. From the Home Display, the user can get to any other display.

ROMC Checklist

Representations

- The client area of the window is occupied by several other windows. The **Map Display** and the **Feature Chart** are the default windows which open on system startup unless the DSS is customized by the user. These displays will be discussed separately.

Operations

- none

Memory aids

- Title Bar displays title of main program "CSAR AIDE".
- **Operator Window** displays the current operator.
- **Date-Time Group Window** displays the current date and time based on the system clock in DTG format.
- **Classification Window** displays the highest classification of any information appearing in any window that is currently open, whether it is displayed, hidden, or iconned.

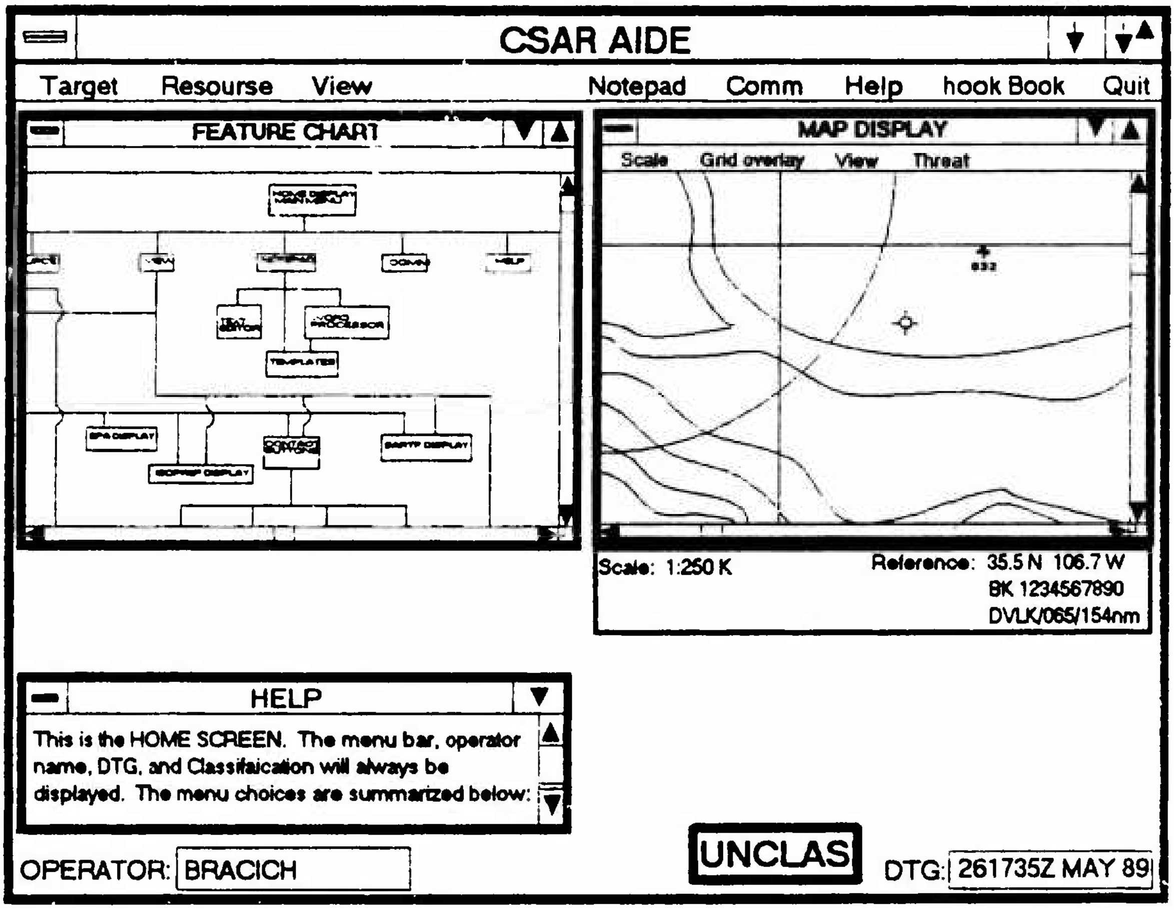
Control mechanisms

- Menu Bar contains the following menus:
  - Target
    - Input Target - opens **Input New Target Display** (see pg G-20).
    - Update Target - opens **Mission Planning Display** (see pg G-26).
    - Delete Target - opens **Mission Closing Display** (not included in this design iteration).
  - Resource
    - Input Resource - opens **Add Resource Window** (see pg G-30).
    - Update Resource - opens **Resource Status Display** and **Update Resource Dialog Box** (see pg G-30).
    - Delete Resource - opens **Resource Status Display** and **Delete Resource Dialog Box** (see pg G-30).

- View
    - **Mission Planning Display** (see pg G-26)
    - **SARTF Display** (see pg G-28)
    - **Resource Status Display** (see pg G-30)
    - **Map Display** (see pg G-16)
    - **EPA Display** (not included in this iteration)
    - **ISOPREP Display** (not included in this iteration)
    - **Feature Chart** (see pg G-14)
- Notepad
    - Message - opens word processor with formatted templates
    - Memo - text editor
    - Word Processor - no templates loaded, but they are still available
- Comm
    - Voice
    - Data
    - Fax
- Help
    - **Help Display** - opens the contextual Help Display window (for use in case it has been closed).
    - **Feature Chart** - opens the **Feature Chart** window which serves as a navigational tool and memory aid (see pg G-14).
    - **Tutorial** - gives user the choice of going through a tutorial on the currently active window or starting a training session where the user left off last time. He may also choose, using a feature chart, a particular area of the system.
- Hook Book - opens **Hook Book Dialog Box** and then the **Hook Book Entry Window** (see pg G-12).
- Quit - opens dialog box asking user if he wishes to save any files that have not been saved been since the last change. Then exits CSAR AIDE.

- **Help Display** is a context sensitive scrollable text screen which reflects information used to assist the non-expert user. The text in the **Help Display** will reflect information on the use and purpose of the currently active window.

Home Display



<u>Hook Book</u>

<u>Description</u>

<u>Purpose</u>. Allows the user to capture his thoughts on desired improvements to the system as he works.

<u>ROMC Checklist</u>

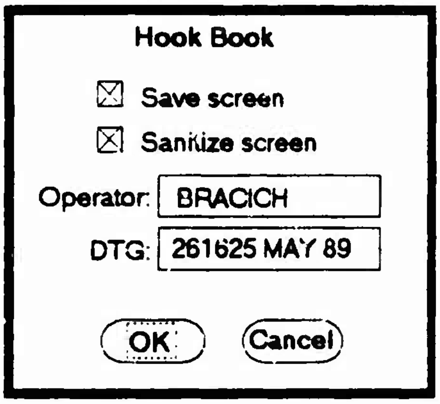
<u>Representations</u>
- **Hook Book Dialog Box**
- **Hook Book Entry Window**

