AD-A199 038

(2)

Final Technical Report HEL-SBIR-Phase I-1/88
Contract Number DAAA15-87-C-0064
March 1988

EMBEDDED PROCESS MODELING, ANALOGY-BASED OPTION GENERATION & ANALYTICAL GRAPHIC INTERACTION FOR ENHANCED USER-COMPUTER INTERACTION:

AN INTERACTIVE STORYBOARD OF NEXT GENERATION USER-COMPUTER INTERFACE TECHNOLOGY

Stephen J. Andriole & Gerald W. Hopple*

Approved for public release;
distribution is unlimited.

Prepared for:

USA Human Engineering Laboratory
Aberdeen Proving Ground, Maryland 21005-5001

Prepared by:

International Information Systems, Inc.
802 Woodward Road
Marshall, Virginia 22115

(703) 364-1504

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

88

EXECUTIVE SUMMARY

This project began with the working premise that the interactive decision aids and support systems that the community has designed and developed over the past decade have largely been unusable by military personnel. There are a number of reasons for these successive failures. Early aids were methodology-driven. Early aiding concepts were constrained by slow, cumbersome and expensive hardware, and many of the systems developed before 1980 were bounded by available software languages and engineering principles. Just as devastating was the insensitivity to the actual problems that the systems were intended to address and the interaction requirements unique to military users and their unique computing environments.

Where did we go wrong? It must be appreciated that "analytical computing" -- the process by which we incarnate analytical software in an interactive computing framework -- is very, very new. Just two decades ago computing was largely restricted to highly scientific users who were happy with the opportunity to work on incredibly badly designed machines and virtually inoperable software. By 1980 expectations about the necessary distribution of computing power had grown to the point where it was considered routine to interactively plan, decide, forecast, and allocate. In reality, however, our systems could trace their lineage to their clumsy scientific predecessors and <u>not</u> to enlightened system concepts anchored solidly in user and problem (versus technology) considerations.

We began with a challenge to initiate work in a new area, an area that would widen the communications channel between men and computers. We conducted a requirements analysis for a specific domain -- Army tactical planning at the Corps level -- and then a user-computer interaction (UCI) requirements analysis, all with an eye toward the design of an interface that would support planning and be useful to inexperienced, computer-naive, intelligent, and infrequent users.

The actual steps taken during the project appear in order on the next page. Requirements drove the design and development of the advanced UCI system concept from which we, in turn, designed and developed a "storyboard" prototype. Storyboard prototypes are simulations of interactive systems designed to help validate requirements. Storyboard prototypes are intended to narrow the gap between perceived requirements and system performance.

The storyboard was also tested and "sized." Sizing refers to the process by which assessments are made about the steps necessary to convert (in this case) a "throwaway" storyboard prototype into an actual working or "evolutionary" prototype (which would be the focus of a Phase II effort, should there be one).

The system concept called for a new UCI. The figures that follow suggest the menu structure that we developed as well as how it might operate during some interaction sequences. Noteworthy are

PROJECT STEPS

USER-COMPUTER INTERFACE (UCI) REQUIREMENTS

PROBLEM-SPECIFIC VERSUS GENERIC REQUIREMENTS APPROACHES
SUBSTANTIVE/FUNCTIONAL PLANNING REQUIREMENTS
UCI/DISPLAY REQUIREMENTS

UCI TECHNOLOGY ASSESSMENT

"CONVENTIONAL" UCI TECHNOLOGY
"UNCONVENTIONAL" UCI TECHNOLOGY
UCI TECHNOLOGY OPTIONS
THE "MASTER" OPTION SET
HIGH PRIORITY OPTIONS

ENHANCED UCI STORYBOARD DESIGN & DEVELOPMENT

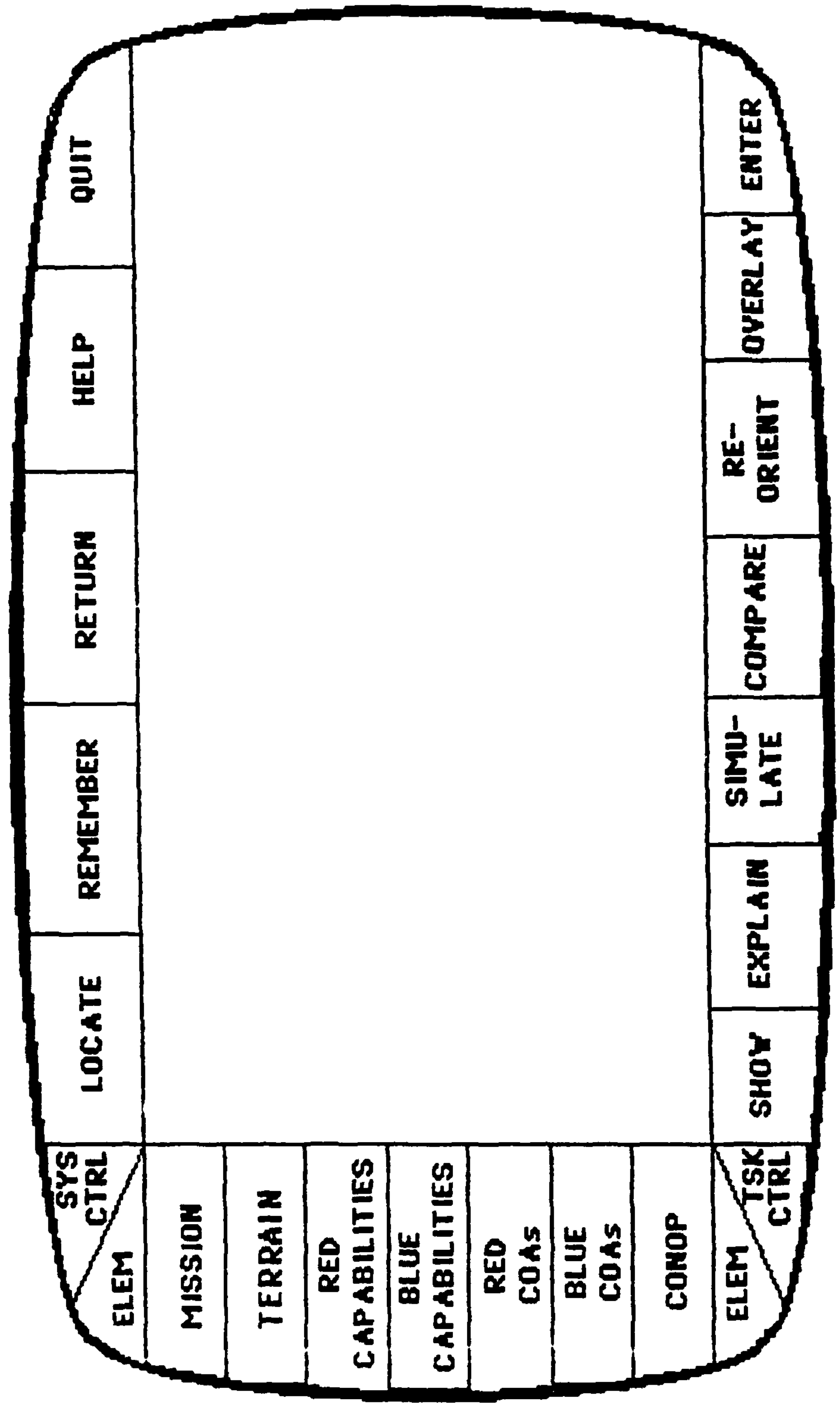
STORYBOARDING DESIGN & DEVELOPMENT PRINCIPLES
SCENARIO SPECIFICATION
THE HEL STORYBOARD PROTOTYPE

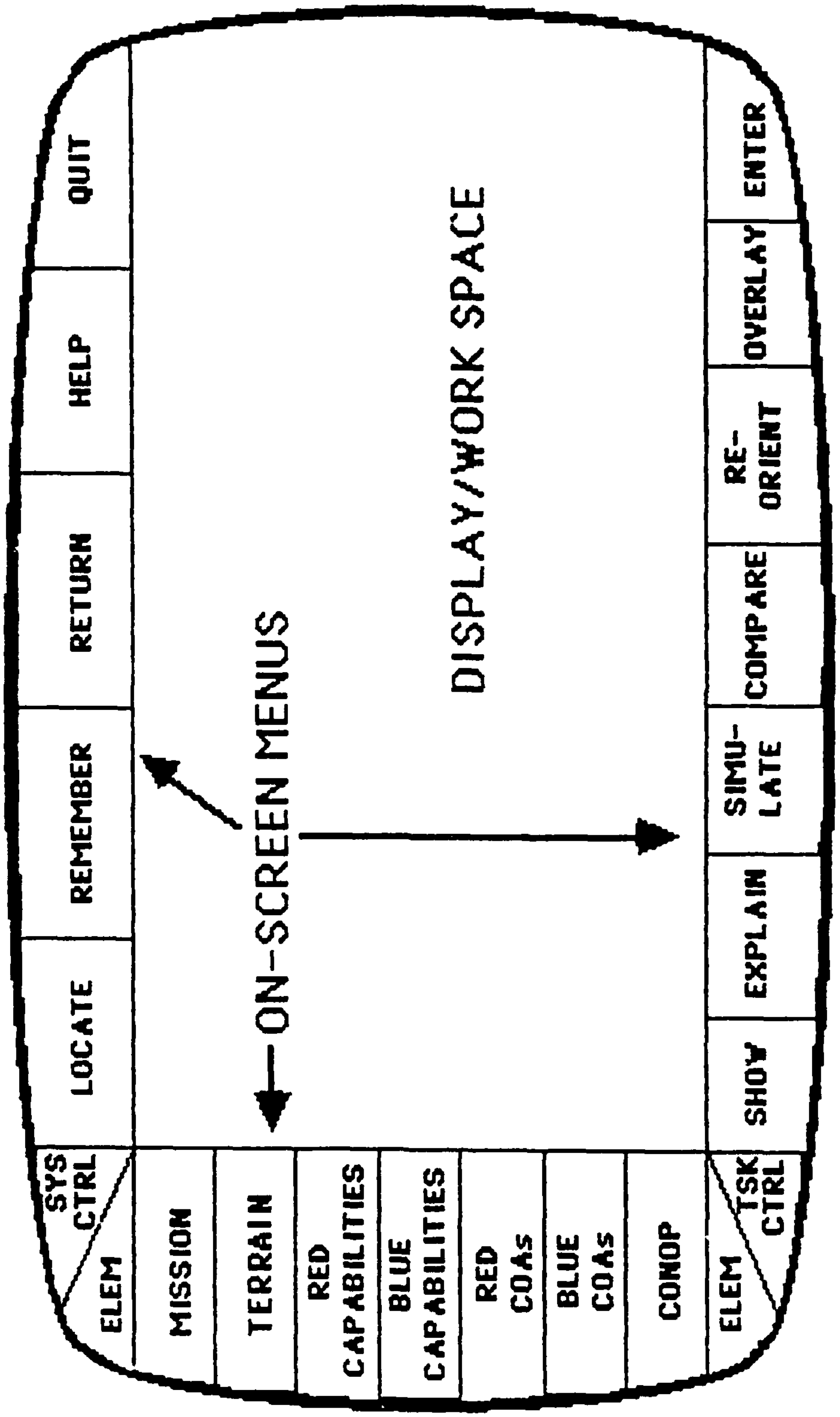
TESTING & EVALUATION

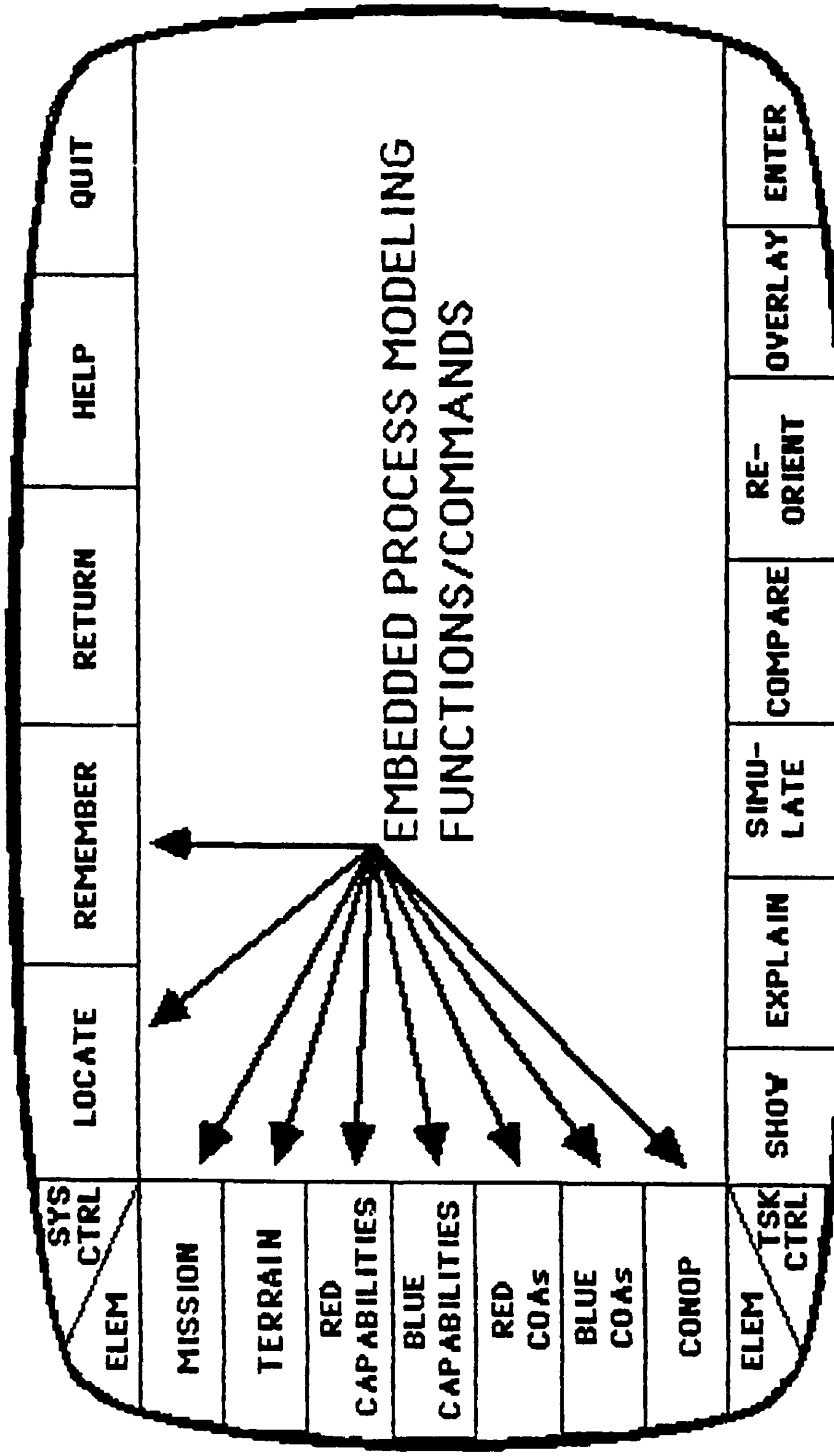
THE EVALUATION MODEL
QUALITATIVE & QUANTITATIVE ASSESSMENTS
SYSTEM "SIZING"
DATA & KNOWLEDGE BASE SPECIFICATION
SPECIFICATION OF CANDIDATE ANALYTICAL METHODS
UCI DEVELOPMENT ISSUES
SOFTWARE ENGINEERING IMPLICATIONS

CONCLUSIONS & RECOMMENDATIONS

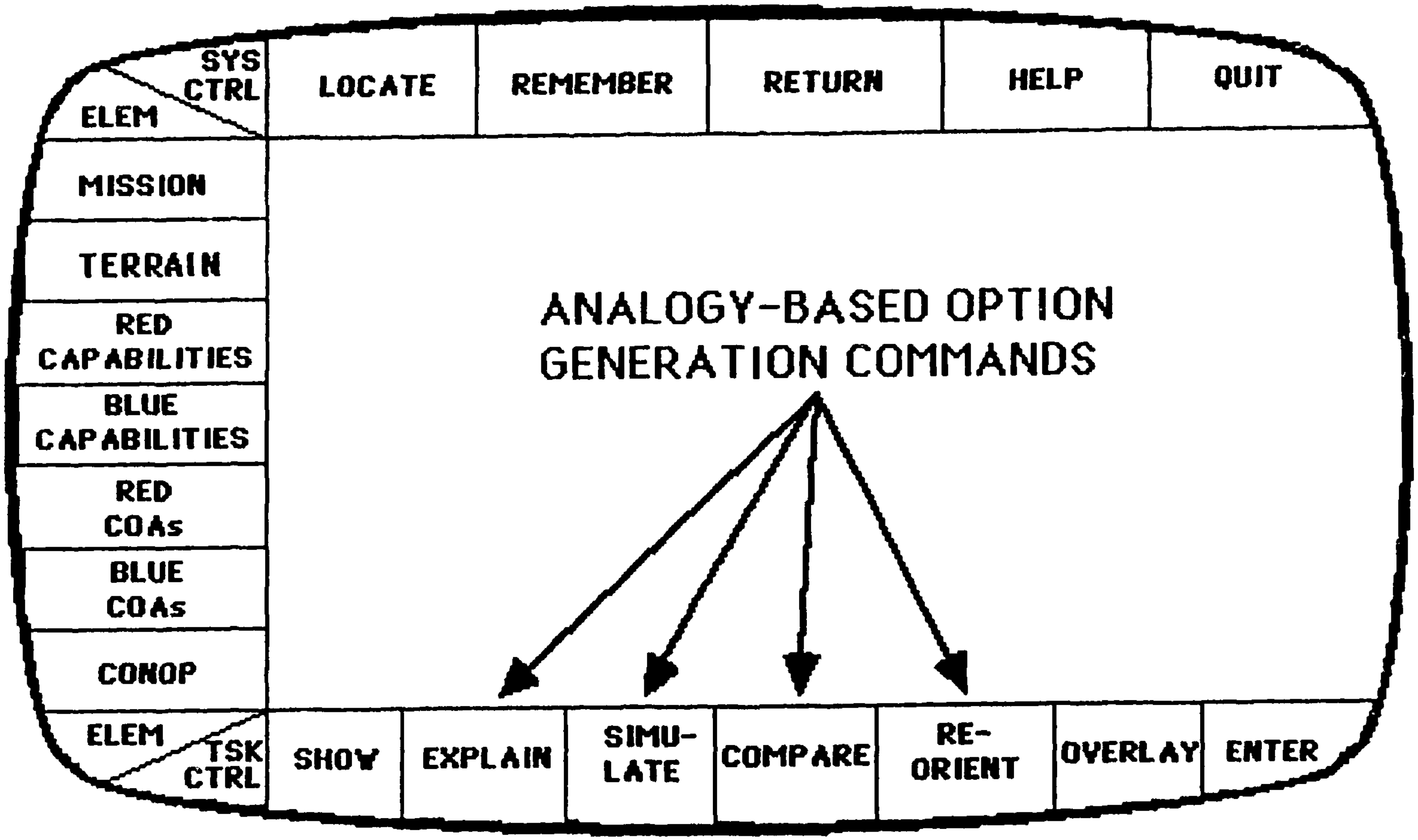
UCI HYPOTHESIS DEVELOPMENT
PROTOTYPE DEVELOPMENT
OTHER ARMY DOMAINS
PHASE II OPTIONS

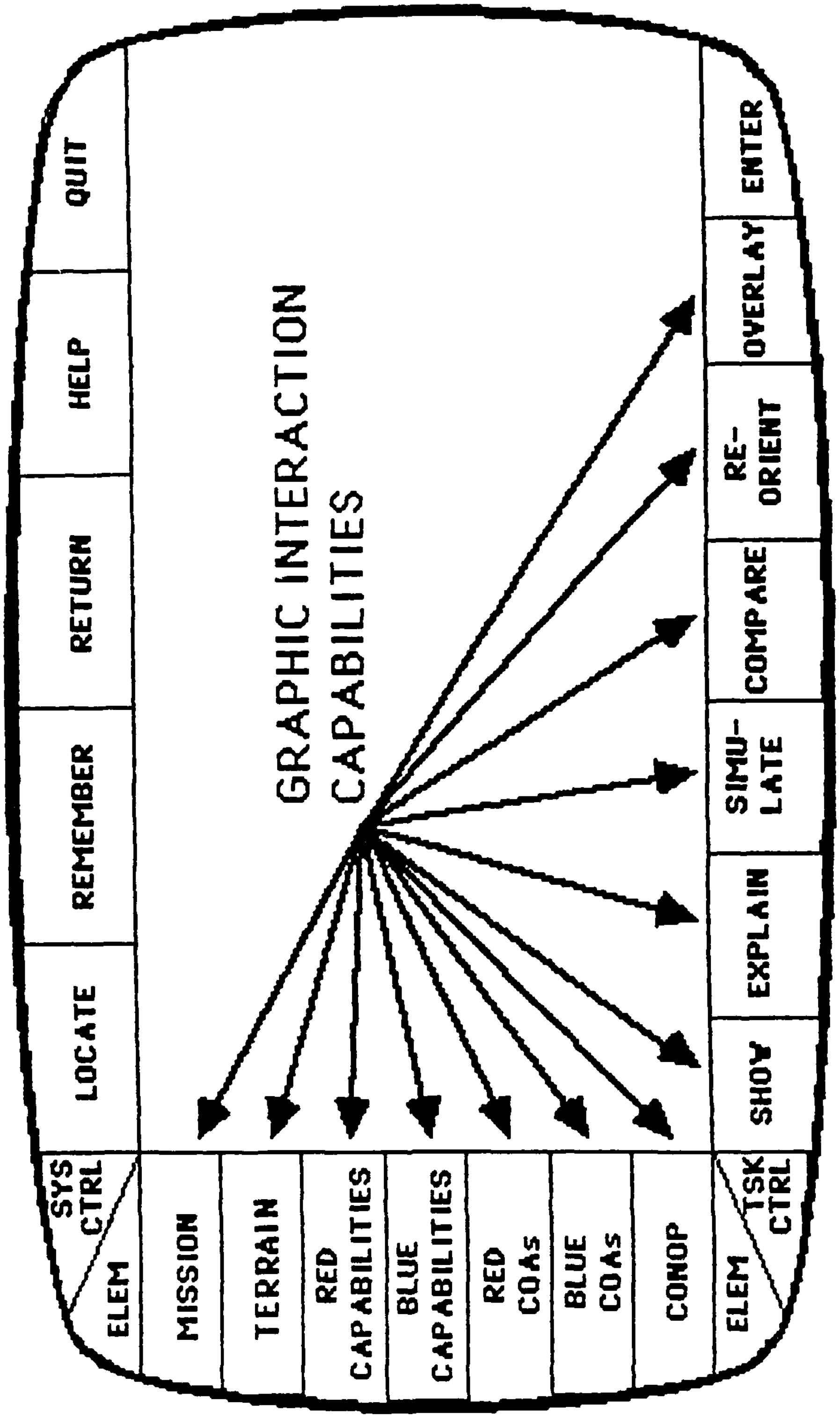

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

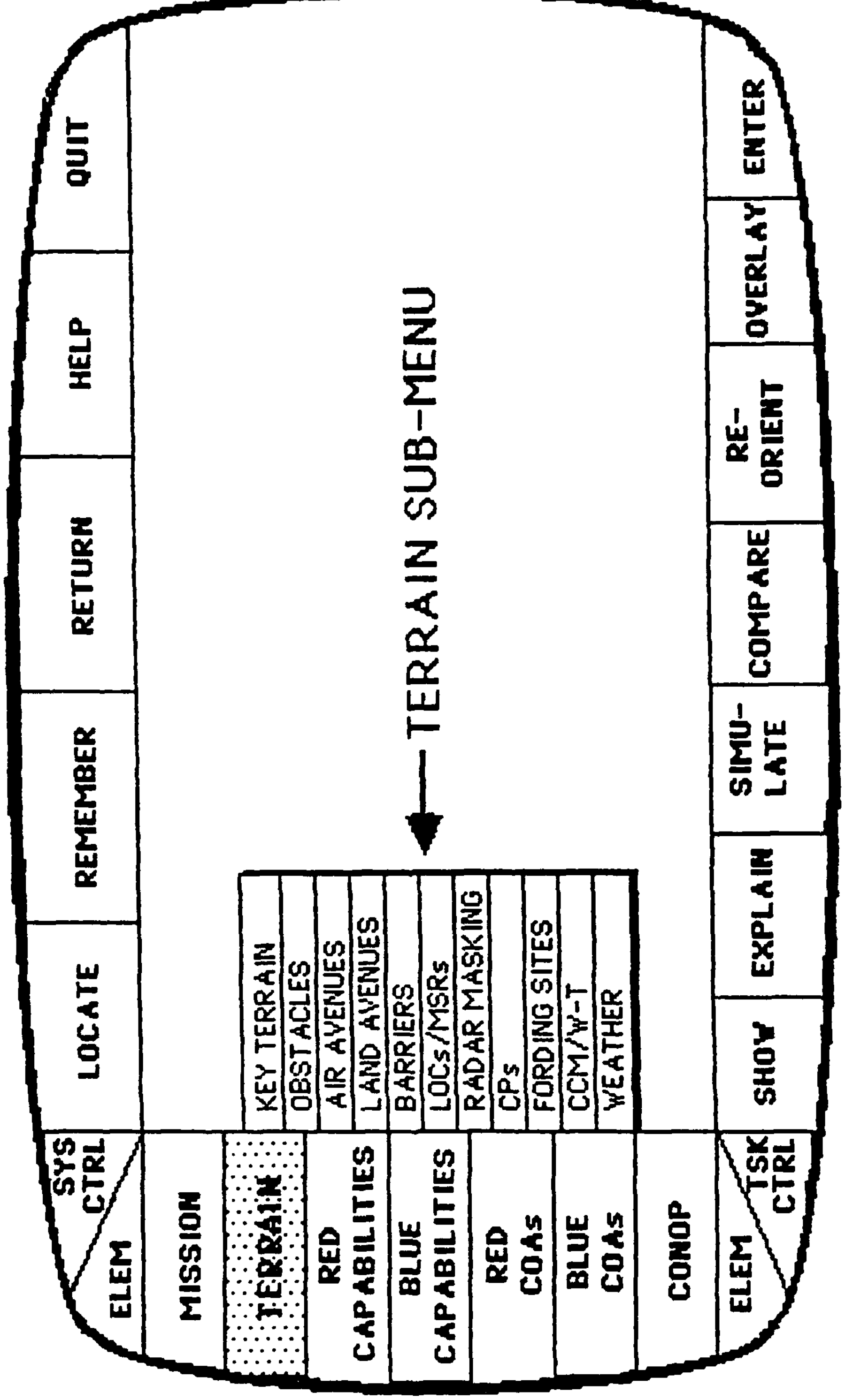

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ON-SCREEN MENUS
DISPLAY/WORK SPACE
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

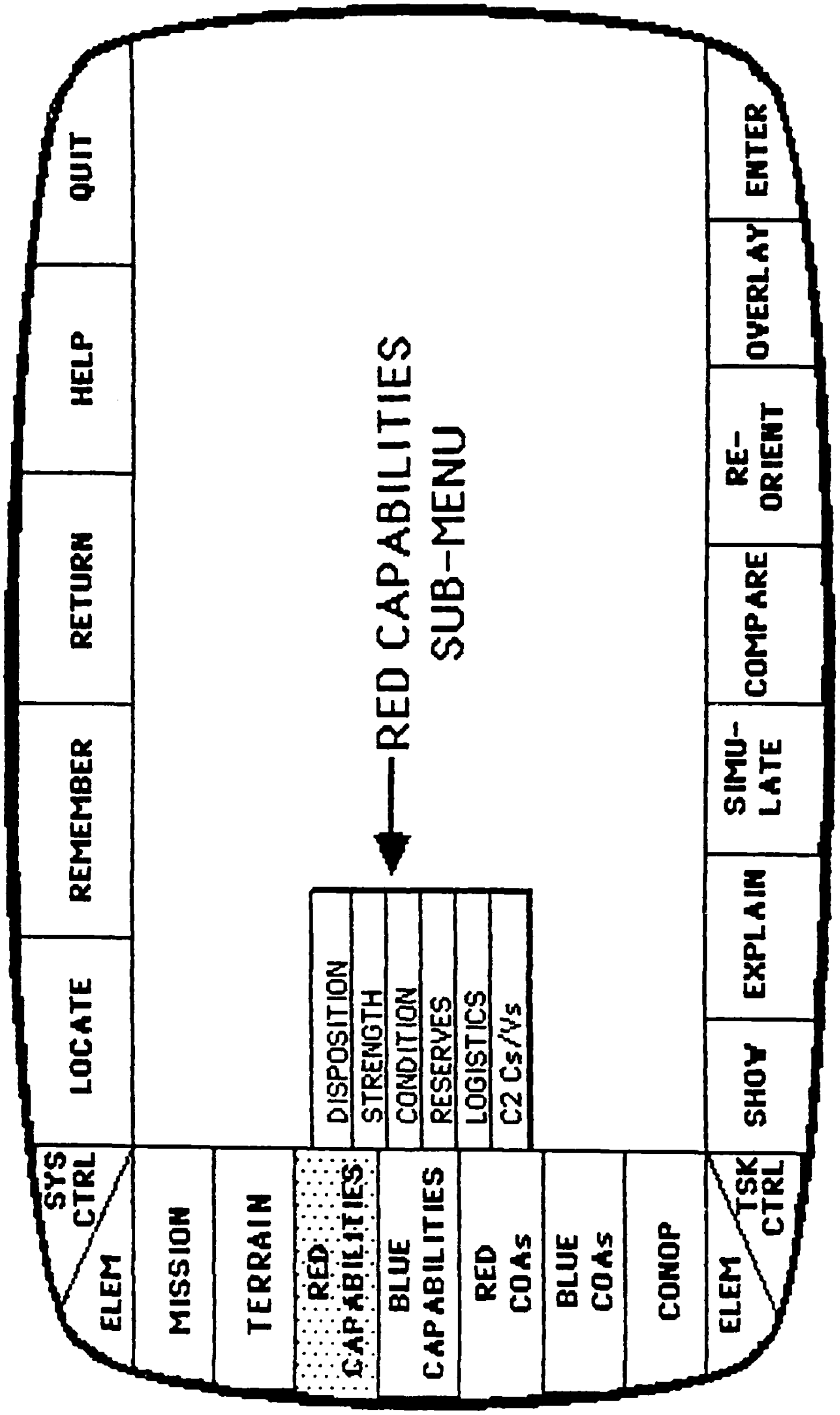


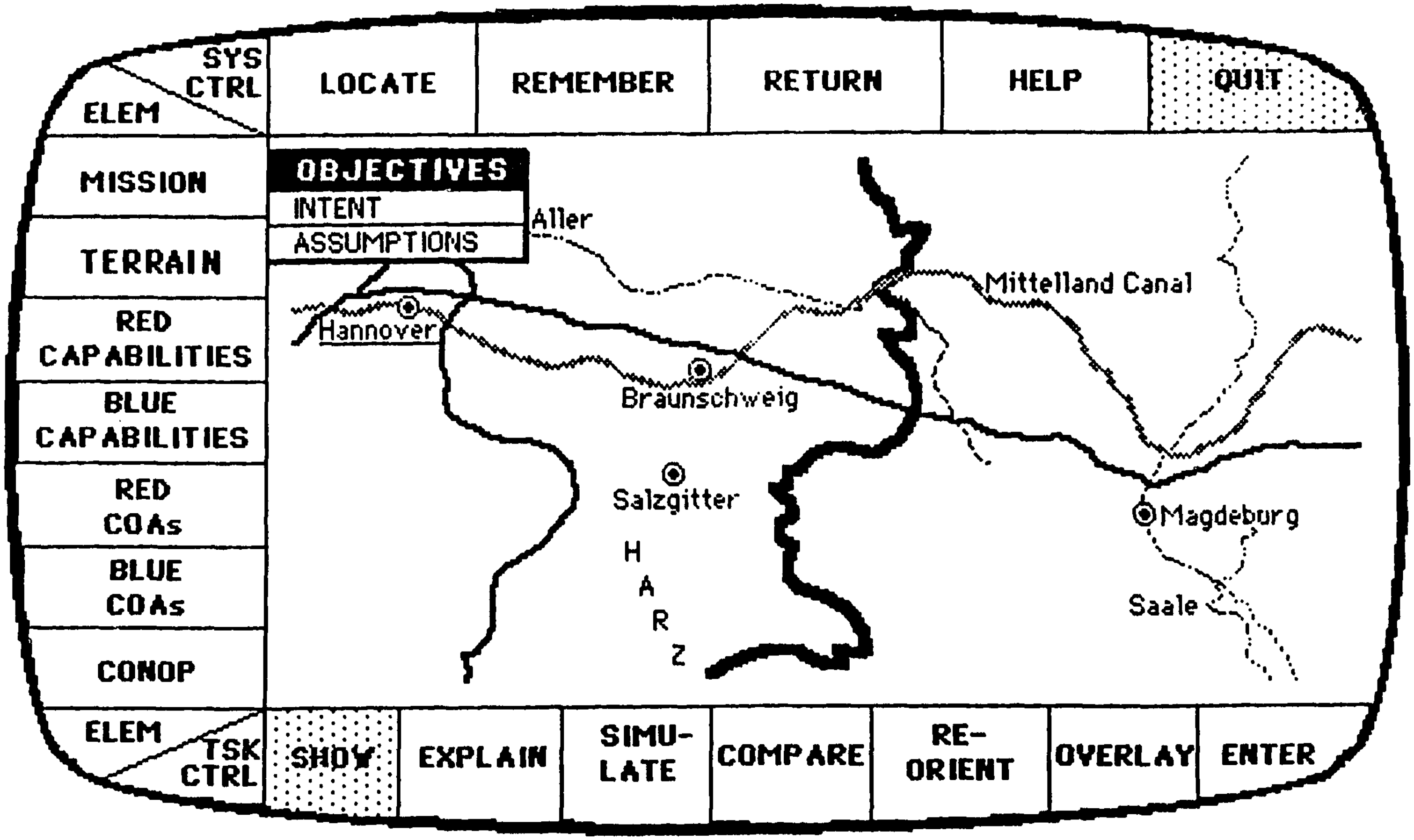
EMBEDDED PROCESS MODELING FUNCTIONS/COMMANDS
SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
TSK CTRL
ELEM
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ANALOGY-BASED OPTION GENERATION COMMANDS
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

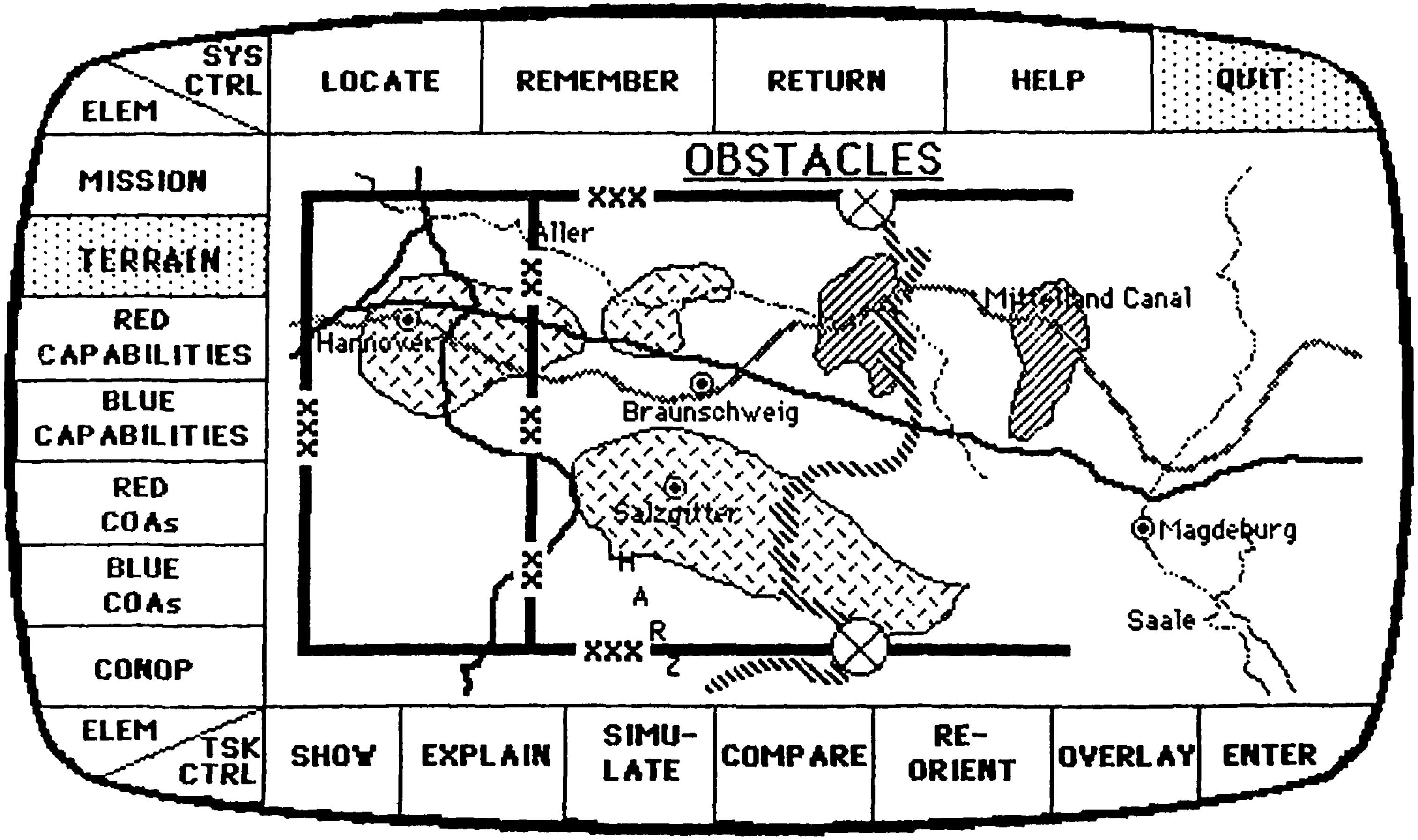

GRAPHIC INTERACTION CAPABILITIES
SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
TSK CTRL
ELEM
SHOW
EXPLAIN
SIMU- LATE
COMPARE
RE- ORIENT
OVERLAY
ENTER