<u>Operations</u>
- none

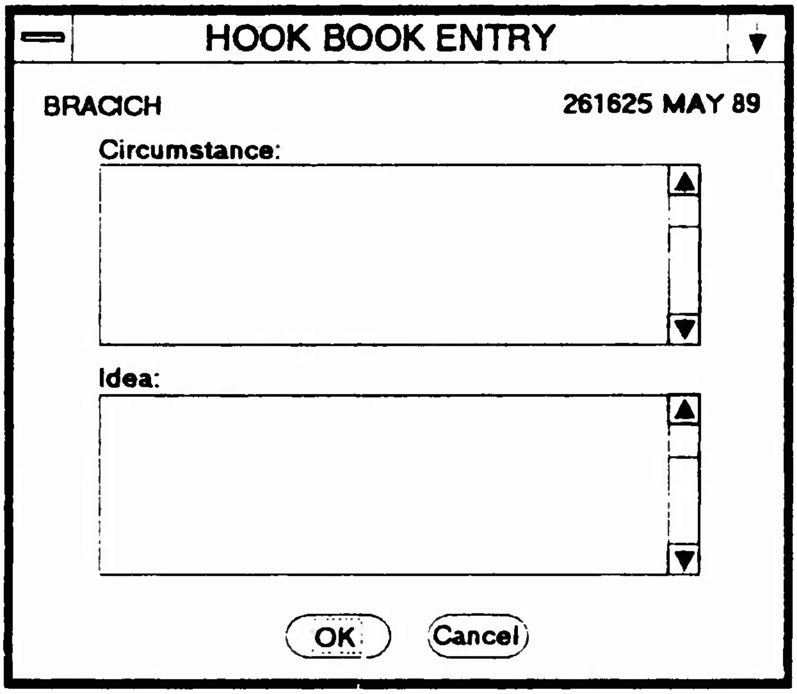
<u>Memory aids</u>
- Title bar displays "Hook Book"
- Current DTG and Operator are entered as default values in the **Hook Book Dialog Box.**
- "Save screen" is default selected.
- "Sanitize screen" is default selected.

<u>Control mechanisms</u>
- "Save screen" takes a snapshot of the current DSS screen for reference purposes. If this is not relevant to the idea to be input, the user may de-select the box.
- "Sanitize screen" will ensure all classified entries are not saved with the screen display. In the event the classified data is needed the box can be de-selected.
- Clicking on "OK" in the **Hook Book Dialog Box** executes the selected actions and opens the **Hook Book Entry Window.**
- Clicking on "OK" in the **Hook Book Entry Window** enters the information in the **Hook Book Database.**

Hook Book Dialog Box



Hook Book Entry Window



Feature Chart

Description

Purpose. Graphic display of DSS feature chart allows user to:
1) navigate the DSS directly, or;
2) maintain sense of where he is and what features are accessable from his current location.

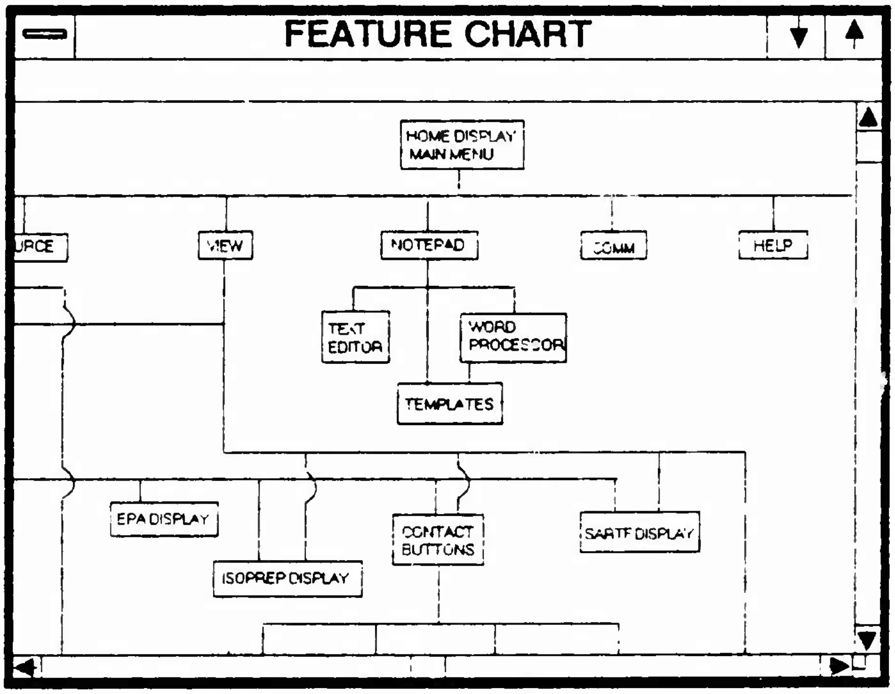
ROMC Checklist

Representations
- Feature Chart

Operations
- Allows user to navigate the DSS by clicking on the desired display or model

Memory aids
- Title bar displays "Feature Chart"
- Each block of the feature chart corresponds to a display, model, database, or menu choice. Allowable paths are linked.

Control mechanisms
- Menu bar has not been designed.



FEATURE CHART
HOME DISPLAY
MAIN MENU
URCE
VIEW
NOTEPAD
COMM
HELP
TEXT
EDITOR
WORD
PROCESSOR
TEMPLATES
EPA DISPLAY
ISOPREP DISPLAY
CONTACT
BUTTONS
SARTF DISPLAY

Map Display

Description

Digitized map showing terrain, targets, threat, and grid overlay in a given scale.

Purpose. Provide visual representation of target situation.

ROMC Checklist

Representations

- **Map Display** work area is a digitized terrain map.
- **Set Scale Dialog Box** (see pg G-18)
- **Reference Point Dialog Box** (see pg G-18)
- **View Target Dialog Box** (see pg G-18)
- **View Coordinates Dialog Box** (see pg G-18)

Operations

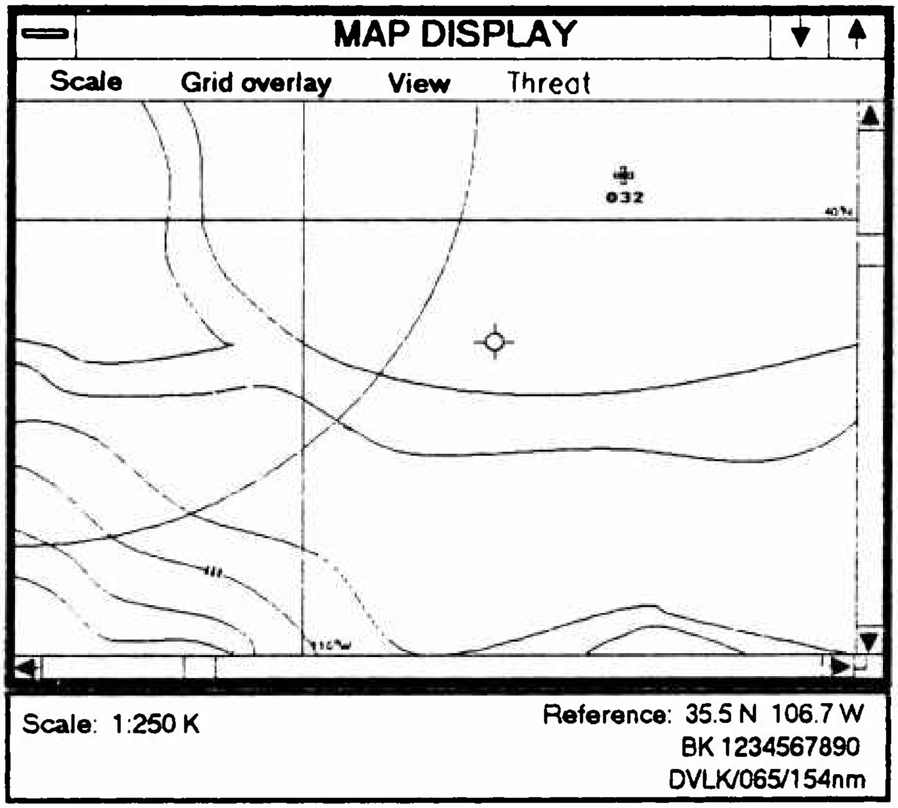
- none

Memory aids

- Title bar displays "Map Display"
- targets are placed based on location input to target database
- **Target Symbol** changes color based on mission status
- mission numbers appear below target symbols
- **Display Scale Window** - shows current map scale
- **Reference Information Window** - based on **Reference Cursor** location and **Reference Point Dialog Box.**