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
KEY TERRAIN
OBSTACLES
AIR AVENUES
LAND AVENUES
BARRIERS
LOCs/MSRs
RADAR MASKING
CPs
FORDING SITES
CCM/W-T
WEATHER
TERRAIN SUB-MENU
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

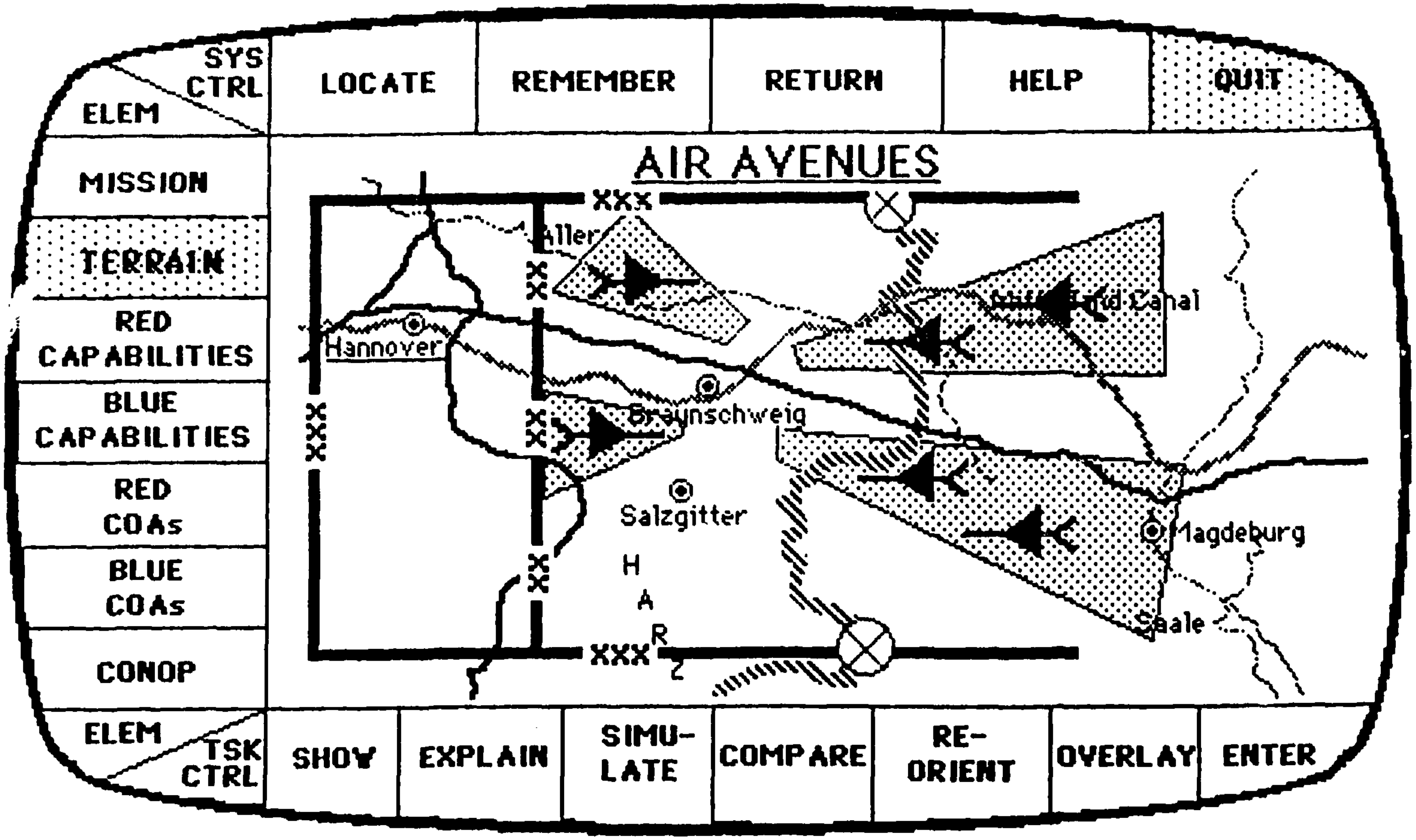

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
DISPOSITION
STRENGTH
CONDITION
RESERVES
LOGISTICS
C2 Cs/Vs
RED CAPABILITIES SUB-MENU
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER



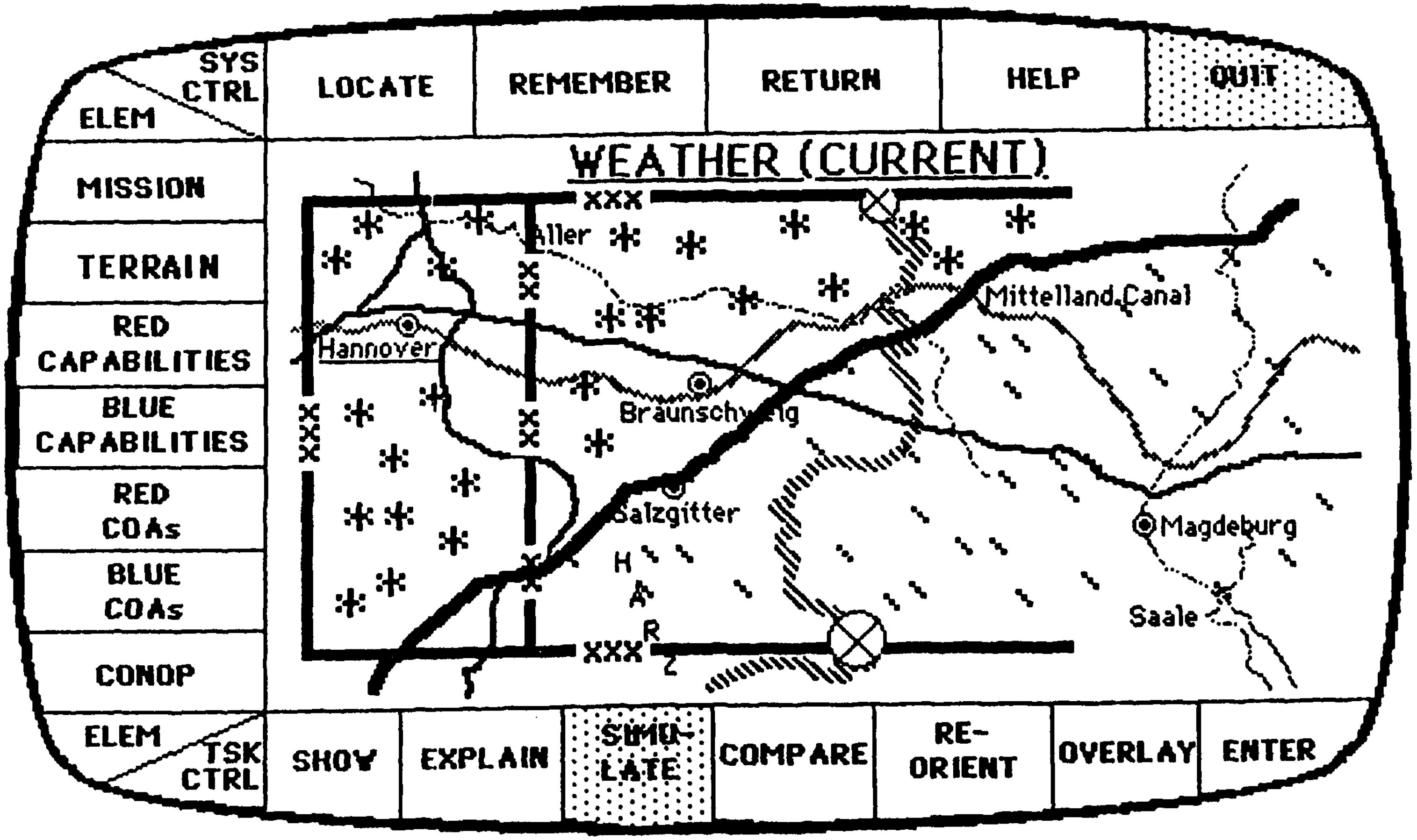
This display suggests how a user can execute planning tasks; in this case, the user is interested in the **Mission** and has selected information regarding the Mission's **Objectives** . . .



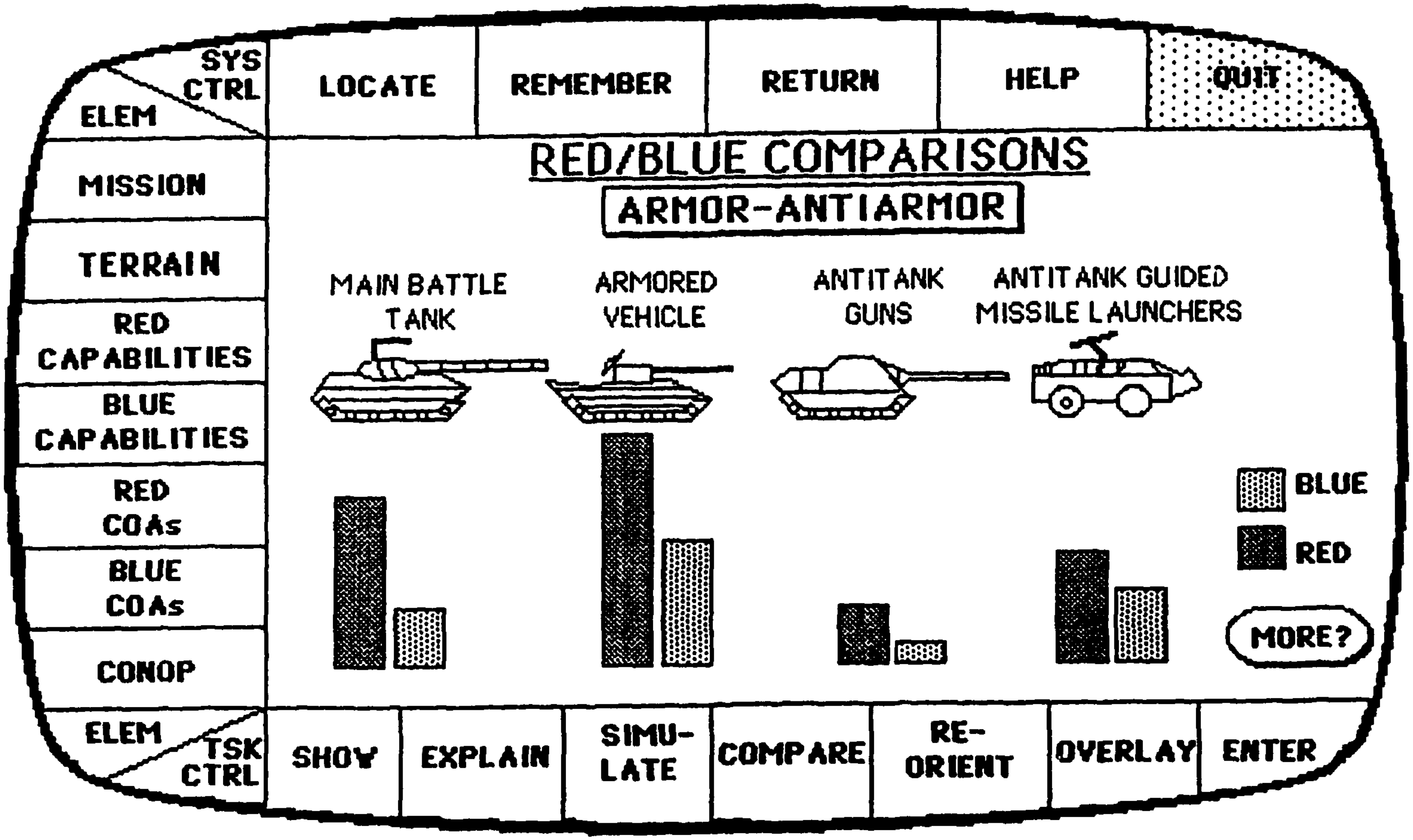
The system displays **Obstacles** to the planner . . . here too the user has the option of requesting more detail about the obstacles via the **Explain** command . . .



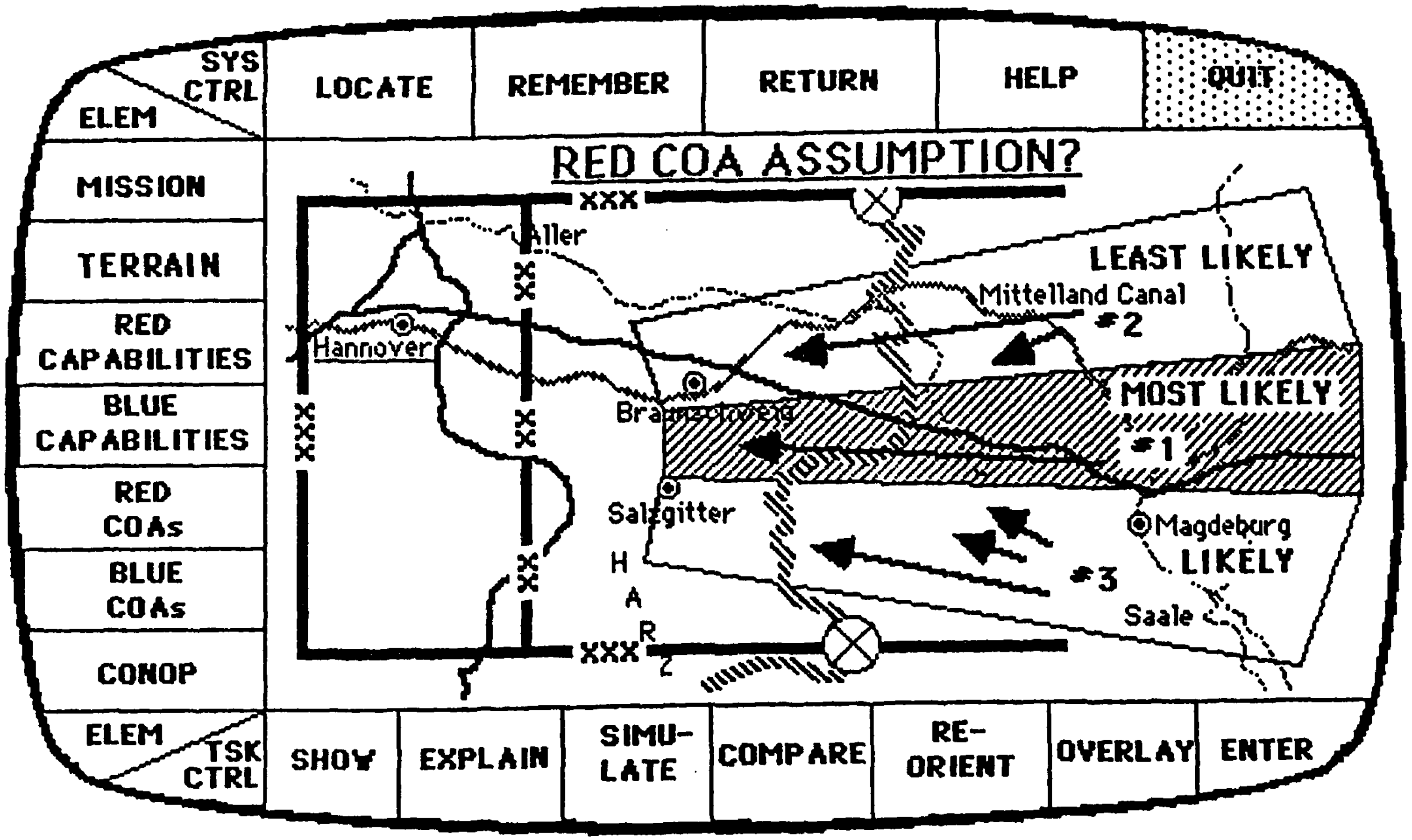
Air Avenues are displayed to the user . . .