Control mechanisms

- See pg G-17 for Menu Bar items.
- **Reference Cursor** - feeds **Reference Information Window** below **Map Display.** Slews to location selected in either of the **View... Dialog Boxes.**
  - move by pointing and dragging to new location. Map will scroll if cursor is dragged off edge of map.
- **Target Symbol** - shows target location and mission number (if assigned). Color will show mission status.
  - double-clicking on **Target Symbol** will open **Mission Planning Display** for that target.



032 Target location and Mission number

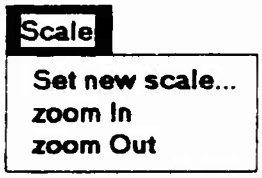
Reference Cursor: to move, "point and drag"
- feeds reference info below map

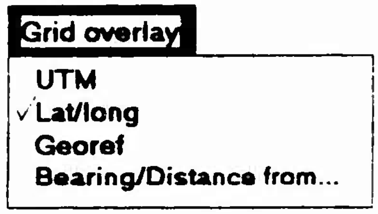
Map Display Menu Choices

Control mechanisms

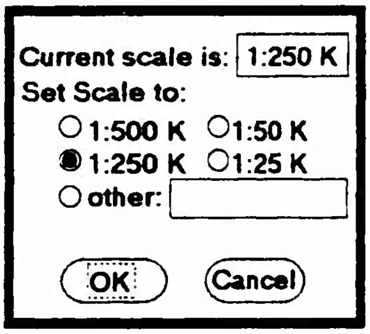
- Menu bar contains the following menus:
  - Scale
    - Set new scale - opens **Set Scale Dialog Box**
    - Zoom in - after selecting an area by pointing to the corner of the desired area, dragging the mouse to the opposite corner of the area, and releasing the mouse button (resulting in a box outlining the selected area), this option will change the scale to fill the display's work area with the selected map area.
    - Zoom out - changes scale to show 5 times current area in the work area (this default multiplication factor can be changed by the user).
  - Grid Overlay - places the selected grid(s) over the work area. If the "Bearing/Distance from..." option is selected, the **Reference Point Dialog Box** opens. A check appears in front of each selected grid. To remove a grid, click it again.
  - View
    - Target - opens **View Target Dialog Box**
    - Coordinates - opens **View Coordinates Dialog Box**
  - Threat
    - Show... - select the desired Order(s) of Battle to display.
    - Shade Enemy AOB - allows planner to see "threat rings" without changing scale or scrolling map.
    - Hide all - erases all threat information. Friendly OB is always shown.

Map Display Menu Choices

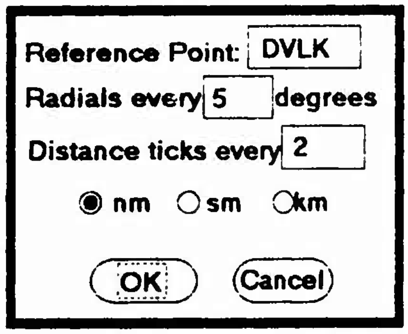


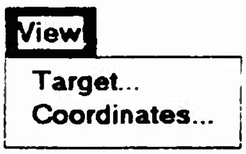


Set Scale Dialog Box

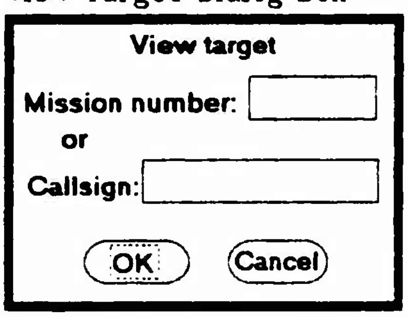


Reference Point Dialog Box

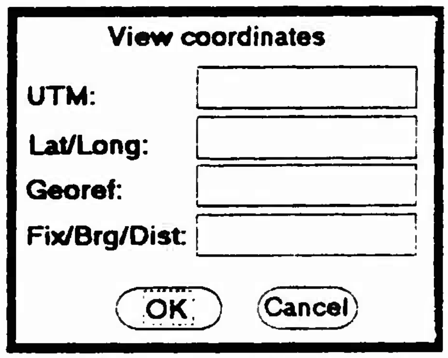


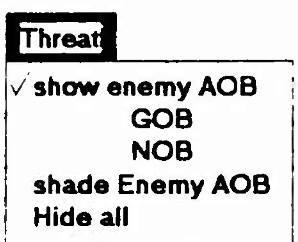


View Target Dialog Box



View Coordinates Dialog Box





Input New Target

Description

Purpose. Enables user to input data on a new CSAR target into the target database.

ROMC Checklist

Representations

- **Input New Target Display** is the data entry form.
- See pp G-21 thru G-24 for other representations accessible from this display.

Operations

- Gather data on target (identification of objective).
- Information input here will affect the target database which feeds other parts of CSAR AIDE.
- Threat level allows user to assign risk. User can set threat level explicitly or, by typing "help" in the Priority block, the user can access the **Threat Level Assignment Model** for assistance.
- Target priority - User can set target priority explicitly or, by typing "help" in the Priority block, the user can access the **Target Priority Model** for assistance.
- The "Request:" check boxes will automatically send the proper request to the unit intel section input above as soon as the user clicks "OK" or "MORE...".

Memory aids

- all information is input into the **Target Database**
- typing "help" in the "# to Recover" block will access the appropriate **Capabilities/Limitations Database** based on the input in the "Type Acft/Unit" block.

Control mechanisms

- user can minimize (icon) this window in case he needs to work another mission.
- if sized smaller that maximum, appropriate scroll bars will appear.
- If # to recover or #indiv callsigns/locations is greater than 1, then **Individual Callsigns/Locations Dialog Box** opens (see pg 19).
- Location and Contact check boxes will open corresponding dialog box when clicked. If dialog box is cancelled then the check box is "de-selected" (see pg 24).
- If Acft on Scene "Yes" button is selected, the **Acft on Scene Dialog Box** opens (see pg 24).
- If the confirmation of Event "Yes" button is selected the **Confirmation/Validation Dialog Box** opens (see pg 24).
- "MORE..." opens the **Mission Planning Display** (see pg 26).
- "OK" inputs all data entered into the Target Database.

INPUT NEW TARGET

Callsign: [ ]

Type Acft/Unit: [ ] Type Msn: [ ]

Home Station: [ ] Unit: [ ]

to Recover: [ ] # Indiv Callsigns/Locations: [ ]

Physical Condition: ○ Known ○ Unknown

Location: ☐ UTM... ☐ GEOREF... ☐ Lat/Long... ☐ Fix/Brg/Dist...

Contact: ☐ Radio... ☐ Trace... ☐ Visual... ☐ None

Aircraft on scene: ○ Yes... ○ No

Event cause: [ ] DTG: [ ]

Reported Threat Level: ○ Unknown ○ None ○ Low ○ Medium ○ High

Type of Threat: [ ]

Confirmation/Verification: ○ Yes... ○ No

Request: ☒ ISOPREP ☒ EPA

Mission #: [ ] Priority: [ ]

Source of Information: [ ]

(OK) (More...) (Cancel)

Individual, Condition, and Location Dialog Boxes

Description

Subordinate dialog boxes of the **Input New Target** window. They are, however, accessible from other parts of the DSS.

Purpose. Allows input of information concerning individual personnel (callsign, location, condition) and various types of location coordinate inputs.

ROMC Checklist

Representations

- **Individual Callsign/Location Dialog Box**
- **Physical Condition Dialog Box**
- **Location in UTM Dialog Box**
- **Location in Lat/Long Dialog Box**
- **Location in GeoRef Dialog Box**
- **Location in Fix/Brg/Dist Dialog Box**

Operations

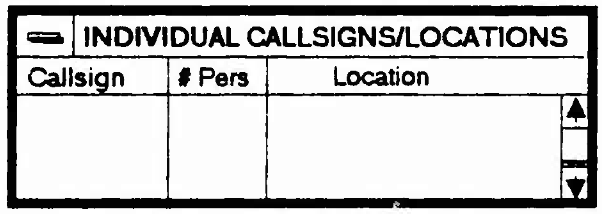
- **Location... Dialog Boxes** will search the map and assign values to the **Land** or **Water** selection boxes unless input by the user. If a conflict exists between user input and results of the map search, an error window will ask the user for confirmation and the map will slew to the input location.

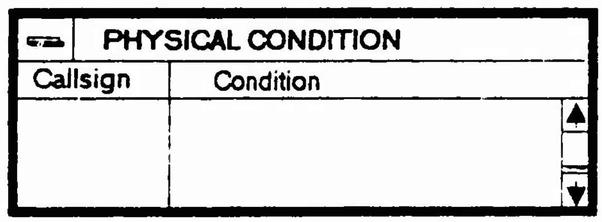
Memory aids

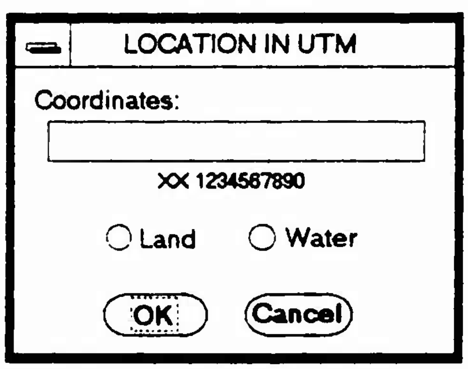
- default values for the **Individual Callsigns/Locations Dialog Box** are taken from values entered in the **Input New Target Display**. Callsigns are based on the number of indiv. callsigns entered and standard lettering sequence (Jolly 69A, 69B, 69C, etc.).
- default values for callsign in the **Physical Condition Dialog Box** will come from either the callsign block in the **Input New Target Display** or the **Individual Callsigns/Locations Dialog Box.**

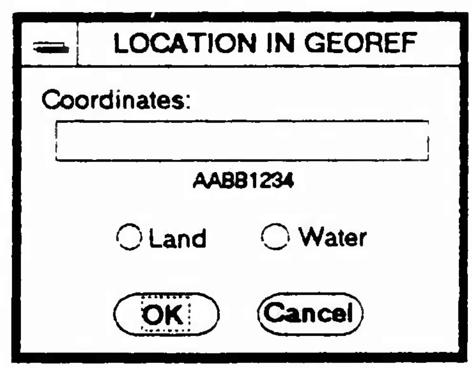
Control mechanisms

- "OK" enters information in the appropriate database.
- "Cancel" closes the dialog box without making any changes to the database.

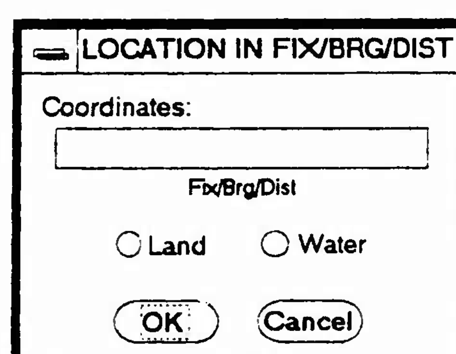

INDIVIDUAL CALLSIGNS/LOCATIONS
Callsign
Pers
Location


PHYSICAL CONDITION
Callsign
Condition


LOCATION IN UTM
Coordinates:
XX 1234567890
Land
Water
OK
Cancel


LOCATION IN GEOREF
Coordinates:
AABB1234
Land
Water
OK
Cancel


LOCATION IN LAT/LONG
Coordinates:
dd/mm/ss N-S; ddd/mm/sss E-W
or
ddd.dd N-S; ddd.dd E-W
Land
Water
OK
Cancel


LOCATION IN FIX/BRG/DIST
Coordinates:
Fix/Brg/Dist
Land
Water
OK
Cancel

Contact, On-Scene, and Confirmation Dialog Boxes

Description

Subordinate dialog boxes of the **Input New Target** window. They are, however, accessible from other parts of the DSS.

Purpose. Allows input of information concerning various types of contacts, aircraft on scene, and confirmation of the event.

ROMC Checklist

Representations

- **Radio Contact Dialog Box**
- **Visual Contact Dialog Box**
- **Trace Spotted Dialog Box**
- **Acft on Scene Dialog Box**
- **Confirmation/Validation Dialog Box**