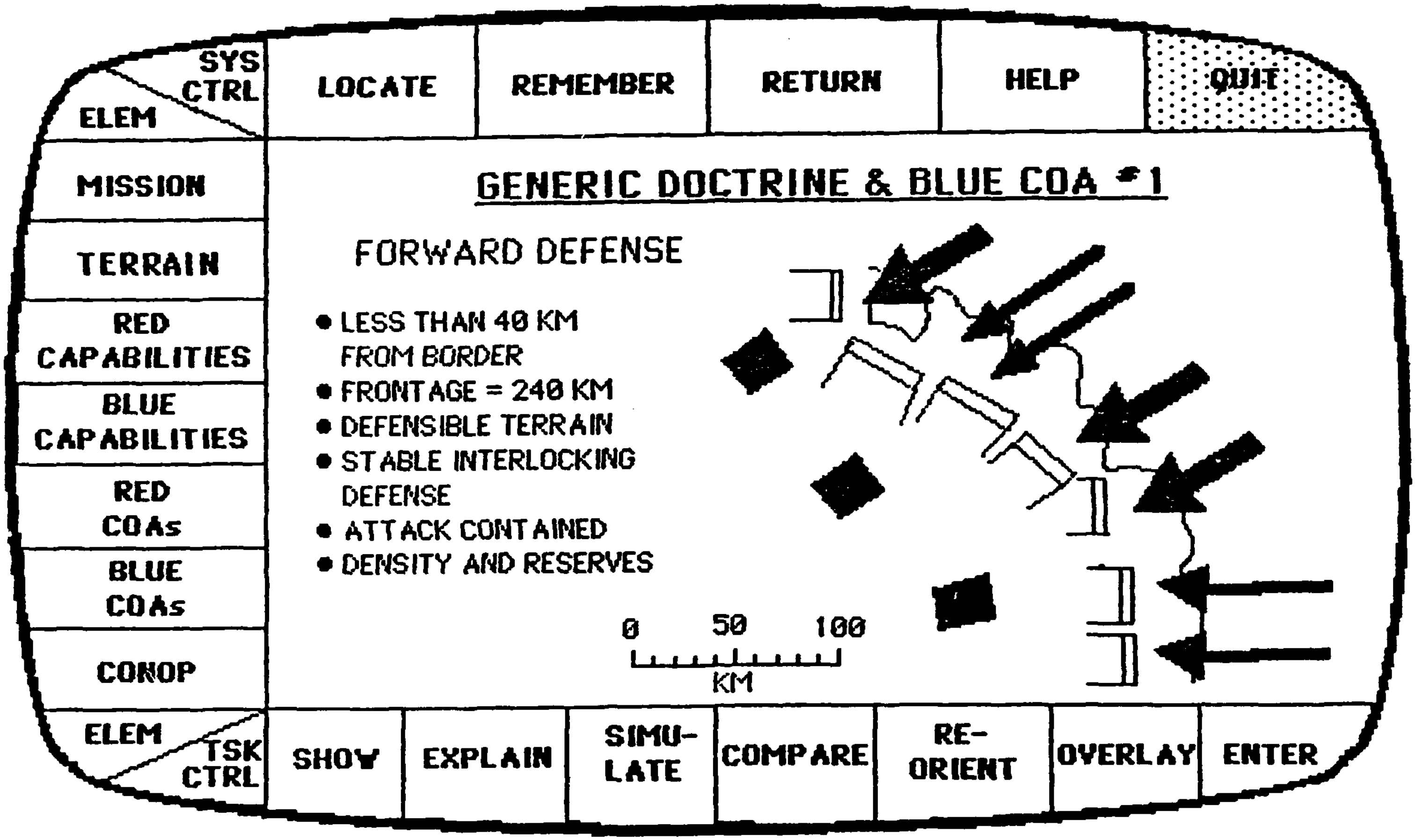
Weather is displayed to the planner -- who then decides to "see" the weather in a simulation (via the **Simulate** command) over time . . .



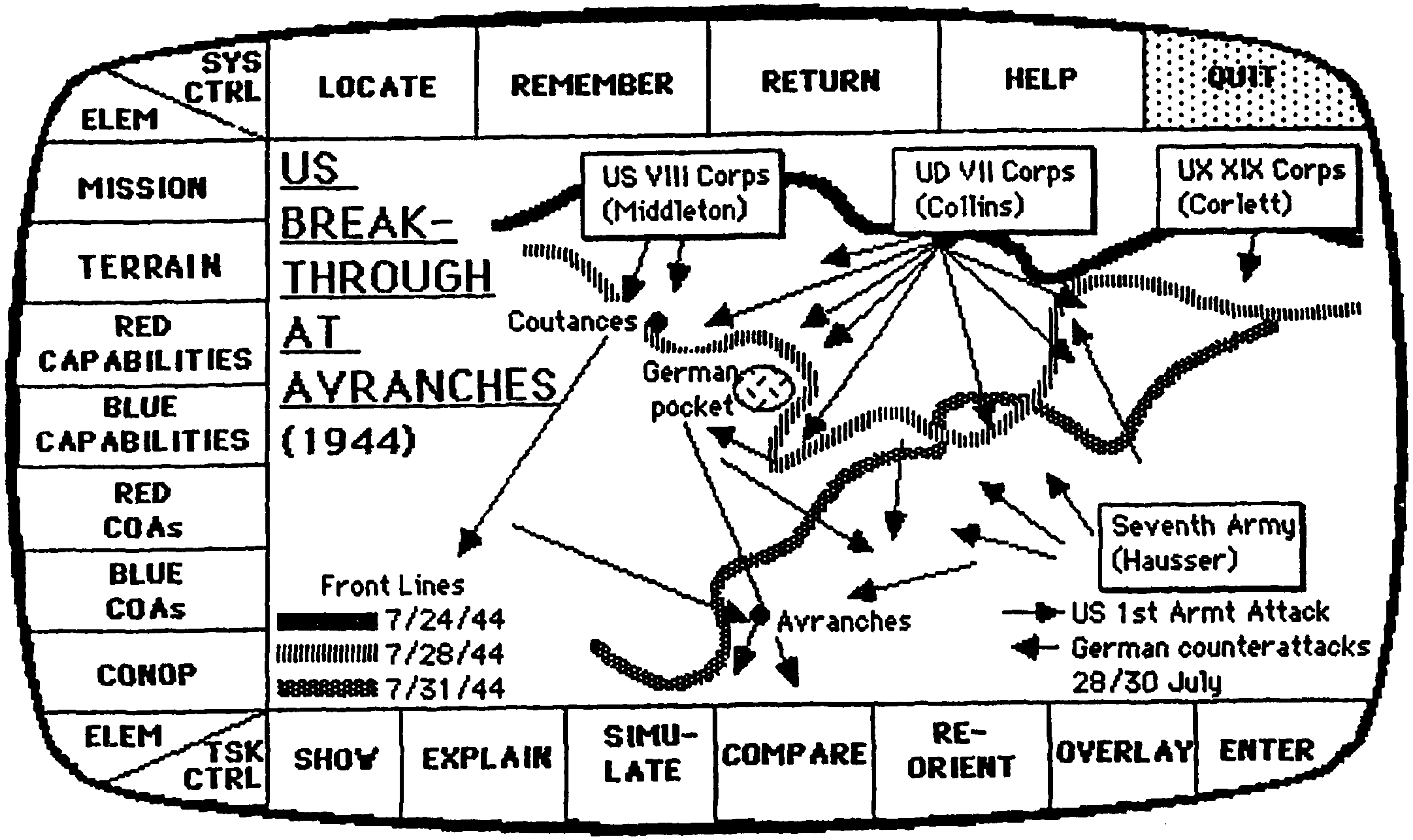
The system displays the data requested graphically. Note that all primary displays are graphic. The use of "analytical graphics" occurs throughout the system concept since our requirements data suggests that planners crave graphic and pictorial data for option generation and evaluation . . . visual cues are used throughout the storyboard to accelerate the planning process . . . Note also the **More?** menu box on the screen . . .



The system first requests information about what the planner thinks Red will do based on his analysis of hypothetical Red COAs . . the planner selects **Red COA #1** by clicking directly on the COA zones previously displayed to the user . . .



Generic doctrinal display . . .



The system presents a set of plans that are relevant to the planning problem at hand . . .

the use of a set of on-screen commands, the non-use of keyboard, and the use of commands that are intuitive to the inexperienced user. The system concept also calls for the extensive use of "analytical graphics," the use of "visual cognition" techniques, analogical reasoning processes, and embedded process modeling for system guidance and self-explanation.

The purpose of the research was to explore some new approaches to enhanced UCI, to break away from the "spreadsheet" mentality and inertia, and to determine the extent to which "unconventional" ideas could be applied to a real military domain, like tactical planning. The results of the research are very promising. We believe that a new UCI process has been specified in our initial prototype, a process that can be transferred to other similar domains. We also believe that the UCI process supported in the storyboard is precisely the kind that will help define future UCI requirements.

TABLE OF CONTENTS

APPENDIX A: "Intelligent Aids for Tactical Planning"

APPENDIX B: "TACPLAN - An Intelligent Aid for Tactical Planning"

*ACKNOWLEDGMENTS

Special thanks are extended to Mr. Clarence Fry, Chief, Aviation and Air Defense Directorate, and Mr. Larry Peterson, COR for the effort, of the Army's Human Engineering Laboratory for their guidance and support. We would also like to acknowledge the contribtuions of Drs. Gary Klein and Leslie Whitaker of Klein Associates, Inc. who contributed to the project's "perspective" and, specifically, Section 3.2 of this report.

1.0 INTRODUCTION

1.1 User-Computer Interface (UCI) Challenges and Opportunities

Computer-based problem-solving systems are often extremely difficult to use. The word processing program used to prepare this SBIR Phase I technical report -- Wordstar(TM) -- is a case in point. There are over a hundred and forty-four commands in Wordstar, but research suggests that the vast majority of users rely heavily upon but eight. In follow-up interviews to human factors studies, respondents stated that it is impossible to remember more than a few commands so they learn to rely upon those that are absolutely necessary to perform their duties. The commands themselves are non-mnemonic and often counter-intuitive, and the system's "help" commands were clearly designed with expert -- not novice -- users in mind.

Wordstar is far from unique. Other word processing, spreadsheet, decision option selection, and data base management programs suffer from the same inadequacies. International Information Systems, Inc. (IIS) has been actively involved in the design and development of decision aids and support systems for command and control (C2) since 1979 and has yet to conceive of a system that did not face significant user-computer interface (UCI) challenges. Sometimes these challenges were met; sometimes they were not.

It is imperative that we not only design systems with good UCIs

(KR)

but that we design systems that can get their users out of trouble as well. There are users that only use systems infrequently. Some have cognitive "styles" (for example, words versus numbers) that cause them to forget commands and command structures over time; and some are unable to compute analytically without interruption (which often results in their "losing their place" in the problem-solving process). Our interfaces must support primary and secondary problem-solving functions and perform duties as "monitors," "managers," and "navigators."

The research that documents the problems connected with poor UCIs and the need for embedded help includes work by Wilbert O. Galitz (1986), Ben Shneiderman (1980), Stephen J. Andriole (1986a, 1988), and Sidney L. Smith and Jane N. Mosier (1984). The Smith/Mosier study represents one of the most comprehensive studies undertaken in the area.

Sponsored by the Air Force, this study identifies hundreds of potential problems and offers just as many solutions. However, even in a study of this magnitude, embedded help receives very little attention. Novel approaches to the problem go virtually unmentioned.

There are a variety of opportunities that can first be defined as a set of goals. We need embedded functions capable of performing at least the following (Smith and Mosier, 1984; also see Section 3.0 of this report):

- Display of guidance information;
- Consistent display format;
- Consistent format of user guidance;
- Distinctive format for user guidance;
- Clear control labels;
- Consistent coding conventions;
- Familiar coding conventions;
- Task-oriented wording;
- Consistent grammatical structure;
- Flexible user guidance;
- Consistent feedback;
- Display identification;
- Feedback for user interrupt;
- Non-disruptive error messages;
- Logical menu structure;
- On-line system guidance;
- Task-oriented help;
- On-line training;
- Adaptive training; and
- Design change management, among many others.

These functions and capabilities represent "conventional" human factors approaches to enhanced user-computer interaction. In some contexts these and related approaches have worked very well; in just as many others they have not worked well at all.

What we need are solutions that solve the problem at a higher interaction level and ones appropriate to the military computing environment, an environment characterized by inexperienced users, "hostile" conditions, and rapid personnel turnover. More specifically, we need methods and techniques that would perform the following kinds of functions and provide their implied capabilities:

- On-line monitoring of user and system interaction;
- Monitoring of system functions;
- Constant navigational cues to users;
- Feedback and feedforward capabilities;
- Sequence/process management;
- On-line and after-the-fact audit trails of the problem-soling process;
- "Hold" and "wait" capabilities;
- Help via analogies;
- Help via "active" process modeling;
- Unobtrusive instruction;
- Graphic equivalence; and
- Graphic explanations of system functions, system data/knowledge, and problem-solving/inference-making/data base management procedures.

These and related functions and capabilities need to become essential elements of our systems, not addenda and certainly never incarnated in off-line user manuals or other "props."

This report describes a technical approach to the design, development, and testing of on-line, embedded UCI enhancements that appear in the second set of items above. The approach described here is eclectic, borrowing from human factors, cognitive psychology, analytical graphics, artificial intelligence, and visual cognition, among other disciplines and fields of inquiry. We designed an on-line aiding systems concept and then demonstrated the concept in a computer-based "storyboard." This report documents the steps we took as well as the results we achieved.

1.2 Project Objectives, Work Plan and Accomplishments

The ultimate objective of the research was to identify a set of methods, tools and techniques that would permit us to embed guidance, help, and training in interactive systems intended for use by inexperienced military personnel in "hostile" environments. The methods, tools, and techniques must permit the development of unobtrusive, system- and graphics-based routines. They must also permit the development of routines whose effectiveness can be measured.

The objectives for Phase I of the research included the following:

- The identification of a problem domain suitable for exploring the design, development, and testing of new methods, tools, and techniques for enhanced UCI;
- The application of the methods, tools, and techniques to the domain in the form of exemplar displays that represent alternative approaches to enhanced UCI;

- The development of the displays in the form of an interactive, computer-based "storyboard" that demonstrate where and how the enhanced UCI will operate; and

- The testing of the effectiveness -- the value-added -- of the functions and capabilities represented in the interactive storyboard.

1.2.1 Selection of the Target Domain - In order to lend credibility to the proposed research, we assumed that it was necessary to carry out the research in context; abstract solutions tend to remain abstract. It was also important for us to select a domain representative of the tactical environment in which Army problems are solved.

We selected the domain of tactical planning. For the past several years we have worked in this area; we have also designed and developed two interactive planning aids (TACPLAN [for tactical planning] and INTACVAL [for intelligent tactical plan evaluation]; Andriole [1986b], Hopple [1986], Andriole and Hopple [1987]). These experimental aids are ultimately intended for use by Army G-3 personnel working at the Corps level. The aids are designed to support tactical plan generation and evaluation. We designed the aids after extensive knowledge acquisition from real tactical planners at the Army War College in Carlisle, PA. We conducted a series of exercises in 1984 and 1985 designed to identify critical planning tasks and sub-tasks. In the process we conducted user profiles of the commanders and staff personnel that would eventually use the systems. It became clear that while they were sophisticated planners they were unsophisticated users of computer-based problem-solving systems. This finding, among several others, suggested that we design aids that were graphically intensive, capable of explaining themselves, and "cognitively consistent" with the way planners plan with grease pencils, maps, and acetate overlays.

TACPLAN and INTACVAL do not, however, have elaborate help or training capabilities. Nor do they exploit creative, hybrid UCI solutions to the perennial interaction problems.

While we did not re-program TACPLAN and/or INTACVAL with elaborate on-line, embedded UCI enhancements in Phase I, we did select the domain of tactical planning and the TACPLAN/INTACVAL aiding concepts to design some new interfaces that will support tactical planners.

The domain of tactical planning is a good one because of the current interest in interactive "intelligent" and "un-intelligent" planning aids and support systems, especially as evidenced in large Defense Advanced Research Projects Agency (DARPA)/Army programs like AirLand Battle Management, the Army's ARES Project at the Center for Tactical Computer Systems at the U.S. Army's Communications-Electronics Command at Fort Monmouth, and the basic research in planning and decision-making that the Army Research Institute (ARI) is supporting.

It is also a good domain because typical users of planning aids and support systems are inexperienced with analytical computing and require their systems to be helpful and "user friendly" in every sense of the term.

Finally, our experience in the area enabled us to hit the ground running. Our work for ARI and CECOM/CENTACS since 1983 has focused specifically on the design and development of interactive aids to support Corps Commanders and their staffs. This work has required us to conduct extensive requirements analyses and to experiment with alternative systems concepts.

We stayed in the domain of tactical planning at the Corps level. We applied our ideas for enhanced UCI in the context of tactical planning and tested the ideas with expert planners (see Section 5.1.4 below).

1.2.2 Embedded Techniques for Enhanced UCI - This task involved the design of a set of techniques that would enhance the processes by which plans are developed and evaluated in a simulated interactive system.

We derived the ideas from three general concepts for enhanced UCI:

- Embedded process modeling for system status monitoring and management;
- Analogy-based graphic explanation and guidance capabilities; and
- Graphic (system) navigational aids.

All three ideas support enhanced UCI through training and via on-line system operation, all as discussed below and represented in the storyboard (see Section 4.0).

Embedded process modeling for system status monitoring and management is a technique that seeks to provide users with a top-down view of the functions, tasks and sub-tasks that they can perform with the system. Too often it is impossible to "see" the overall structure of the problem-solving process that the system is intended to support. Embedded process models can serve as on-line compasses, making it very difficult for users to get lost during the problem-solving process.

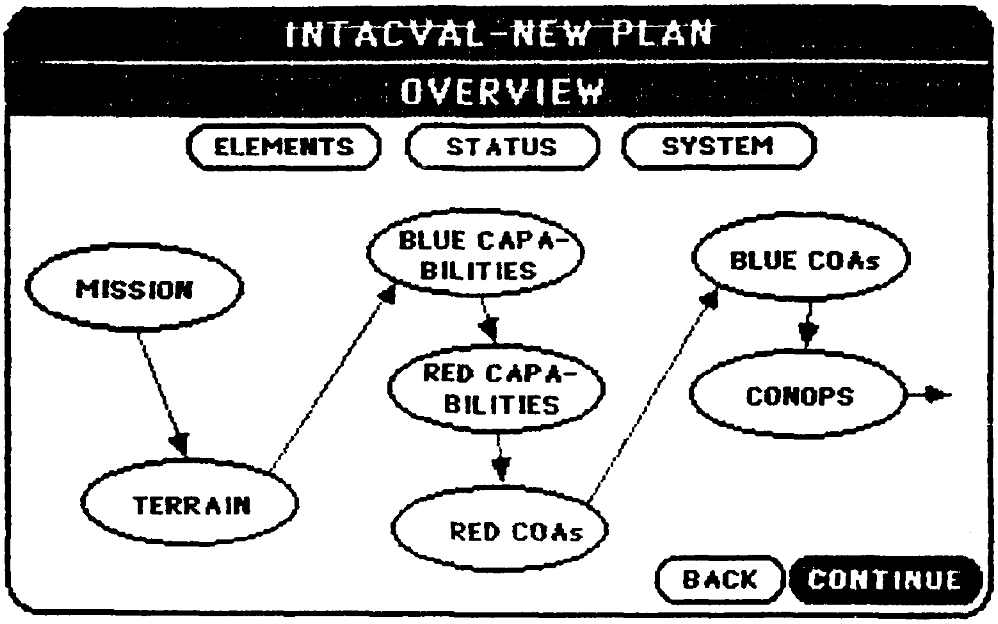
The Wordstar example may be worth returning to here. As users process words with the program there are no clues provided regarding where the user is within the larger word processing process. Ideally, it would be possible for novice users to "see" that they are now entering data into the system -- which presumes that they have opened and named a file and that they will close and print at some point in the future. These steps in the process might be communicated to users graphically or simply alphanumerically.