Operations

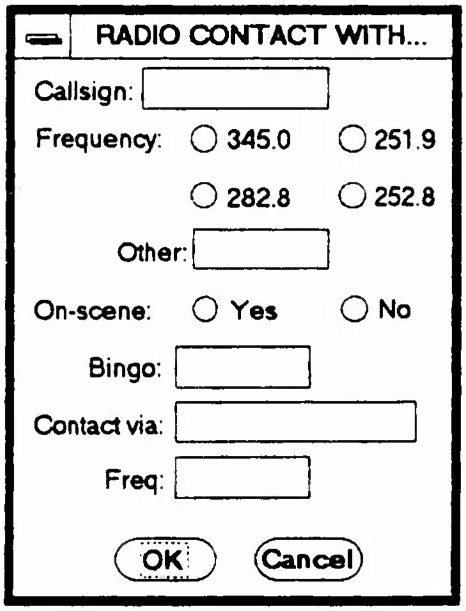
- none

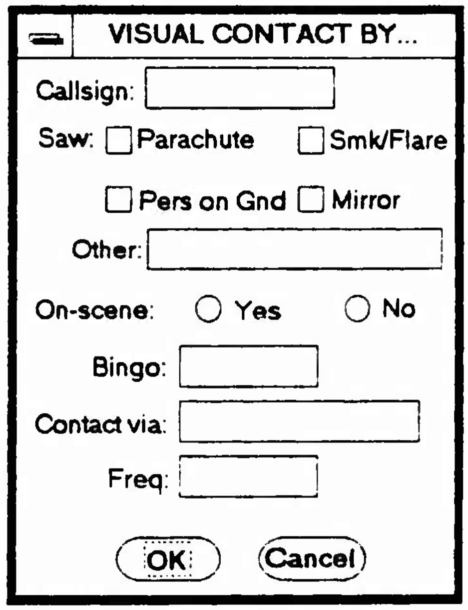
Memory aids

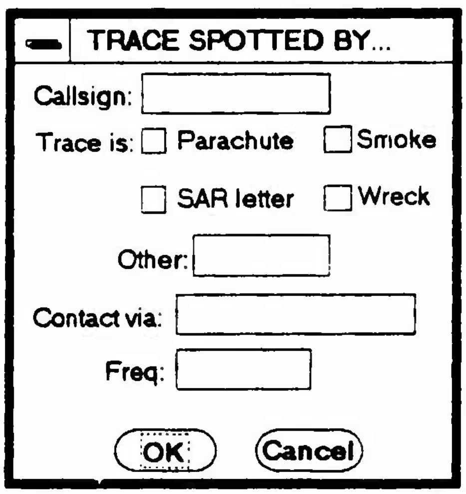
- default values for "callsign", "contact via...", and "Freq" in the **Contact Dialog Boxes** are based on any values that have already been input in corresponding blocks in other dialog boxes.

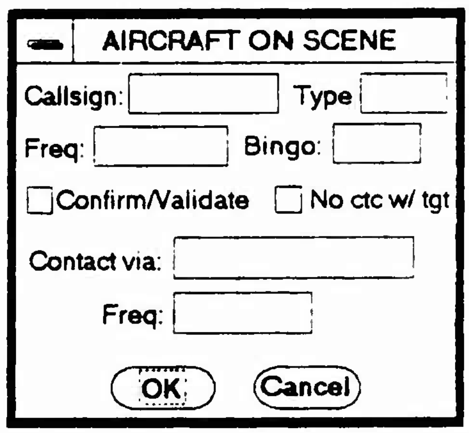
Control mechanisms

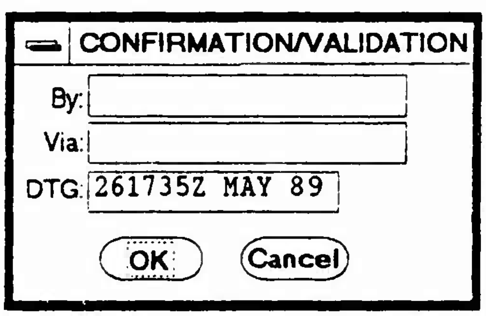
- "OK" enters information in the appropriate database.
- "Cancel" closes the dialog box without making any changes to the database.


RADIO CONTACT WITH...
Callsign:
Frequency: 345.0 251.9
282.8 252.8
Other:
On-scene: Yes No
Bingo:
Contact via:
Freq:
OK Cancel


VISUAL CONTACT BY...
Callsign:
Saw: Parachute Smk/Flare
Pers on Gnd Mirror
Other:
On-scene: Yes No
Bingo:
Contact via:
Freq:
OK Cancel


TRACE SPOTTED BY...
Callsign:
Trace is: Parachute Smoke
SAR letter Wreck
Other:
Contact via:
Freq:
OK Cancel


AIRCRAFT ON SCENE
Callsign: Type
Freq: Bingo:
Confirm/Validate No ctc w/ tgt
Contact via:
Freq:
OK Cancel


CONFIRMATION/VALIDATION
By:
Via:
DTG: 261735Z MAY 89
OK Cancel

Mission Planning Display

Description

Purpose. Used to continue entering or to edit Target and Mission Planning data. Provides the user with access to all information required to plan a CSAR mission.

ROMC Checklist

Representations

- Data entry/edit form
- Callsign/Location/condition window is scrollable text window that is a child window of the **Mission Planning Display.**

Operations

- Threat level block - either edit known threat level or type "help" to open **Threat Level Assignment Model).**
- Typing "help" in "Press Alt" block will access **Pressure Altitude Lookup Table** based on temperature and altimeter.
- Selecting a Method or Tactic Selection Button could open a window explaining the suggested method/tactic for a given situation or make a recommendation using an expert or knowledge system based on information input thus far.

Memory aids

- "OK" inputs information into **Target/Mission Databases.**
- Classified information is preceded by the proper classification, if known. If it is not known, but the information is likely classified, a classification block is provided. If information is input into the data block and the classification has not been entered, the cursor will return to the classification block (see Hook Book 5/7/89 17:21).
- defaults for survival gear reflect the type of information the user should input if box is selected.

Control mechanisms

- "Method" or "Tactics" selection buttons open appropriate display when selected.
- Contacts "YES" selection boxes are marked when contacts have been entered. If the user wishes to enter a contact, selecting the appropriate "YES" box will open the corresponding dialog box. If a contact has been entered, but the dialog box is not visible, clicking the appropriate "VIEW" selection box will open the corresponding dialog box for review/editing.
- "More WX..." opens a text window which provides more detailed weather information (DD Form 175-1).
- "OK" and "Cancel" perform their standard functions.

MISSION PLANNING DISPLAY

Mission #: ____ Priority: ____ Confirmed/Validated ◯ Yes ◯ No

Callsign: ____ Type acft/unit: ____ Type Mission: ____

Home station: ____ Unit: ____ Source of Info: ____

TARGET: Total #: ____ Event DTG: ____ Event cause: ____

Callsign	#	Location	Phys Cond	Auth #	Question	Answer

Survival Gear: ☐ Std. ☐ 'Chute [Color] ☐ Beeper [Freq] ☐ Mirror ☐ Raft [Color] ☐ Seadye [Color] ☐ Flares [Color] ☐ Smoke [Color] ☐ Strobe [Color]

(☐) Other equipment: ____
Encoded: ____

Terrain: Elevation: ____ Vegetation: ____

(S) Cities: ____ (☐) Key terrain features: ____
Encoded: ____ Encoded: ____

(☐) Landmarks: ____ (☐) Other: ____
Encoded: ____ Encoded: ____

Intel: Encoded:

(S) Enroute: Threat level: ____
SAM (Type/Loc): ____
AAA (Type/Loc): ____
Other: ____

(S) Target Site: Threat level: ____
SAM (Type/Loc): ____
AAA (Type/Loc): ____
Other: ____

Weather:

	Winds	Ceiling	Visibility	Altimeter	Temp	Press Alt	Water Temp	Sea State
Enroute:								
Target site:								

☐ More WX...

Method: ◯ SARTF ◯ Single SRU ◯ UW/SOF ◯ Precautionary ◯ HELP

Tactic: ◯ Overt ◯ Clandestine ◯ Covert ◯ HELP

Contacts: (YES / VIEW) ☐ ☐ Radio; ☐ ☐ Visual; ☐ ☐ Trace; ☐ ☐ OSC - no ctc

OK

Cancel

UNCLAS

SARTF Display

Description

Purpose. To assist the user in planning and monitoring a SARTF mission.