Mechanics using interactive systems to repair automobiles would also benefit from a top-down view of the repair process, just as data base managers would find it helpful to see why certain data must be stored, retrieved, and displayed.

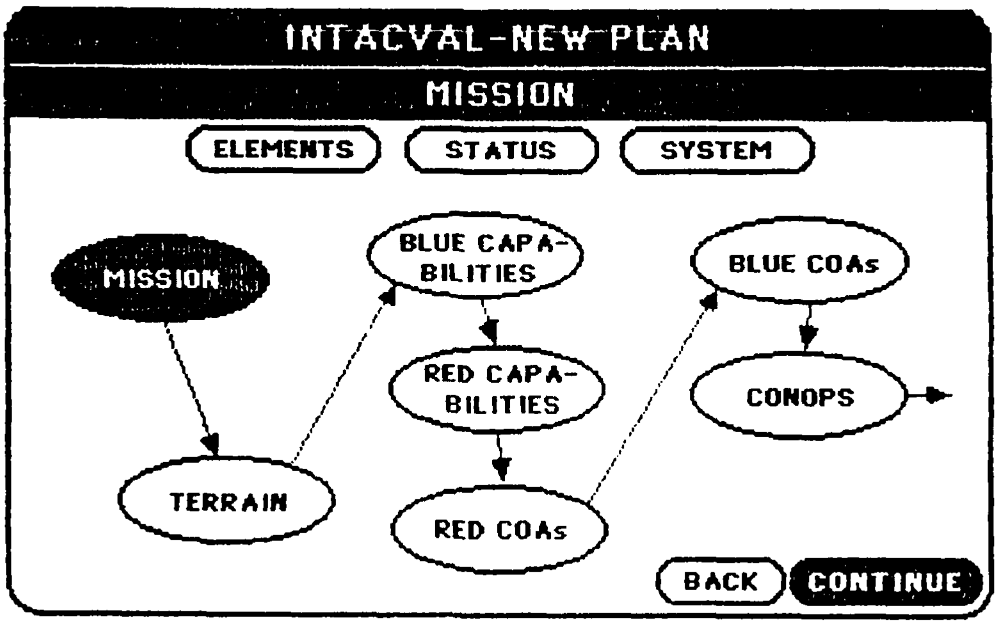
How might process models be used? First, they should be conceived as on-line compasses capable of communicating direction and purpose to users. When designed properly, embedded process models can keep track of the problem-solving process and report to users regarding where they are in the process, where they have been, and what they have left to do. Embedded process models can also be used to accelerate the training process, since each step in the process can be organized as a "lesson."

All of this is illustrated below in some simple process models of the tactical planning process. Note that the process model contains information about the steps that planners must take to develop a Concept of Operations. As the planner completes successive steps, the process model updates itself. A quick glance at the model at any point in the problem-solving process would reveal to the planner exactly where he was in the process and what he had left to do (see the figures below). A planner that was interrupted from a planning session could return to see exactly where he was in the process, just as a trainee could see the steps that comprise the planning process and how they interrelate.

We extended the examples presented here several levels down to the point where each of the top-level steps in the process has equivalent sub-levels and so on down to the lowest diagnostic level. In other words, we built a hierarchical process model and embedded the hierarchy into the aid. Depending where the planner

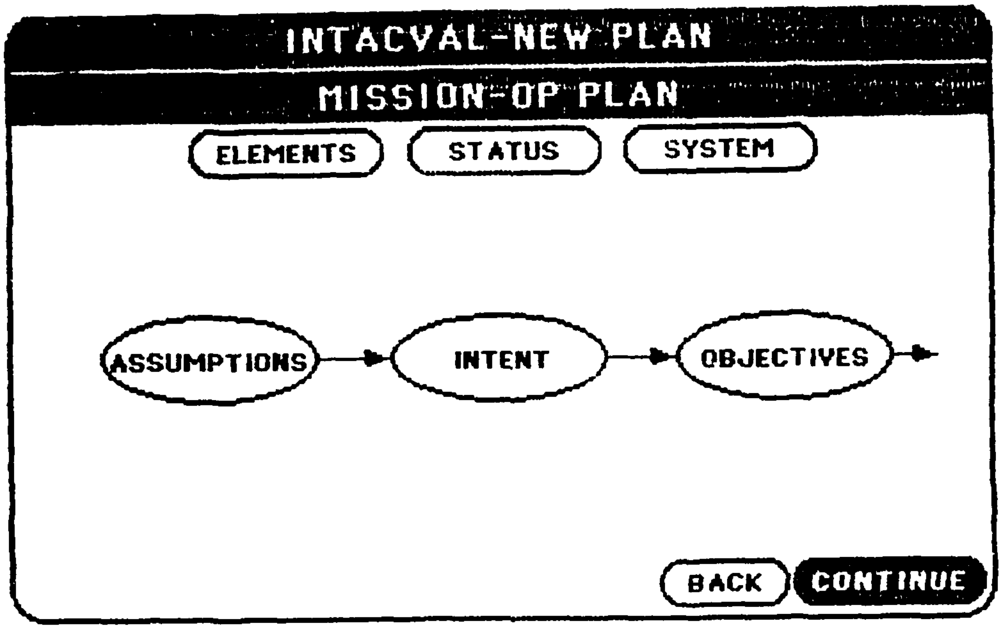


This screen presents the first "process model" for the planner to inspect. Note that the planner does not have to see this model. Experienced planners will not want the system to keep track of their problem-solving processes; inexperienced ones may well like the navigational convenience that process models provide



The process model suggests where the planner is in the process

FIGURE 1.1: Illustrative Process Model



A sub-process model for "Mission" appears

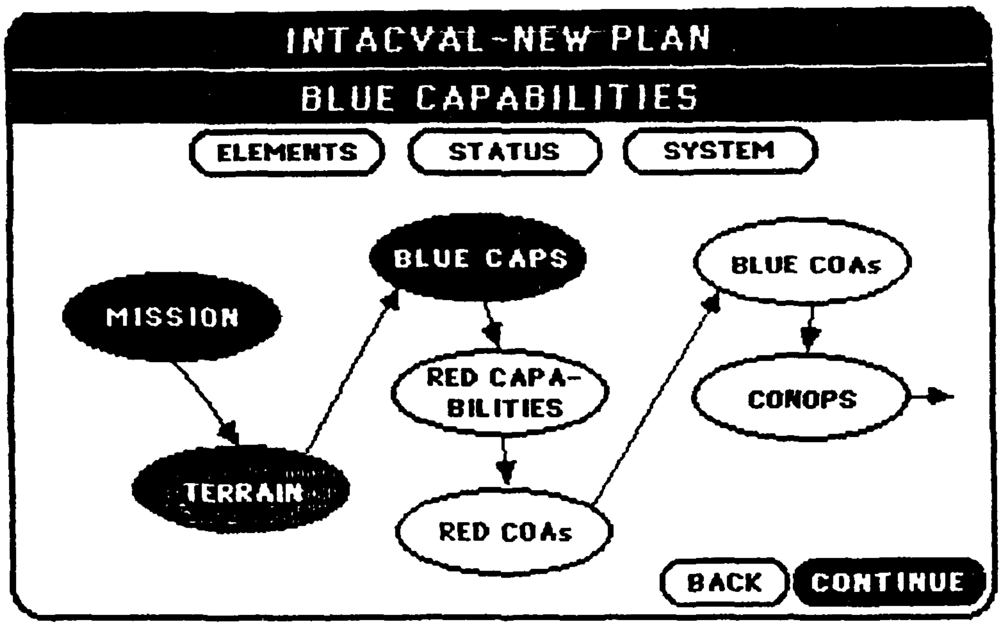


FIGURE 1.2: Illustrative Process Model

is in the process he will be guided by the appropriate level of the hierarchy (while at the same time having access to the entire hierarchy for on-line or instructional assistance).

We also explored the potential of making the models intelligent. Implicit in every process is order. There are optimal ways of solving problems and there are sub-optimal ones. In the domain of tactical planning, for example, it makes little sense to assess adversary courses of action (COA) until one has assessed terrain and adversary capabilities. If a planner jumped ahead in the process and began making assessments of adversary COA likelihoods before looking at terrain or capabilities the system might well inform the user that he has deviated from doctrine.

Artificial intelligence (AI) can help make process models intelligent. It is possible to endow the process models with knowledge about order and sequence and use that knowledge to guide and teach the user.

With regard to analogy-based graphic explanation and guidance capabilities, we explored (with Klein Associates, Inc. personnel) some new approaches to system self-explanation. Too often the results of computer-based problem-solving -- the system "output" -- is inexplicable to users. Users frequently complain about meaningless displays or displays that were designed for the convenience of the programmer, not the user. We developed some analogy-based explanation and guidance capabilities for the planning system. These explanations are in a form that is consistent with the training that tactical planners receive; it is also appropriate to the rank of the planner and the echelon at which he is working.

The concept of analogy-based reasoning is akin to the approach described in this report. We developed a set of "scripts" that can be called up and displayed to users with questions about the nature or doctrinal integrity of specific output. For example, if a planner queried the system to explain why a particular COA was considered more likely or valuable than another, the system might first respond with an explanation of the COA in question by presenting the planning factors that led the system to generate a likelihood or value. If that explanation failed then it would proceed to a related example and so on until the user was satisfied with the COA. This approach requires the system to be able to identify analogous plans from a library of plans with an existing one and then display it to the user using the same kinds of map-based graphics he is used to seeing.

Creating a library of analogous plans proved do-able since the domain was constrained to a specific geographic area -- like Western Europe. We worked with a specific scenario -- the Army War College Letort Scenario (unclassified) over the past few years. Letort was specific enough to permit us to identify a set of analogous plans that can be used to explain output to a planner (or even suggest alternative plans to the planner). We stayed with the Letort Scenario and developed a small library of graphic (map-based) examples of alternative plans that can be used as explanations or training devices.

New hardware and software configurations have made the integration of graphic navigational aids cost-effective. We developed a set of icon-based options that permitted system execution (including the execution of the functions described above). In fact, new software systems like those resident on the Apple Macintosh and its emulators permit the design of pop-up and pull-down menus, as well as continually-present options, without great investments in programming. It is now possible to design input and output routines that are icon-executed, as our storyboard suggests (see Section 4.0).

We developed a set of input and output routines that depend upon graphic (and alphanumeric) icons. We developed a set that is pop-up/pull down, that is, non-stationary, as well as a set that continually appears on the screen, and then tested the alternative designs (see Section 5.4.1).

1.2.3 The Storyboard Prototype - In order to test designs and system concepts it is necessary to incarnate them in software. We developed an interactive storyboard of the displays that represent the functions tha an actual system would perform. "Storyboards" are screen displays that when taken together simulate how a system will function when it is actually programmed. Storyboarding has become an accepted part of the prototyping process (Boar, 1984; Andriole, 1988). It is a powerful technique for validating requirements and testing advanced systems concepts.

IIS has developed an in-house storyboarding capability that involves the use of Apple Macintoshes and some software developed for interactive storyboarding, notably Slide Show Magician (TM)* and Storyboarder (TM).** These tools permit us to design displays and then string them together in exactly the way they would appear in the actual system.

The value of storyboards can be seen in their try-before-buy nature, in the insight they provide for requirements validation, and for their contribution to structured evaluation (see Section 5.1). Moreover, they represent a tangible demonstration of advanced systems concepts early in a research project.

We first developed a set of paper storyboards and then converted them to the Macintosh for demonstration and evaluation (see Section 4.0).

1.2.4 Testing and Evaluation - We implemented a structured evaluation of the techniques via the use of multi-attribute utility (MAU) assessment techniques (Edwards and Newman, 1982).

---

*Slide Show Magician is a trademark of Magnum Software, Inc.

**Storyboarder is a trademark of American Intelliware, Inc.

Performance criteria were established and the systems concept evaluated vis-a-vis the criteria. IIS uses applications software to conduct MAU assessments quickly and inexpensively. We used expert planners to judge the enhancements to human-computer performance that the new designs will hypothetically yield. We also used human factors engineering experts.

2.0 USER-COMPUTER INTERFACE (UCI) REQUIREMENTS

2.1 Problem Specific Versus Generic Requirements

"Conventional" treatments of human factors often deal at the generic, abstract level. There are numerous texts devoted to "general principles" of human factors engineering. It is our contention that generic approaches to human factors engineering -- particularly as they pertain to the design and development of enhanced user-computer interfaces -- are inherently flawed. While this is not to suggest that there is little value in the generic approach, it is to suggest that there are clear limits to the generalizability of generic human factors principles to specific problem domains.

This assertion is grounded in our systems design experience which suggests that while generic approaches represent good starting points for subsequent systems design, they must be tailored to the specific requirements at hand. The implications here are profound. On the one hand, they suggest that what the community has taught us about human factors engineering is not always applicable to immediate design problems; on the other hand, they suggest that human factors approaches are perhaps best implemented from the bottom-up and not from the traditional top-down perspective.

This realization caused us to step back somewhat from the general literature on human factors engineering and user-computer interface technology -- at least initially. The approach we took involved first developing a requirements hierarchy for the substantive requirements that any computer-based support system should -- ideally -- fulfill. We then used this substantive hierarchy to identify a set of UCI requirements that we also arranged hierarchically. We began with requirements and not the generic opportunities for enhanced user-computer interaction.

2.2 The Tactical Planning Domain

The domain of Army tactical planning has occupied our research time for a number of years. During that time we have developed insights into the planning process (particularly at the Corps level) that permitted us to design and develop two interactive planning aids (TACPLAN and INTACVAL). As suggested elsewhere in this report, however, TACPLAN and INTACVAL were not oriented toward UCI. We used this requirements experience to derive a new set of planning requirements from which we then derived a set of UCI requirements.

2.2.1 Substantive/Functional Planning Requirements - The substantive planning requirements we identified appear in the hierarchy below (Figure 2.1). Note that these requirements are organized around the elements of tactical planning; they are also arranged hierarchically, though they are not weighted as to their relative importance. The hierarchy contains information about the kinds of data, information and knowledge necessary to "solve"

- R Plan'ng Requirements
  - 1 Mission Statement
    - 2 Military Objectives
      - 3 Specific Objectives
      - 3 Obj Rank-Ordering
  - 1 Area Characteristics
    - 2 Geographic
      - 3 Topographic
      - 3 Hydrographic
      - 3 Climatic/Weather
    - 2 Transportation
    - 2 Telecommunications
  - 1 Combat Capabilities
    - 2 Red Capabilities
      - 3 Strength/Reinforce
      - 3 Composition
      - 3 Location/Disposition
      - 3 Time/Space Factors
      - 3 "Efficiency"
    - 2 Blue Capabilities

- 3 Strength/Reinforce
- 3 Composition
- 3 Location/Disposition
- 3 Time/Space Factors
- 3 "Efficiency"

- 2 Relative Assessments
  - 3 Strengths
    - 4 Red Strengths
    - 4 Blue Strengths
  - 3 Vulnerabilities
    - 4 Red Vulnerabilities
    - 4 Blue Vulnerabilities

- 1 Operational Concepts
  - 2 CORs
    - 3 Objectives
    - 3 Area Assumptions
    - 3 Strawman CORs
      - 4 Suitability
      - 4 Acceptability
      - 4 Success Probability
  - 2 Pertinent Red Caps
    - 3 Red Military Obj
    - 3 Red CORs

- 3 Red Vulnerabilities
- 1 Operations Concept
  - 2 Red Capabilities
    - 3 Operational Caps
    - 3 Distilled Red Caps
  - 2 Blue COAs
    - 3 Advant/Disadvantages
    - 3 "Sensitivity" Analy
    - 3 COA Vulnerabilities
  - 2 COA Selection
    - 3 Alternative COAs
    - 3 Relative COA Compar
    - 3 COA Rank-Ordering
  - 2 COA -> COO
    - 3 Force Alloc & Timing
    - 3 Supporting Operat'ns
      - 4 Logistics Operations
      - 4 Other Operations
    - 3 Command Relations
    - 3 Deployment Summary
    - 3 Employment Summary

FIGURE 2.1: Graphic Substantive Requirements Hierarchy

a tactical planning problem. In fact, as Table 2.1 suggests, many of the requirements can and should be understood as "needs."

2.2.2 User-Computer Interface (UCI) Requirements - The substantive requirements in Table 2.1 permitted us to derive UCI requirements. Figure 2.2 presents the UCI requirements hierarchy, while Table 2.2 describes the entries in the hierarchies again in terms of needs. One aspect of the UCI requirements hierarchy is its comprehensiveness. There are many, many UCI requirements listed in the hierarchy; in one sense, the hierarchy presents too great a challenge, but in another sense it provides us a working compass toward enhanced UCI, regardless of how difficult the challenge might be.