ROMC Checklist

Representations

- Data input/edit form for mission info
- Scrollable data input/edit child windows for "player" info (one separate one for "strike" players due to different information block headings
- Scrollable data input/edit child windows for "SAR Frequencies"

Operations

Memory aids

- data already input in other displays will automatically appear in proper place here.
- players are listed in the "player" child window include: FAC, CAS, RESCORT, MIGCAP, HELO, TANKER, AMC, SEARCH (see Hook Book 5/7/89 18:50)
- SAR Frequency Nets are listed in "net" column of "SAR Frequencies" child window.
- "Base +/-" block filled automatically based on "Target Pickup DTG" block and codewords input into Codeword Special Instructions Display.
- Classifications are shown in the applicable places. An overall classification for the display is also provided.

Control mechanisms

- "OK" inputs the data into the Mission Planning Database.
- "Cancel" closes the dialog box without making any changes to the database.

SARTF DISPLAY

Mission #: ______ Callsign: ______ Priority: ______

(S) Target pickup DTG: ______ Deconflicted ○ Yes ○ No

Base +/- : ______ Type Mission: ______

Player	ATO Msn #	Callsign	(C) Type	#	Home Station	Unit	(S) Orbit Location	(S) Orbit Start	RTB	(S) Rendezvous Location	(S) Rendezvous Time
FAC											
CAS											
RESCORT											
MIGCAP											
AMC											
TANKER											
HELO											
SEARCH											

							(S) Target	(S) Location	(S) T-ON	(S) T-OFF	(C) Ord
STRIKE											

(C) SAR Frequencies:

NET	HF	UHF	VHF	FM
PRI RCC				
SEC RCC				
AMC				
OSC				
HELO				
Other				

UNCLAS

OK Cancel

Resource Status Display

Description

Purpose. Allow the user to input, update, and review CSAR resource status.

ROMC Checklist

Representations

- **Resource Status Display** is a scrollable edit window with information about CSAR resources needed to plan a CSAR mission.
- **Update Resource Dialog Box** allows the user to input the unit and ID of a resource to edit its status. The cursor will be placed in the MX Status block of the selected resource.
- **Find Dialog Box** allows the user to input any text string to search for in the **Resource Database**.
- **Add Resource Dialog Box** will input new resource data into the **Resource Database** when "OK" is clicked.
- **Delete Resource Dialog Box** allows user to delete a given resource from the **Resource Database**.

Operations

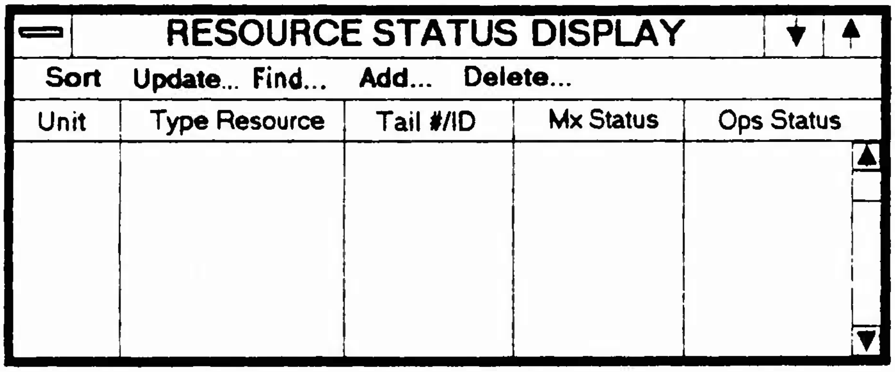
- None

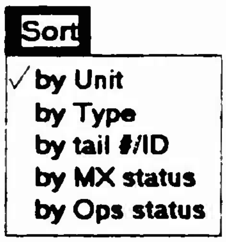
Memory aids

- Title Bar reflects **Resource Status Display**.
- default value for **Find Resource Dialog Box** is the last "find" input used this session.
- default value for the **Delete Resource Dialog Box** is the currently selected (highlighted) resource in the main window.

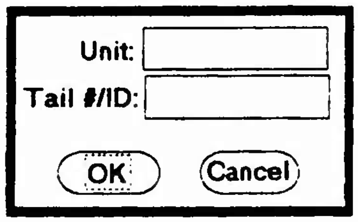
Control mechanisms

- Menu Bar contains the following commands:
  - Sort - allows user to select sort method for the display. Default is by unit.
  - Update - opens **Update Resource Dialog Box**
  - Find - opens **Find Resource Dialog Box**
  - Add - opens **Add Resource Dialog Box**3<
  - Delete opens **Delete Resource Dialog Box**

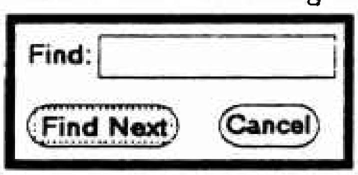


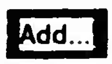


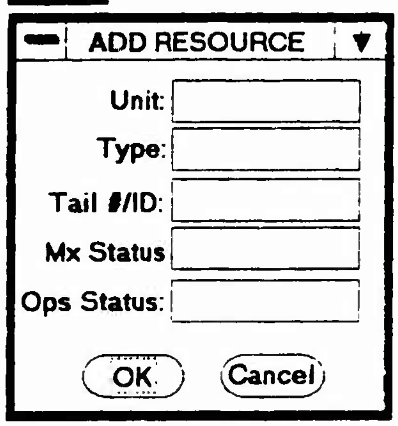
Update Resource Dialog Box



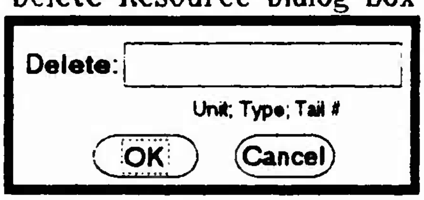
Find Resource Dialog Box







Delete Resource Dialog Box



Appendix H: CSAR AIDE Hook Book

There were, of course, many more entries to the hook book than appear below. Those included in the design have been deleted.

8/15/1988 6:45
Circumstances: working on storyboard explaining Hook Book
Idea: Windows cardfile allows user to save graphics and text on same card (2 layers. Can I use some type of snapshot utility to put a picture of the current screen on the graphics layer?

11/19/1988 12:03
Circ: Reading Andriole's article on storyboarding
Idea: find out about Army's Target Value Analysis (TVA) Model for establishing target priority.

5/06/1989 3:06
Circ: Working on Home Display frame
Idea: what should other opening screen window be? Perhaps the graphic feature chart?

5/06/1989 20:21
Consider using this feature (the notepad .LOG file as the hookbook instead of using cardfile. Advantage = automatic date stamp; disadvantage = no graphics, 8-page (?) limit.

5/07/1989 4:53
Map Display:

- Reference Information Window - need to add GEOREF coordinates.
- need to add ability to degrade threat rings for various altitudes.
- View Target Dialog Box - need to add scrollable list box to choose target from.
- Reference Point Dialog Box - need to allow selection of distance ticks--NM, SM, KM--this should change the tick marks and the distance in the Reference Information Window. (partially done)

5/07/1989 5:14
Circ: reviewing old Input New Target Display
Idea: for "Trace Spotted By..." Dialog box add a text box that opens when "other" is selected so user can tell what the "Other" trace is.

5/07/1989 16:09
Circ: reviewing Map Display
Idea: Should the user be able to change target location from the map display?

5/07/1989 16:17
Circ: review of Map Display
Idea: Target sysmbol should show mission number and callsign (memory aid)

5/07/1989 16:40
Circ: review of "input new target"
Idea: Mission number sho   ie input automatically by the DSS.

5/07/1989 17:21
Circ: review of "mission planning display"
Idea: if info is input but classification block is left empty, can an Knowledge System assign the classification?

5/07/1989 18:50
Circ: review of "SARTF Display"
Idea: need to add duration column to "player" child window.

5/13/1989 14:54
Circ: working on Map Display
Idea: need access to the DSS default values through either -
1) a "System" menu added to the CSAR AIDE menu bar, or
2) a "System defaults" choice added to the "Help" menu.

5/14/1989 11:16
Circ: working on "Mission Planning Display" frame
Idea: DSS should take information input in the Visual Contact and Trace Spotted Dialog Boxes and compare it to target information in the Target Database to see if there is any correlation.

Appendix I: Hippenmeyer and Valusek's Adaptive Design Methodology for DSS Development (Hippenmeyer and Valusek, 1989:14-18)

Hippenmeyer and Valusek approach adaptive design from the more traditional "builder as designer" approach. While this seems to be the *modus de jour*, it is often not the method most beneficial to mid-level military decision makers. Hippenmeyer and Valusek provide a methodology based on a combination of "specific mechanisms" of research in "adaptive design" and "rapid prototype" methodology from the *builder's* perspective using the framework of Pressman's "six stage iterative approach to rapid prototyping" (Hippenmeyer and Valusek, 1989:14). They tout their methodology as an "integrated development scheme" using the tools of adaptive design (Hippenmeyer and Valusek, 1989:14). Their approach is outlined below, and while it focuses on the builder's function in adaptive design, it offers keen insight into the general principles and overall process of adaptive design and is the best example available of conducting adaptive design with the builder acting as the designer.

1. Evaluate the software request and determine whether the software to be developed is a good candidate for prototyping.

- Is the problem unstructured or semi-structured?
- Does the problem dictate the use of dynamic visual displays or heavy interaction with the user?
- Are algorithms or combinatorial processing that require evolutionary development necessary?

Any of the above conditions would mean the problem is a good candidate for prototyping.

2. Given an acceptable candidate project, the analyst develops an abreviated representation of requirements.

- The user and builder (functioning as designer) construct concept maps describing and bounding the application problem domain.
- The builder/designer (with frequent interaction with the user) constructs the storyboard from the above map.
- The user, designer, and builder review the storyboard and establish a realistic target subset (kernel) for prototype implementation. The kernel is chosen based on relative importance and feasibility.

3. After the representation of requirements has been reviewed, a set of abbreviated design specifications are created for the prototype.

- Based on the builder's experience, the complexity of the problem, and the choice of development environments, the builder may choose to use a formal design technique or may choose to simply perform a basic high level structural decomposition.

4. Prototype software is created, tested, and refined.

- Ideally, DSS generators and pre-exising software building blocks are available and used to create the prototype in rapid fashion.
- Each sub-routine module is tested separately before inclusion in the prototype.
- Aim for high module cohesion and low inter-module coupling.

5. Once the prototype has been tested, it is presented to the customer, who 'test drives' the application and suggests modifications.

- The user uses the hook book as he test drives the prototype to suggest modifications.
- The user, designer, and builder sort and prioritize the collected hook book entries considering them for implementation.
- The user modifies his storyboards to reflect the evolving system. The storyboard, however, is maintained separately from the working prototype.

6. Steps 4 and 5 are repeated iteratively until all requirements are formalized or until the prototype has evolved into a production system.

- Depending on the direction of the evolutionary process, the user may elect to "freeze" a mature set of requirements and allocate the necessary resources to implement a full scale production model of the prototype system or the prototype may evolve into a full features production system.

Bibliography

Air Force Center for Studies and Analysis. 1988 Catalog of Computer Simulation Tools Used by the United States Air Force Center for Studies and Analysis. Washington: HQ USAF, August 1988.

"Air Force Selects Computer Sciences Corp. To Develop Information Processing System," Aviation Week & Space Technology, 130: 51-52 (23 January 1989).

Air Land Forces Applications Agency [ALFA]. Multi-Service Procedures for Combat Search and Rescue--Development Draft, Langley AFB, VA: Air Land Forces Applications Agency, May 1988.

Alavi, Maryam and H. Albert Napier. "An Experiment in the Adaptive Design Approach to DSS Development," Information & Management, 7: 21-28 (February 1984).

Anderson, William C. Bat-21. Englewood Cliffs, NJ: Prentice-Hall Inc., 1980.

Andrews, Nancy, and Craig Stinson. Running Windows: The Microsoft Guide to Windows 2.0, Windows/286, and Windows/386. Redmond WA: Microsoft Press, 1986.

Andriole, Stephen J., Mark Akey, Kirk Dunkelberger, and Philip J. Millis. "Storyboarding for C2 Systems Design: A Combat Support System Case Study," paper presented at the Fifth Annual Workshop on C2 Decision Aiding. Kansas City MO, October 1987.

Bridges, Col John F., 23rd Air Force Assistant Deputy Chief of Staff for Plans and Programs. Personal interview. Dayton OH, 25 October 1988.

Capacik, Lt Col, MACOS/XON. Personal correspondence. Headquarters Military Airlift Command, Scott AFB IL, 9 June 1988.

Electronic Systems Division, Air Force Systems Command. Contract F19628-89-C-0007 with Computer Sciences Corporation, Attachment 3. Hanscom AFB MA, 14 September 1988.

Every, Martin G. Navy Combat Search and Rescue. Naval Air Systems Command Contract N00014-77-C-0253. Falls Church, VA: September 1979 (AD-A074593).

Fletcher, Maj Charles D. Design Requirements for a Decision Support System for the Dynamic Retasking of Electronic Combat Assets. MS Thesis, AFIT/GST/ENS/87M-3. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1987 (AD-A190-762).

Hippenmeyer, Capt William and Lt Col John R. Valusek. Adaptive Design: A Decision Support System Development Methodology. Department of Operational Sciences Working Paper WP89-4. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1989.

Keen, P.G.W., and M.S. Scott Morton. Decision Support Systems: an Organizational Perspective. Reading, MA: Addison-Wesley, 1978.

Knittle, D.L., S. Ruth, and E.P. Gardner. "Establishing User-Centered Criteria for Information Systems: A Software Ergonomics Perspective," Information & Management, 11: 163-172 (November 1986).

Manning, W. B. Navy Combat Search and Rescue: A Need For Training. Student research paper. Naval War College, Newport, RI: 1986 (AD-B102-614).

Marsh, Lt Col Tom, Chief of Staff. Telephone interview. Headquarters Aerospace Rescue and Recovery Service, Scott AFB IL, 17 April 1989.

Mathus, LCDR Bill, Assistant Branch Chief. Telephone interview. USCG National Search and Rescue School, Governor's Island, NY, 24 Aug 1988.

McConnell, Malcolm. Into the Mouth of the Cat: The Story of Lance Sijan, Hero of Vietnam. New York: W.W. Norton & Co., 1985.