The key point about the UCI hierarchy -- the requirements essence of this project -- is that it identifies a set of display, dialogue, and interaction requirements that together represent an integrated approach to UCI design. The requirements in Figure 2.2 include some expected display requirements (for such things as terrain and mobility corridors) as well as some unexpected requirements, such as the entire module of the hierarchy called "interpretive" and "interaction" displays. The entries on these levels of the hierarchy are unconventional at best, and purely speculative at worst. Suffice it to say here that they are derived directly from the substantive requirements tempered by our understanding of the domain and our experience working with tactical planners. The UCI requirements hierarchy also suggested

A> Plan'ng Requirements	List of Functional Planning Requirements
1> Mission Statement	Requirement to Understand Mission
2> Military Objectives	Requirement to Understand Military Objectives
3> Specific Objectives	Requirement to Understand Specific Objectives
Obj Rank-Ordering	Requirement to Understand Rank-Ordering of Objs
1> Area Characteristics	Requirement to Understand Area
2> Geographic	Need to Understand Geographic Features
3> Topographic	Topographic Information Requirements
Hydrographic	Hydrographic Information Requirements
Climatic/Weather	Climatic/Weather Information Requirements
2> Transportation	Transportation Information Requirements
Telecommunications	Telecommunications Information Requirements
1> Combat Capabilities	Relative Combat Capabilities Information Req
2> Red Capabilities	Need to Understand Red Combat Capabilities
3> Strength/Reinforce	Overall & Reinforcements Strength Info Requirements
Composition	Need to Understand Red Composition
Location/Disposition	Need to Identify Location & Understand Disposition
Time/Space Factors	Need to Understand Red Time/Space Factors
"Efficiency"	Need to Assess "Efficiency"
2> Blue Capabilities	Need to Understand Blue Combat Capabilities
3> Strength/Reinforce	Need to Assess Blue Strength & Reinforcements
Composition	Need to Understand Blue Composition
Location/Disposition	Need to Identify Blue Location & Understand Dispos
Time/Space Factors	Need to Assess Time/Space Factors
"Efficiency"	Need to Assess Blue "Efficiency"
2> Relative Assessments	Need to Infer "Net" Effects
3> Strengths	Need to Assess Relative Strengths
4> Red Strengths	Need to Determine Relative Red Strengths
Blue Strengths	Need to Determine Relative Blue Strengths
3> Vulnerabilities	Need to Assess Relative Vulnerabilities
4> Red Vulnerabilities	Need to Determine Red Vulnerabilities
Blue Vulnerabilities	Need to Determine Relative Blue Vulnerabilities
1> Operational Concepts	Need to Formulate Initial COAs
2> COAs	Need to Develop Strawman Courses of Action

3> Objectives	Need to Re-assess Military Objectives
Area Assumptions	Need to Identify Area Assumptions V-a-V COAs
Strawman COAs	Need to Develop Strawman COAs
4> Suitability	Feasibility Vis-a-Vis "Suitability"
Acceptability	Feasibility as to "Acceptability"
Success Probability	Need to Determine Success Probability
2> Pertinent Red Caps	Need to Determine Pertinent Red Caps V-a-V COAs
3> Red Military Obj	Need to Re-Visit Red Military Objectives
Red COAs	Need to Re-Visit Likely Red COAs
Red Vulnerabilities	Need to Re-Visit Red Vulnerabilities
1> Operations Concept	Need to Develop Concept of Operations
2> Red Capabilities	Need to Re-Visit Red Capabilities
3> Operational Caps	Re-Determination of Red Caps
Distilled Red Caps	Distillation of Red Caps Vis-a-Vis Blue COAs
2> Blue COAs	Re-Analysis of Blue Courses of Action
3> Advant/Disadvantages	Determine Advantages & Disadvantages of Each COA
"Sensitivity" Analy	Need for Sensitivity Analy Via Variation of Assump
COA Vulnerabilities	Determine Vulnerabilities of Each Blue COA
2> COA Selection	Need to Analyze & Select Among Alternative COAs
3> Alternative COAs	Re-Visitation of Alternative Blue COAs
Relative COA Compar	Need to Compare & Contrast Alternative COAs
COA Rank-Ordering	Final Rank-Ordering of Blue COAs
2> COA -> COO	Need to Translate COA into Concept of Operations
3> Force Alloc & Timing	Need to Determine Force Allocations & Timing
Supporting Operat'ns	Need to Identify & Describe Supporting Operations
4> Logistics Operations	Need to Determine Logistics Operations Info Req
Other Operations	Need to Identify Other Supporting Operations
3> Command Relations	Need to Determine Command Relations
Deployment Summary	Need to Development Deployment Summary
Employment Summary	Need to Develop Employment Summary (Oper Concept)

TABLE 2.1: Textual Substantive Requirements Hierarchy

R UCI Requirements/HEL

- 1 Area Display Req'mts
  - 2 Mobility Displays
    - 3 OPFOR Mobility
    - 3 Blue
  - 2 Key Terrain Displays
    - 3 M'jor Obstacle Disp
      - 4 River Displays
      - 4 Mountains
      - 4 Cities
      - 4 Swamp Areas
      - 4 Other
    - 3 "Feature" Displays
      - 4 Contours/Relief/Topo
      - 4 M'jor Elevation Disp
      - 4 Man-Made Objts Disp
  - 2 Planning Displays
    - 3 OPFOR
      - 4 Avenues of Approach
      - 4 Assem Areas/Attk Pos

- 4 Maj'r Comm Lines
- 4 Maj'r Supply Pts
- 3 Blue
- 4 Avenues of Approach
- 4 Assem Areas/Attk Pos
- 4 Maj'r Comm Lines
- 4 Maj'r Supply Pts
- 2 Weather Displays
- 2 Other Displays
- 1 OPFOR Displays
- 2 Disposition Displays
- 2 Condition/Strength
- 3 Conventional Force
- 3 Nuclear Forces
- 2 Air Support Displays
- 2 Maj'r Logistics Disp
- 2 COAs Displays
- 1 Blue Displays
- 2 Disposition Displays
- 2 Condition/Strength
- 3 Conventional Forces
- 3 Nuclear Forces
- 2 Air Support Displays

2 Maj'r Logistics Disp

2 COAs Displays

1 Interpretive Disp

2 "Qualitative" Disp

3 Risk Displays

3 Constraints Disp

3 Vulnerability Disp

3 Opportunity Displays

3 Other Qual Displays

2 "Quantitative" Disp

3 Relative OPFOR Caps

3 Relative Blue Caps

2 "Cognitive" Displays

3 Cognitive Consis'ncy

4 Conceptual Equiv

4 Transition Displays

3 Option Generation

4 Analogical Displays

5 Current Analog Disp

5 "Old" Analog Disp

4 Doctrinal Displays

5 Definitional Disp

5 Doctrinal Options

- 1 Interaction Displays
  - 2 Navigational Disp
    - 3 "Fly-Around" Caps
    - 3 "Hold & Wait" Caps
    - 3 Process Model Disp
      - 4 Primary Processes
      - 4 Sub-Process Displays
    - 3 Adaptive Help Disp
      - 4 "Active" Help Disp
      - 4 "Passive" Help Disp
    - 3 Adaptive Training
      - 4 "Active" Training
      - 4 "Passive" Training
  - 2 Manipulation Disp
    - 3 Graphic Equiv Disp
      - 4 Summary Data Disp
      - 4 Explanations
    - 3 Map-Based Displays
      - 4 Overlays
      - 4 Explanations
  - 2 Dialogue Displays
    - 3 Alphanumeric Dialog
    - 3 Graphic Dialog Disp

4 Iconic

4 Other

FIGURE 2.2: Graphic UCI Requirements Hierarchy

A> UCI Requirements/HEL

1> Area Display Req'mts	Display Requirements for General Area of Interest
2> Mobility Displays	Displays of Red & Blue Mobility Corridors
3> OPFOR Mobility	Requirements for OPFOR Mobility Options Displays
Blue	Requirements for Blue Mobility Options Displays
2> Key Terrain Displays	Requirements for Key Terrain Displays
3> M'jor Obstacle Disp	Major Obstacles Displays
4> River Displays	Displays of River Obstacles
Mountains	Dsiplays of Mountain Obstacles
Cities	Displays of Urban Obstacles
Swamp Areas	Displays of Major Swamp Areas
Other	Other Displays of Major Obstacles
3> "Feature" Displays	Key Terrain "Features" Displays
4> Contours/Relief/Topo	Contours/Relief/Topo Displays
M'jor Elevation Disp	Major Elevation Displays
Man-Made Objts Disp	Displays of Man-Made Objects/Features
2> Planning Displays	Displays for General Planning
3> OPFOR	Displays for OPFOR Planning
4> Avenues of Approach	Displays of Possible Avenues of Approach (Red)
Assem Areas/Attk Pos	Displays of Assembly Areas & Attack Positions (A)
Maj'r Comm Lines	Displays of Maj'r Comm Lines (Red)
Maj'r Supply Pts	Displays of Major Supply Points (Red)
3> Blue	Displays for Blue Planning
4> Avenues of Approach	Displays of Possible Avenues of Approach (Blue)
Assem Areas/Attk Pos	Displays of Assembly Areas & Attack Positions (B)
Maj'r Comm Lines	Displays of Major Comm Lines (Blue)
Maj'r Supply Pts	Displays of Major Supply Points (Blue)
2> Weather Displays	Displays on Seasonal/Current Weather
Other Displays	Displays of Other Area Characteristics
1> OPFOR Displays	Displays of OPFOR Characteristics & Capabilities
2> Disposition Displays	Displays of Red Disposition
Condition/Strength	Displays of Condition & Strength of Red
3> Conventional Force	Displays of Conventional Forces & Readiness
Nuclear Forces	Displays of Nuclear Forces & Readiness
2> Air Support Displays	Displays of Red Air Support
Maj'r Logistics Disp	Displays of Major Logistical Capabilities
CORs Displays	Displays of Likely Red Courses of Action (CORs)

1> Blue Displays	Displays of Blue Characteristics & Capabilities
2> Disposition Displays	Displays of Blue Location & Disposition
Condition/Strength	Displays of Blue Condition & Strength
3> Conventional Forces	Displays of Conventional Forces & Readiness
Nuclear Forces	Displays of Nuclear Capabilities & Readiness
2> Air Support Displays	Displays of Blue Air Support Capabilities
Maj'r Logistics Disp	Displays of Major Logistics Capabilities
COAs Displays	Displays of Feasible Blue COAs
1> Interpretive Disp	Displays that Support Interpretation of Substance
2> "Qualitative" Disp	Displays of "Qualitative" Phenomena
3> Risk Displays	Osiplays that Convey Risk
Constraints Disp	Displays that Communicate Operational Constraints
Vulnerability Disp	Displays that Communicate Vulnerabilities (R & B)
Opportunity Displays	Displays that Communicate Opportunities (R & B)
Other Qual Displays	Displays of Other Qualitative Aspects of Situation
2> "Quantitative" Disp	Displays of "Quantitative" Information
3> Relative OPFOR Caps	Displays of Relative OPFOR Combat Capabilities
Relative Blue Caps	Displays of Relative Blue Combat Capabilities
2> "Cognitive" Displays	Displays that Support Specific Cognitive Functions
3> Cognitive Consis'ncy	Displays that Support Doctrinal Models of Planning
4> Conceptual Equiv	Displays that Support Conceptual Equivalence
Transition Displays	Displays that Support Easy Cognitive Transition
3> Option Generation	Displays that Support Option Generation
4> Analogical Displays	Displays that Present Analogical Information
5> Current Analog Disp	Displays of Current Relevant Analogs (Cases)
"Old" Analog Disp	Displays that Present "Old" but Pertinent Cases
4> Doctrinal Displays	Displays that Present Information on Doctrine
5> Definitional Disp	Displays of Current Doctrinal Explanations
Doctrinal Options	Displays that Present Doctrinal Planning Options
1> Interaction Displays	Displays that Support Smoothe User Interaction
2> Navigational Disp	Displays that Support Efficient System Navigation
3> "Fly-Around" Caps	Capability to "Fly Around" System Options & Data
"Hold & Wait" Caps	Capability to "Hold" System or have System "Wait"
Process Model Disp	Displays that Present the Problem-Solving Process
4> Primary Processes	Displays of Primary (Overall) PS Process
Sub-Process Displays	Displays that Present Sub-Process PS Models

3> Adaptive Help Disp	Displays that Present Help
4> "Active" Help Disp	Displays that Present System-Controlled Help
"Passive" Help Disp	Displays that Respond to User Queries for Help
3> Adaptive Training	Displays that Support Adaptive Training
4> "Active" Training	Displays that Support System-Managed Training
"Passive" Training	Displays that Support Training by User Request
2> Manipulation Disp	Displays for Data/Process Manipulations
3> Graphic Equiv Disp	Graphic/Alphanumeric Equivalence Displays
4> Summary Data Disp	Displays of All Data & Information
Explanations	Explanation Displays of System-Generated Options
3> Map-Based Displays	Displays that Support Map Manipulations
4> Overlays	Displays that Permit "Mix & Match" Overlays
Explanations	Displays that Support Graphic/Map-Based Explana'ns
2> Dialogue Displays	Displays that Support Appropriate Dialogue
3> Alphanumeric Dialog	Displays that Support A/N Dialogue Options
Graphic Dialog Disp	Displays that Support Graphic Interaction
4> Iconic	Displays that Support the Use of On-line Icons
Other	Other Displays that Support Graphic Dialogue

TABLE 2.2: Textual UCI Requirements Hierarchy

the direction in which the project should move. It was clear that in order to fulfill the requirements in the hierarchy a hybrid approach would be necessary, an approach that would have to develop UCI "solutions" from the "conventional" literature and the some very "unconventional" interpretations of and creative additions to the literature. The next step in the project thus required us to map the conventional UCI technology terrain before turning to the unconventional. Section 3.0 of this report examines both dimensions of UCI.

3.0 USER-COMPUTER INTERFACE (UCI) TECHNOLOGY FOR ENHANCED PERFORMANCE

This section of the report provides inventories of both conventional and next generation or "unconventional" user-computer interface (UCI) technologies and relates an overall UCI framework to the domain of tactical planning. In addition, a master list of options is developed and a ranked list -- that is, high priority options -- is also generated.

3.1 Conventional User-Computer Interface (UCI) Technology

The overall approach taken here assumes a sequence in which alternative interface (input and output) devices (technologies) and concepts are identified in a systematic and exhaustive fashion. Then, this typology is filtered through an intervening variable cluster comprised of the "universe" of problem solving contexts (in this case, the concern is with military planning at the tactical level). The interaction of these two typing schemes produces the specific UCI configuration for the application at hand. Succinctly stated, then, the analytical approach is: universe of interface alternatives ---> universe of problem solving contexts and domains ---> application. Note that this contrasts with the many off-the-shelf UCI typologies, which generally do not explicitly provide for a domain filter. In our case, we have a specific domain (tactical planning), and we have a set of substantive and UCI requirements guiding our examination of the literature.

Hopple (1988) provides a detailed overview of UCI alternatives for decision support systems (DSSs); the discussion here will draw on this work. The UCI connects the user to the DSS (or planning support system) and all of its components. Bennett (1977) offers a lucid template for the UCI design and development task at a very high level of analysis by distinguishing between and among three core facets of the interface: the action language (what the user can do when he or she communicates with the DSS); the display or presentation language (what the user sees from the system); and the knowledge base (what the user must know in order to interact efficiently and effectively with the system). Action language ranges from the very conventional use of a keyboard or the use of function keys and touch panels to joysticks and voice command. The options in the display language arena are equally extensive (use of a character or line printer, display screens, graphics, color, plotters, audio output, and so forth).

Gaines and Shaw (1986a, 1986b) provide a very abstract and useful tour of the UCI landscape which spans the six generations of human-computer interaction. These "bundles" (comprised of hardware, software, artificial intelligence technology, and UCI research principles and guidelines) consist of three central mechanisms for interfacing the user to a DSS:

- Formal dialogue (which represents the computer, with its structures based on the underlying virtual machine);

- Graphic dialogue (which reflects the world or the domain, where the structure of the interface is a mapping from the physical world [for example, the use of icons, which have inherent meaning to users but represent only a position marker to the computer]);

- Natural language (which represents the person, with its embodiment consequently grounded in the linguistic basis of knowledge representation, communication, and inference).

The computer "prefers" the formal dialogue modality, and very early or first generation systems were designed for computer experts, lacked what we have come to regard as user friendly features, and required a serious user to know one or more programming languages.

Across the generations, there has been a profound evolution in the styles and their relative use in systems. During the era of the first generation (1948-55), the machine dominated and its use required cumbersome interactions and virtually demanded the use of an abstruse formal style of interaction; the person was expected to adapt to the computer. The second generation (1956-63) witnessed quite a few developments in the realm of software and the advent of at least some attention to ergonomic considerations; graphic styles were available (but were very expensive and tended to be restricted to simulators) and natural language capability was limited to output (what the user sees) rather than input (what the user can do). The third generation (1964-71) saw the proliferation of man-machine studies along with the emergence of primitive (keyword-based) natural language dialogue, conventional or state of the art formal interaction mechanisms, and the increased availability of graphic styles.

The fourth generation (1972-79) marked a transition to the avowedly user oriented epoch. Data base access and personal computers both surfaced, expert systems research exploded in scope and quantity, and the first book on human-computer interaction appeared. (There have since been many.) The computer was now viewed as the servant of the user (and, increasingly, the partner?) and rules for the design of UCIs began to accumulate.

The fifth generation, from 1980 to 1987, is the age of the expert system. Among the hallmarks of the dialogue modalities during this period are low cost graphic interfaces, integrated formal dialogue systems, and fairly sophisticated natural language systems to link users to DSSs. Ease of use and user friendly became shibboleths as Xerox Star, the IBM PC, and Apple Macintosh began to show up in the trade literature -- and in user's offices and homes.

The sixth generation spans 1988 to 1993. The human and the computer are expected to become genuine partners; artificial intelligence (AI) and human-computer interaction will converge. Computers with "intelligence" (capable of recognizing situations in a more intuitive manner, using inductive inference skills, and actually learning) are anticipated. At the same time, even better UCIs can be envisioned.

Formal language, natural language, and graphics constitute a tripartite scheme for UCI design alternatives (including, of course, various combinations of the above). But there are multiple specific UCI design alternatives. Mechanisms for linking the user to the computer include physical devices (keyboards), actions taken with the devices (keystrokes), computer programs and outputs (visual/auditory information), and dialogue or interaction styles (command languages, menus, question and answer formats, natural language, direct manipulation, etc.). MacLean (1986) catalogues a number of the new interface devices; his list includes the selection menu, trackballs, high resolution graphics, voice activation, aural prompting, cursor control keys, fixed function keys, scrolling, windowing, the use of spreadsheets, the mouse, user assignable function keys, and the touch screen.

Which particular UCI technique or device should be selected for the action and presentation languages of a decision support system (recognizing, of course, that interface mechanisms are frequently comprised of various combinations of the dialogue styles and devices delineated above)? There are two complementary answers to this fundamental question.

First, UCI design guidelines have increasingly become available, and there are hundreds of such general principles and detailed specifications. Smith and Mosier (1984), for example, present 679 guidelines for designing UCI software (for information systems generally rather than DSSs per se) in the functional areas of:

- Data entry;
- Data display;
- Sequence control;
- User guidance;
- Data transmission;
- Data protection.

MacLean (1986), Landee-Thompson (1986), and many others offer catalogues of basic standards for UCI design and development. The major principles include:

1. For displays and controls, keep all displays immediately understandable, ensure that the user always feels in control, and provide a capability for navigating through the DSS without getting lost;

2. For the user-system dialogue component, minimize the complexity of the user entry tasks as well as the probability of entry errors; it is often advisable to employ cursor position entry selection menus to avoid the need to type. (This kind of feature can be particularly valuable for DSSs targeted at high level executives, who are unfamiliar with typing and often regard extensive keyboard entry procedures as menial.);

3. Maintain the consistency of displays and dialogue throughout the DSS;

4. Users may often be interrupted; for reentering purposes, provide for the preservation of the work that was interrupted and guarantee user friendly reentry;

5. Alternative entry methods should be available (such as, the ability to define macrocommands);

6. Error routines should be carefully identified;

7. Ad hoc and canned report capabilities should be provided (along with a library of standard reports available to the user);

8. On-line report display (including scrolling and paging capabilities) is vital;

9. An on-line report help facility should be available to facilitate understanding reports (for example, clear explanations of the type and source of data in reports);

10. For managers, a graphics display capability is key -- as well as the capability to transform tabular data into charts and graphs;

11. Data review and modification facilities must be well structured;

12. Accurate, efficient data base management procedures are necessary for bulk data maintenance tasks (including entry and update capabilities);

13. For external data capturing, data communications capabilities must be built into the DSS; and

14. On-line data maintenance capabilities are also required (including a data entry form capability and an ability to list transactions for verification purposes).

Second, since lists of design principles tend to be generic (although some are linked to distinct classes of special circumstances), it is also necessary to specify the needs of potential users (a part of the requirements analysis stage of the DSS design/development cycle, the subject of Chapter 3 of Hopple [1988]). The type of user(s), task(s), and decision situation(s) covered must drive the overall UCI development process. Senior commanders will require and prefer very different UCI techniques compared to lower level users.

The user driven character of decision support is thus encountered again. In addition, the value added perspective intrudes here. Often, a distinction is made between passive understanding of a DSS and the user's active understanding of the system (Benbasat, 1984, for example). Passive understanding refers to the ease of use/user friendly criterion or the mechanics of system use (that is, operation of the terminal, input and output procedures, the syntax of the dialogue employed).

Active understanding is a more demanding evaluative standard. The active form of understanding taps the DSS's capabilities as a decision aid and indexes those characteristics of the interface that genuinely (and measurably) augment a user's decision making capabilities. Relatively few such analyses are available.

The point is that both forms of understanding enhancement are necessary to a DSS UCI. Furthermore, a DSS could be very user friendly on the basis of user preferences (and perhaps even actual use) without contributing to the decision performance aspect at all. Conceivably, a DSS could also improve the effectiveness of decision making (especially if the unaided procedures and routines are unusually biased or flawed) but be viewed very unfavorably by users. In evaluating a DSS, as will become clear later, it is vital to assess both the user driven facet and the DSS-decision making process nexus.

An exhaustive excursion into the many types of UCI input and output devices would turn into a lengthy volume and many such overviews are readily available (for example, Andriole and Hopple, 1987; Galitz, 1983, 1984; Shneiderman, 1987). Here, the emphasis is on illustrating the potential range and variety of DSS/UCI alternatives, and three specfic options will be profiled:

a menu-based interface; recent work on the use of graphics technology to link users to DSS inputs and outputs; and the increasingly popular natural language choice.

3.1.1 The ZOG Menu-Based Interface - Almost a decade of research has accrued on ZOG, a generalized UCI system based on the concept of menu selection and anchored in an extensive data base of menus and the idea of rapid response to selections (McCracken and Akscyn, 1984). ZOG integrates all of the computer functions that the user will need.

The basic unit of representation in ZOG is a frame. A frame is essentially equivalent to everything a user could see at once on the terminal screen; high resolution screens now permit several such screens to be shown at once. A ZOG data base may contain tens of thousands of interconnected frames.

The user can interact with ZOG in three different ways: navigation; invoking programs; and editing. Navigation is the default interaction mode, where the user makes a selection via the keyboard (or pointing device/mouse) and the system moves on to show the next frame. Some selections lead to the running of a program. The user can also enter the frame editor at any time and make changes to the frame.

ZOG development dates back to the early 1970s at Carnegie Mellon; it was initially applied in the real world as the UCI foundation for a computer-assisted management system for the Navy's newest nuclear-powered aircraft carrier, the U.S.S. Carl Vinson. This effort, launched in 1980, yielded a system which provided a distributed data base of over 20,000 frames and over 30 agents (application programs). The ZOG system reflects a set of general UCI principles (that there should be a homogeneous UCI environment, the tool should be under the user's total control, there should be no dangerous, irreversible actions, and so forth).

The philosophy underlying ZOG also embodies principles related to the system's data base, user interaction, and functional extension elements. The data base architecture must be capable of accommodating hundreds of thousands of frames -- without adversely affecting the system's responsiveness -- and simultaneous use by many different users. The data base should have a network structure in which data items can be linked to other items; specifically, tree structures should be preferred. The menu UCI style should be ubiquitous; the data base should consist of nothing but menus. (As this aspect of ZOG demonstrates, it is possible to apply a data base management approach to the UCI, a feature which has appeal from the perspective of an integrated architecture for the DBMS and UCI -- and, potentially, model base aspect -- components of a DSS.

Furthermore, this point underlines the fact that the separate components of the architecture of a DSS can be and are interrelated; they were trichotomized in Hopple [1988] for purposes of facilitating the exposition of each.)

ZOG also assumes a style of user-system interaction, one in which almost all UCI involves making selections from the menus. Very fast reponse is emphasized (typically, response well under 1s). Also, there should be no hidden selections (no concealed keyboard commands the user is required to remember). The editor is always available as a common command selection.

Finally, there is a requirement for a mechanism to extend the system to provide new functions for the user. The first step in adding a new application to the system is to map the data structures involved into frame formats and interconnection structures within the data base. Programs needed to implement new functions are embedded within ZOG -- available via active menu selections.

3.1.2 Graphic Aids for Enhanced User-System Interaction - Many graphics-based UCIs are available. The example here illustrates an experimental effort designed to accelerate the transition from "conventional" to enhanced human factors approaches to UCI front-ends to decision support (Andiole and Hopple, 1987). The research is designed to enhance the processes by which plans are developed and evaluated in a simulated interactive problem solving system. The enhancements include:

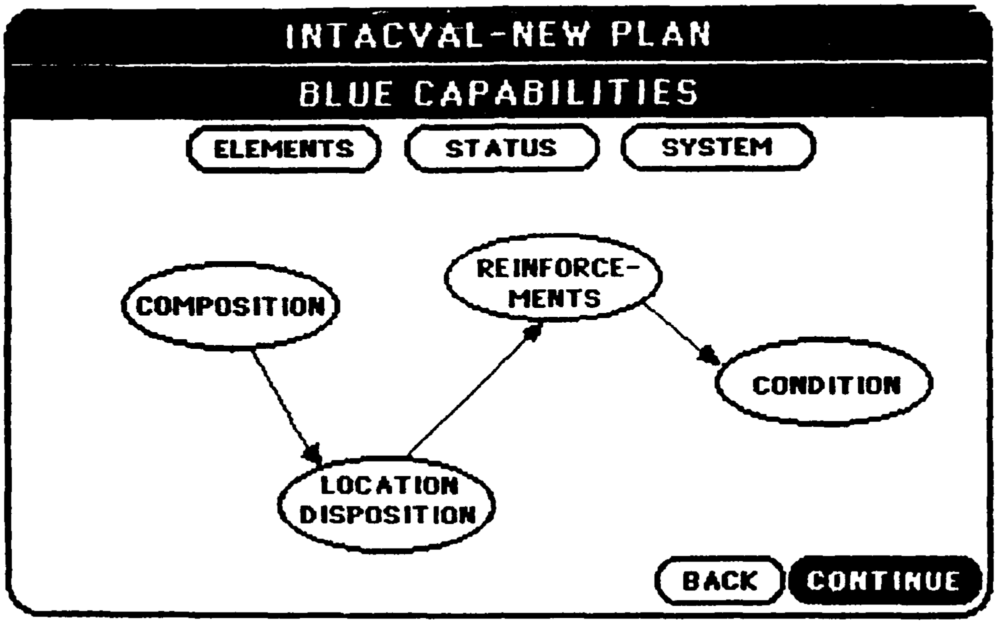
- Embedded process modeling for system status monitoring;
- Analogy-based graphic explanation facilities; and

• Graphic navigational aids.

All three ideas support enhanced UCI tools through training and via on-line system operation.

Embedded process modeling for system status monitoring and management is a technique that seeks to provide users with a top down view of the functions, tasks, and subtasks that they can perform with the system. Too frequently with a DSS, it is virtually impossible to "see" the overall structure of the problem solving process that the system is intended to support. Embedded models serve as on-line compasses, making it very difficult for users to get lost.

The word processing example may again be useful for illuminating this idea. As users process words with the typical program, there are no clues provided regarding where the user is within the larger process. Ideally, it should be possible for novice users to "see" that they are now entering data into the system -- which presumes that they have opened and named a file and that they will close and print at some point in the future. These steps in the process might be communicated to the user graphically or simply alphanumerically. Mechanics using interactive systems to repair automobiles would also benefit from a top down view of the repair process, just as data base managers would find it helpful to see why certain data must be stored, retrieved, and displayed.



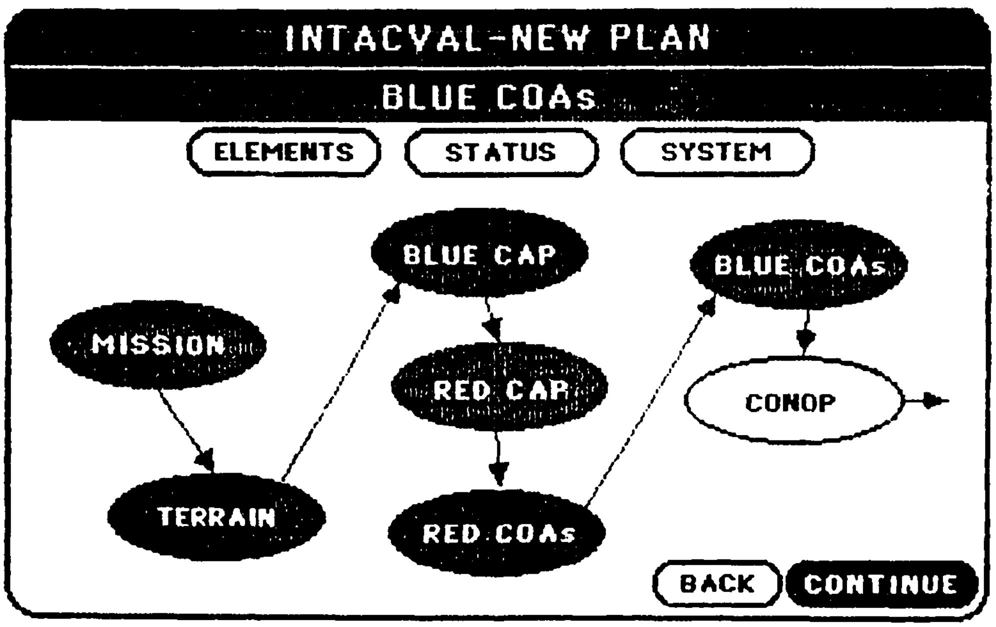
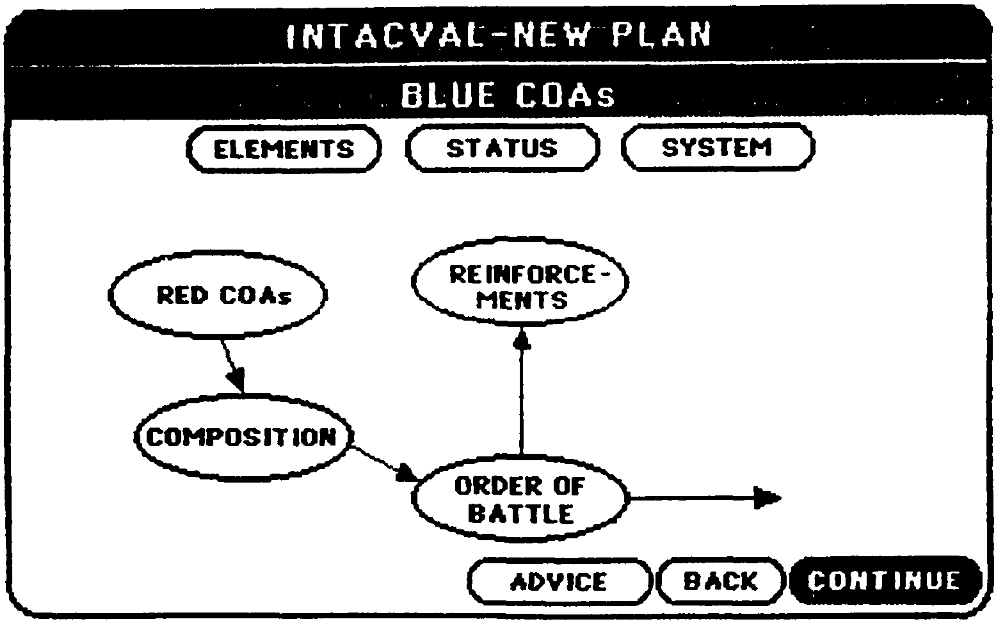


FIGURE 3.1: Illustrative Process Model



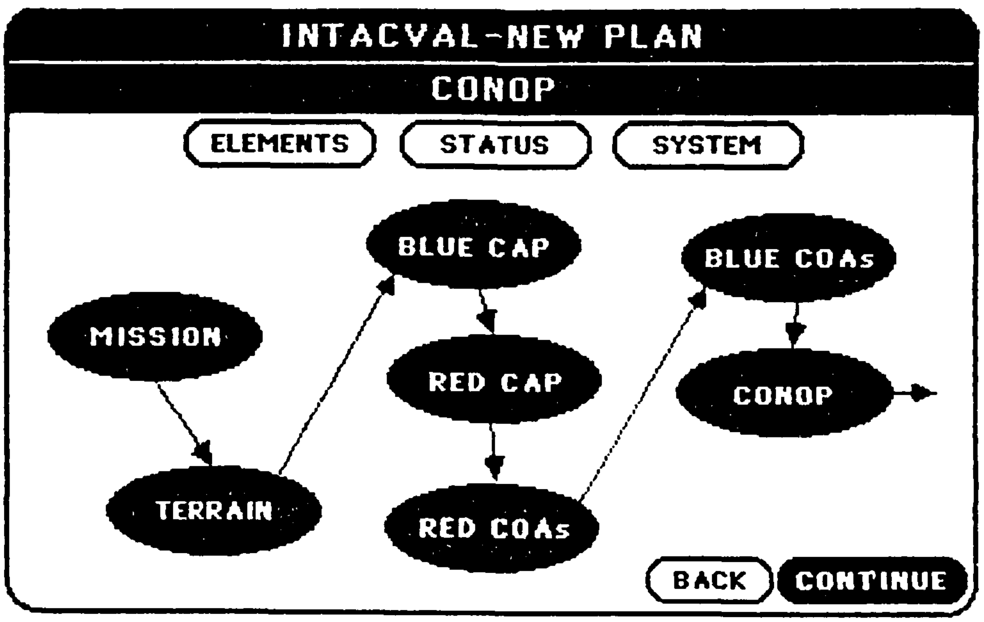


FIGURE 3.2: Illustrative Process Model

How might such process models be used as graphic UCI devices? First, they should be conceived as on-line compasses capable of communicating direction and purpose to users. When designed properly, embedded process models can keep track of the problem solving process and report to users regarding where they are in the process, where they have been, and what they have left to do. Embedded process models can also be used to accelerate the training process, since each step in the process can be organized as a "lesson."

All of this can be illustrated in some simple process models used as a UCI structure in a tactical (military) planning DSS (Figure 3.1). Note that the process model contains information about the steps that planners must take to develop a Concept of Operations. As the planner completes successive steps, the process model updates itself. A quick glance at the model at any point in the planning process would reveal to the planner exactly where he/she was in the process and what was left to do. A planner who was interrupted could return to see exactly where he was in the process.

With respect to analogy-based graphic explanation facilities, another graphic/UCI enhancement candidate, new approaches to system self-explanation are worth exploring. Too often, the results of DSS use -- the system "output" -- are offered in a presentation language which is inexplicable.

The concept of analogy-based reasoning can be based on a set of "scripts" that can be called up and displayed to users. For example, if a planner queried the system to explain why a particular defensive plan was considered more likely or valuable than another, the system might initially respond with an explanation of the plan in question by presenting the planning drivers that led the system to generate the relevant likelihood or value. If that explanation failed to satisfy the user, then the system would proceed to a related example, requiring the DSS to be capable of identifying analogous plans from a library of plans. The basis for the analogy (that is, the similarity in causal factors, and the like) could be displayed to the user. This assumes extensive progress in research on the formation of valid analogies, but also offers a potentially viable UCI explanation facility.

New hardware and software configurations have made the integration of graphic navigational aids cost effective. A set of icon-based options that will permit system execution (including the execution of the functions above) can be conceived. In fact, new systems like the Apple Macintosh and its emulators permit the design of pop up and pull down menus, as well as continuously displayed options, without requiring significant investments in programming. It is now feasible to design input and output routines that are icon-executed.

On graphics interface technology generally, see Foley and Van Dam (1982), the "classic" source which discusses hardware, software data structures, mathematical manipulation graphics, the user interface, and the fundamental implementation algorithms. A comprehensive treatment of color theory is also included.

Preece (1983) features a useful case study of the utility of graphed data in DSSs, noting at the outset that the task of interpreting graphed data constitutes a complex interpretation function. Domain variables, graph concepts, the number of graphs involved, the number and grouping of curves, and other elements enter into this.

Senach (1983) looks at computer-aided problem solving via the graphical display of information in an exploration of the question about whether the display of information in such a format can actually lead to (induce) an inaccurate cognitive representation of the task or inappropriate problem structuring techniques. He concludes that badly designed information displays can produce poor mental representations of systems in users (including the analytically dangerous result of inappropriate problem space reduction). Studies of this genre emphasize the critical nature of the need to carefully design and rigorously test graphic interfaces.