McFarren, Capt Michael R. Using Concept Mapping to Define Problems and Identify Key Kernels During the Development of a Decision Support System. MS Thesis, AFIT/GST/ENS/87J-12. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1987 (AD-A185-636).

Microsoft Corporation. Microsoft Windows User's Guide. Software documentation, Redmond WA, 1987.

Novak, Joseph D. and D. Bob Gowan. Learning how to Learn. New York: Cambridge University Press, 1984.

National Search and Rescue School. Training Objectives for a proposed course in Joint Service Combat Search and Rescue. U.S. Coast Guard Training Center, Governor's Island, NY, 1987.

Office of the Joint Chiefs of Staff. Catalog of Wargaming and Military Simulations Models. Washington: OJCS, 1988 (AD-A169-472).

Office of the Joint Chiefs of Staff. Department of Defense Dictionary of Military and Associated Terms. Joint Chiefs of Staff Publication Number 1. Washington: OJCS, 1 June 1979.

Roark, Maj Stephen M., Director of Rescue. Personal correspondence. HQ Fifth Air Force, HTACC, Osan AB, Republic of Korea, 5 January 1989.

Seagle, John P. and Salvatore Belardo. "The Feature Chart: A Tool for Communicating the Analysis for a Decision Support System," Information & Management, 10: 11-19 (1986).

Shadra, Ramesh, Steve H. Barr, and James C. McDonnell. "Decision Support System Effectiveness: A Review and an Empirical Test," Management Science, 34: 139-159 (February 1988).

Simon, Herbert A. Sciences of the Artificial. Cambridge, MA: MIT Press, 1969.

Sprague, Ralph H. Jr. and Eric D. Carlson. Building Effective Decision Support Systems. Englewood Cliffs, NJ: Prentice-Hall, 1982.

Sprague, Ralph H. Jr. and Hugh J. Watson, ed. Decision Support Systems--Putting Theory into Practice. Englewood Cliffs, NJ: Prentice-Hall, 1986.

Sullivan, Kerry J. Navy Combat Rescue: The Forsaken Mission. Student research paper, Naval War College, Newport, RI: 1983 (AD-A144-011).

Tilford, Capt Earl H, Jr. Search and Rescue in Southeast Asia, 1961-1975. Washington: Government Printing Office, 1980.

Valusek, Lt Col John R., Assistant Professor of Operations Research. Personal discussions. Department of Operational Sciences, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, July 1988 through June 1989.

-----. "Adaptive Design of DSSs: A User Perspective," DSS-88 Transactions, Proceedings of the Eighth International Conference on Decision Support Systems. 105-112. TIMS, 1988.

-----. Class lectures in OPER 652, Decision Support Systems. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, July-September 1988.

----- and Dennis G. Fryback. "Information Requirements Determination: Obstacles Within, Among and Between Participants," Proceedings of the Twenty-first Annual Computer Personnel Research Conference Co-sponsored with Business Data Processing. Association of Computing Machinery, ACM Order #443851, New York, 1985. Also in Information Analysis Selected Readings. Ed. by Robert Galliers. Sydney: Addison-Wesley Publishing Co., 1987.

WESTPAC RCC. Combat SAR Mission Folder (KAB From O-79 and associated worksheets). Kadena AB Japan: Western Pacific Rescue Coordination Center, June 1986.

Wilson, MAJ Gary. "Combat Seach and Rescue Today--Will It Still Work?" Air Land Bulletin, 88-2: 8-11 (30 June 1988).

Whitewater Group, Inc., The. Actor Language Manual. Software documentation, Technology Innovation Center, Evanston IL, 1989.

Wohl, Joseph G. "Force Management Decision Requirements for Air Force Tactical Command and Control," IEEE Tansactions on Systems, Man, and Cybernetics, Vol SMC-11, No. 9: 618-638, September 1981.

Woodfin, William C. Increasing Carrier Air Wing Combat SAR Capabilities. Student research paper, Naval War College, Newport, RI: 1987 (AD-B111-218).

Ziehm, Maj Bob, Rescue Force Structuring. Telephone interview. USAF/XOXFL, Washington D.C., 26 April 1988.

VITA

Captain Mark E. Bracich was born on [redacted], [redacted] He graduated from Culver Military Academy in 1976 and went on to the U.S. Air Force Academy from which he graduated in 1980, earning a Bachelor of Science Degree in History.

Captain Bracich is an Honor Graduate of Undergraduate Helicopter Training at Ft. Rucker, Alabama. As a CH/HH-3E Combat Rescue helicopter pilot stationed in the Philippines, his duties included tactics, mobility, operational plans, computer operations, current operations/scheduling, and exercise planning/controlling. In this last capacity, Captain Bracich was a key player in ensuring on-going large-scale Rescue training for aircrews and JRCC controllers in major exercises such as Team Spirit, Tangent Flash, and Cope Thunder.

Upon returning from overseas, he was assigned to the 1550th Combat Crew Training Wing where he served as Instructor Pilot, Flight Examiner, Executive Officer, and Academic Instructor. Among his duties as an instrutor, both in the classroom and in the air, was to ensure adequate understanding of the command and control relationships involved in Rescue efforts.

Captain Bracich holds a Master of Aviation Management degree from Embry-Riddle Aeronautical University, and is currently a graduate student in the Strategic & Tactical Sciences program at the Air Force Institute of Technology. He will be assigned to the HQ Military Airlift Command Analysis Group following graduation.

Permanent address: [redacted]

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GST/ENS/89J-2

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL (If applicable)
AFIT/ENS

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
Air Force Institute of Technology (AU)
Wright-Patterson AFB, Ohio 45433-6583

7b. ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
CSAR AIDE: DESIGN REQUIREMENTS FOR A COMBAT SEARCH AND RESCUE DECISION SUPPORT SYSTEM FOR JOINT RESCUE COORDINATION CENTERS (UNCLASSIFIED)

12. PERSONAL AUTHOR(S)
MARK E. BRACICH, Captain, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ______ TO ______

14. DATE OF REPORT (Year, Month, Day)
June 1989

15. PAGE COUNT
136

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
23	06	
25	05	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Search and Rescue (SAR), Combat SAR, Rescue Coordination Center (RCC), Adaptive design, Decision Support System (DSS), Command & Control Decision Aid.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Advisor: John R. Valusek, Lt. Col., USAF
Assistant Professor of Operations Research

17. COSATI CODES (Continued)

Field	Group
05	02
05	01
15	06

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
John R. Valusek, Lt. Col., USAF

22b. TELEPHONE (Include Area Code)
(513) 255-3362

22c. OFFICE SYMBOL
AFIT/ENS

DD Form 1473, JUN 86

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

This thesis is an application of a methodolgy being researched at AFIT to define requirements for decision aids. The specific application of interest is Combat Search and Rescue Command and Control at the Joint Rescue Coordination Center.

It covers the current status of information management and control in the JRCC and recommends the development of an integrated decision support system (DSS). Such a system should be designed to aggregate information, provide the user with modeling and "what if" capability, and present data and model results in a manner which facilitates the decision making process.

An adaptive design methodology was used to capture requirements and ensure the design suggested meets JRCC needs. Modifications to the methodology are suggested. The result of this effort may be used as the cornerstone for a Statement Of Need for an automated Decision Support System to aid decision makers in the JRCC.