3.1.3 A Natural Language Interface System - Menu-controlled and graphics-based interfaces are common in DSSs; natural language interfaces, once deemed to be exotic and rare, are becoming increasingly available, and this interface style can be expected to proliferate as a modality for users of decision support (and other forms of problem solving) software in the future. The case study of a natural language interface presented here is drawn from Simmons (1986).

The Apple Macintosh is an excellent example of a hardware base which integrates graphics (icons) with a one button mouse and a hierarchy of menus. The Macintosh protects the user from the "danger" of asking impossible questions -- only meaningful sequences of menu choices can be employed. The operating system, which may be controlled via a mouse and menu or by formal language commands (that is, via the utilization of two of the most popular interface styles), provides a set of operations that users can combine to accomplish a set of information processing/ decision support goals. These goals can be used -- with care -- to tell the system appropriate things about user intentions.

In menu-driven interfaces, possible DSS goals must be completely anticipated by the designer (and are therefore embedded or prespecified). In formal language interface systems, the user generally has more latitude to achieve his or her system use goals, but at the cost of learning several formal command languages at the operating system and production program levels. Natural language emerges as a potentially viable and more robust and sophisticated interface for handling the full complexity of user intentions.

Natural language dialogues can be used as flexible mechanisms for communicating with a computer-based system. For example, one "microworld" developed at Stanford Research Institute features an air compressor, an apprentice, and a robot that understands a subset of English concerning the system and includes models of how it is assembled. The goals and intentions (of both the apprentice and the expert) drive the programs.

One system based on the above configuration is concerned with multiple agent planning and problem solving (a domain of obvious relevance to DSSs focused on military integrative planning and decision processes). The robot in the system has a model of what it knows that the apprentice knows (a very important knowledge base from the perspective of training or instruction). The system is endowed with the ability to plan both the action and the communicative acts required to accomplish it.

Compared to other interface styles, natural language must not only guide the user -- it must also predict and supply information that the user wants (despite the fact that the question does not specify the user's intention). On occasion, natural language systems must correct user misconceptions and provide answers to questions that were not posed. Theories of speech acts and intentions are central to any effort to capture the process of human dialogue. UCI in natural language is already available in restricted DSS contexts.

Zoeppritz (1983) offers a very useful theoretical account of the human factors of a natural language system in the context of a

case study of a natural language interface to a relational data base system. She marshalls the array of pro and con arguments on natural language, a continuum which ranges from the position that natural language is the ultimate nonprocedural language to the hypothesis that users may not even be aware of the subtle semantics of question asking.

Pro arguments include that people already know natural language and it is therefore easy to "learn" and natural to use. Ideas can be expressed directly to the computer in the form in which they occur. Even the inexperienced user can express complex facts. The use of a natural language interface presumably reduces or eliminates the psychological barrier to the use of computers.

Among the con arguments is the fact that any natural language system must inevitably have a restricted vocabulary and syntax. Further, natural language front ends (such as, to data bases) may very well yield little benefit at great cost. English may be too ambiguous a language to use in such an interface. Natural (person to person) communication cumulates multiple small errors; such errors may cause complete computer-human communication failure. Finally, natural language interfaces are potentially problematic because they may lead to queries to obtain information which is not in the data base.

Overall, there are both good and bad human factors associated with natural language. Empirical research demonstrates that natural language does <u>not</u> require more typing; inputs via natural language are often shorter in length. Restrictions imposed by the technology are generally acceptable (but there are large differences in the associated ease of learning the system).

Ishikawa et al. (1987) provide an excellent example of the state of the art here. They portray a knowledge-based natural language interface to a data base system. The overall approach is based on the increasingly popular object-oriented features of LOOPS. Thus far, applications have been made to real estate, medical tests, and merchandising; plans are in the works for extending the approach from data base queries to data base updates, decision support, and use of the technology in expert systems. Comparisons are made to such current natural language systems as IRIS, FRED, and TEAM.

3.1.4 Hybrid Approaches to Enhanced UCI - The formal, graphics-based, and natural language interface styles emerge as the central aspects of UCI. If these three dimensions are cross-classified with the potential computer system functions or tasks delineated in Smith and Mosier (1984), the framework or matrix depicted in Table 3.1 emerges. This framework draws not only on Smith and Mosier; the work of Gaines and Shaw (1986a, 1986b) is also very relevant.

Several comments should be made about the tasks listed in the chart -- especially in terms of their relevance to the planning domain. First, data entry focuses on the full range of alternative ways to input data (point at a display, enter numbers, letters, or text, key information into forms/tables) and computer-data processing aids (help, error checks). This is a very important function, but it is relatively unimportant for this application for two reasons. First, tactical planning focuses more on the use (display) of information and a variety of "higher level" cognitive activities (hypothesis generation and analysis, for example). Second, in the prototype, the assumption is that all input data emanate from G2 (intelligence) and data entry is simply taken as a given and is simulated.

Second, data display features such alternatives as text, forms, tables, graphics, and combinations of the above. The human factors issue of density is a core concern here. From the vantage point of plan generation and evaluation (and replanning), the question of how to present information -- the focus of this second task realm -- is absolutely critical. In this area technology advances often drive developments; current examples include the increased use of graphics and moving pictures.

Third, there is the task of sequence control. The primary consideration here is that the user must always feel in control; this makes user attitudes and confidence more positive and leads to higher acceptance (and use) of computer-based systems.

Task requirements and user skills should interactively determine the type of dialogue(s) employed. For instance, routine data entry suggests computer-initiated question-and-answer interaction. The eight primary types (and their associated training requirements and speeds of response) are:

TYPE	REQUIRED USER TRAINING	SPEED OF SYSTEM RESPONSE
1. Question & Answer	Little/None	Moderate
2. Form filling	Moderate/Little	Slow
3. Menu selection	Little/None	Very fast
4. Function keys	Moderate	Very fast
5. Command language	High	Moderate/Slow
6. Query language	High/Moderate	Moderate
7. Natural language	Moderate	Fast
8. Interactive graphics	High	Very fast

A key generic guideline in this realm concerns the need to guarantee maximum possible user control of the on-line transaction sequence (that is, the ability to go to any task/transaction needed or desired -- at any time). (This of course has pitfalls, requiring that the designers anticipate user errors and ensure that damaging actions are made very difficult.) The desiderata of maximum feasible user control is completely in line with the nature of this user group (higher level planners and decision makers) and domain (planning dictates that a support system provide for the capability of implementing the transaction sequence very flexibly).

Fourth is user guidance, which includes error messages, alarms, prompts, and labels as well as the all important formal instructional material about a system (off- or preferably on-line) -- all designed to achieve the objectives of promoting efficient use of the system (the quick and accurate use of the system's full complement of capabilities), with minimal memory load on the user and therefore limited time required to learn to use the system. Flexibility for supporting users of different skill levels should also be accommodated. High quality user guidance features presumably lead to faster task performance, fewer errors, and increased user satisfaction.

On-line instruction and on-line documentation are both desirable subelements of this task category. Also pertinent are status information (a subfunction achieved to a great extent by the previously profiled embedded/graphic process modeling module), routine feedback, error feedback, job aids, and user records. Job aids are particularly noteworthy and range from logical menu structures and hierarchical menus to on-line system guidance, help of various kinds (generic help, task oriented help, multilevel help, browsing help), on-line training (via the simulation of "hands on" experience), flexible training (by user type and level of experience with the system), and genuinely adaptive training.

Several generalizations can be advanced about this function. First, generally feedback should <u>always</u> be provided; a user input should always lead to a system output and speed of response emerges as a particularly important design criterion. Second, this is perhaps the key functional area in terms of planning support; a planner is in desperate need of sophisticated "help" (for example, via embedded analogy "tutorials"), constant status updates about the plan and the planning process, routine and error feedback, and a variety of job aids.

Next comes the data transmission task area. This subsumes a variety of activities relating to message exchange among users of a system and with external systems. This function involves the use of computers for communications (including words, pictures, and numbers); systems with the primary goal of supporting communication are electronic mail systems. A lot of data transmission is automated and entails no direct user involvement.

Currently, the planning support system does not envision this kind of communications capability. However, this task can be expected to become relevant in the future -- for two reasons. The first is the fact that the use of computers for communications in a dispersed or centrally located group decision support context is attracting considerable attention from DSS designers, users, and theorists. The second consideration is that this kind of planning is a very cooperative endeavor (across echelons and between G2, G3, and others at the corps levels), reinforcing the

Finally, data protection is a major task in almost any DSS. Both data security (very pertinent in the military planning sphere) and error prevention are vital. The security of data must be protected from unauthorized access, destructive user actions, and computer failure. The key here is how to make the system easy to use but difficult to misuse. This task is certainly relevant to this application (but less so given the prototype status of the aid being storyboarded); however, general principles of data security and error prevention should be applied.

The three UCI approaches shown in Table 3.1 and the six distinct computer system functions yield 18 cells. Actually, if the three most salient formal UCI techniques (question and answer, menus, command language) and the subfunctions (such as, data input and data processing aids for data entry) are taken into consideration, there are at least 80 "opportunities" for forging links between a user and a system. Guidelines and principles could be delineated for each cell (the existing literature does this, at least to an extent).

For the planning domain, data (knowledge) display, sequence control (system navigation), and user guidance emerge as the three primary tasks in need of support. All three could be provided for via formal, natural language, and graphics-based tools (or various combinations of these three high level families of UCI techniques). After a discussion of emerging/next gener-

	FORMAL				NL	GRAPHICS
	Q&A	MENUS	CMD LANG	OTHERS		
● DATA ENTRY						
- DATA INPUT						
- DP AIDS						
● DATA DISPLAY						
-TEXT						
- FORMS						
- TABLES						
- GRAPHICS						
● SEQUENCE CONTROL						
- START						
- INTERRUPT						
-STOP						
● USER GUIDANCE						
- STATUS INFO						
- ROUTINE FEEDBACK						
- ERROR FEEDBACK						
- JOB AIDS						
● DATA TRANSMISSION						
● DATA PROTECTION						
- DATA SECURITY						
- ERROR PREVENTION						

TABLE 3.1: The General UCI Analytical/Applications Framework

ation technology options, we shall return to this issue and identify high priority options for domain-specific UCI candidates.

Which UCI technique or style (if any) should be favored? Is there a preferred candidate? The answer to these queries is that it depends -- on the decision context, the substantive problem, the users and tasks involved. Generally, natural language interfaces consume much more time and effort than their cousins to design and implement and require extensive (often, excessive) amounts of computer power (especially on today's state of the art micro systems); there are also many who maintain that natural language interfaces are neither necessary nor particularly desirable. The popularity of the Macintosh with its unique interface demonstrates that natural language is not a sine qua non for the achievement of usable, useful human-machine interfaces. (Natural language may, however, become more popular -- and more realistically available -- as tomorrow's state of the expectation replaces today's state of the art.)

More generally, there is a critical need for extensive, systematic scientific inquiry on the effectiveness and efficiency (especially from the vantage point of the active understanding criterion previously delineated) of alternative tools (and combinations of tools) and dialogue styles. Benbasat (1984) reviews the many studies of UCI use and effectiveness (from the passive understanding or system usability perspective) and the much smaller corpus of works which deal with decision enhancement or active understanding. For instance, one study suggested that, when the relative advantages of natural language interfaces versus artificial query languages are compared, fewer invalid queries are generated from users with the use of an artificial query language (SEQUEL). Additionally, the experimental design tested for not only the effects of SEQUEL versus natural language input, but also looked at the two in sequence. In this test, the SEQUEL-English sequence outperformed (that is, produced fewer invalid queries) the reverse (English-SEQUEL) pattern.

One set of interface variables that has been studied repeatedly (although, unfortunately, with mixed and decidedly tentative results) is the use of graphics and color. People tend to assume that the use of flashy colors and fancy graphics "must" be beneficial and enhance both efficiency and effectiveness of decision making (as well as maximize the positive reactions of DSS users). Not all of the research has shown these dramatic effects.

A recent study (Benbasat et al., 1986) suggests that the value of color and graphics is related directly to how well the two support the achievement of a solution to the DSS task (rather than the plausible but sweeping generalization that color and graphics are invariably positive in impact). They investigate the impact of color-enhanced and graphical information presentation on the multiple dependent variables of decision quality, decision making time, the use of information, and user perceptions (preferences).

Benbasat and his colleagues discover that the presentation mode does affect performance and the perceived value of information. However, these effects operate only in clearly specified contexts. The use of graphics yields benefits limited to a reduction in the time required to make a decision -- but only when the graphic report capability directly aids in solving the task at hand. Multicolor reports produce benefits -- again, in limited conditions.

Additional basic research is thus necessary. This can and should be conducted across a variety of field (real world) and experimental (artificial) research design contexts, as Benbasat (1984) recommends in his useful survey of work on the effects of UCI alternatives. A way to facilitate this kind of vital research (which can also be used in applied system design and development efforts) is the use of a UCI simulator (also referred to as MMI or man-machine interface simulators).

One example is reported in Morgan et al. (1985) -- the Man/Machine Interface Simulation Tool or MMIST. MMIST is a versatile, interactive software system which resides on a VAX 11/780 with a device independent graphics package. DSS designers can use the system to develop and demonstrate detailed interactive displays/menus and processing sequences early in the software design phase of the DSS life cycle.

Three capabilities are at the core of the overall objective of MMIST:

1. A robust display/menu construction tool which permits sufficient detail;

2. A friendly user interface -- usable even by inexperienced users; and

3. Impressive interface simulation capabilities -- by which users can examine and assess menu hierarchies and display formats before the coding of any production software.

MMIST and other UCI simulators -- available in increasingly sophisticated versions -- are fully compatible with (and in fact virtually mandated by) the newer protoyping strategy for developing decision support systems. The idea is to allow users to see and explore and actually use (in an interactive but simulated mode) the proposed UCI for the DSS application; the interface can then be modified as frequently and as extensively as user feedback warrants. The implementation of a storyboarding/rapid prototyping design and development strategy in this effort fits in well with this perspective, and our previous experience on rapidly generated simulations of a pilot system's UCI (such as, Andriole and Hopple, 1987) have already demonstrated the utility and viability of this tack.

There is no paucity of UCI design configurations. Creative syntheses (such as, Clarke's [1986] description of window-icon-mouse-pulldown menu or WIMP techniques) continue to emerge, and icon-based interfaces (GEM, Windows, Desq, and others) are particularly prolific in the marketplace. The case studies presented above, then, illustrate only three particular examples of three generic styles or tools (menu-based, graphic, and natural language); there are many variants and permutations -- and other specific UCI approaches above and beyond these. But the DSS designer should not become lost in the forest of styles, acronyms, and tools; the key consideration is the importance of a consistent and user friendly system interface which functions effectively and efficiently to link the user to the overall system and its data base and model base components.

3.1.5 An Aggregate Assessment of Conventional UCI Technology - What can be concluded about conventional UCI technology and this domain? In any high level problem solving or cognitive task domain, psychological modeling of the users is an indispensable facet of both requirements analysis and system modeling and storyboarding. Shneiderman (1987) presents a cogent and well stated overview of this task in his discussion of the need for a high level theoretical or syntactic/semantic model of user knowledge.

Syntactic knowledge refers to device-dependent details of the system and UCI; semantic knowledge indexes understanding of and familiarity with the central concepts in the domain. Semantic knowledge in turn bifurcates into knowledge about task concepts (objects and actions) and computer concepts (objects and actions). Thus, both the domain and the machine represent sources of knowledge the user must bring to the task environment.

Another way of looking at this appears in the previously cited book on graphics and computers by Foley and Van Dam (1982), who organize the process of psychologically modeling the user into a four level approach. First, there is the user's mental model of the system. Second, the user also perceives and processes information at the semantic level (assigning <u>meanings</u> to inputs to and outputs from the system). Third, there is syntax, the assembly of units (that is, words) into sentences. Semantics and syntax are of course intertwined in reality. Finally, there exists the lexical level, which concerns device (machine, man-machine interface, UCI technologies) dependencies and serves as the substratum for the specification of syntax. Obviously, the lower three levels interactively shape the high level mental model of the overall system.

Related to all of this is the increasingly popular concept of system transparency. Maass (1983) provides a useful introduction to this notion. Why transparency? In the planning arena (as well as in other cognitive problem solving task realms), the computer functions as a communication machine (rather than a simple number cruncher). System transparency is vitally important in this kind of domain so that the user can look inside the "black box" and can easily build up a veridical internal model of the system.

System transparency means that there are <u>natural</u> dialogue conventions. In addition, the user very rarely sees one totally consistent interface -- because of the frequency with which susbsystems are independently designed. But system transparency dictates consistency of the user interface (and all of the human factors/cognitive psychology UCI research places a similar emphasis on consistency). Finally, a system with a coherent self image can give the user explanations about itself and its behavior (for example, explanations of input alternatives and expected formats).

Second, another conclusion that emerges -- based in part on the extensive requirements analysis data summarized earlier in this report and on the cognitive modeling knowledge gained and the principles of system transparency -- is the centrality of graphics in tactical planning support systems. The requirements analysis/cognitive modeling results underline the fact that planning is a non-numeric, symbolic, map-driven, and graphic process. Perhaps because of the central role accorded to maps and map-based displays (along with a cognitive preference for zooming around maps and animation), planners think in very graphic terms. This resulted in the decision early on in the previous Army planning research to go with a dual screen configuration, where one screen depicted the map (with user options for annotating, decluttering, and so forth) and the other showed comparable alphanumeric data.

This in turn led to the use of graphics in a more generalized sense as well. For example, the union of decision analytic and artificial intelligence tools in one effort reached fruition in a knowledge-based decision analytic methodology base where the decision analysis techniques of multiattribute utility (MAU) assessment, influence diagramming, and Bayesian-based hierarchical inference structuring were incarnated in the system and displayed to the user as graphics (Andriole and Hopple, 1987). All of this culminated in the emphasis given to analytical graphics as a relatively new principle for the several generations of our planning work.

Analytical graphics suggests such system architectural notions as embedded (graphic) process models, graphic equivalence (showing the same data or knowledge alphanumerically and graphically), graphic explanation facilities (showing G2 assessments in the form of an influence diagram, for instance), and other such features. This clearly implies that graphics-based UCI technologies rank very high on the agenda.

What about artificial intelligence? This raises the third general conclusion, that natural language systems, knowledge-based systems, and (in general) fifth (and emerging sixth) generation systems will pervade UCIs of the future. Hahn (1983) surveys the potential and current contributions of AI to the human factors of application software. His account traverses across vision, robotics, theorem proving, speech recognition, and natural language processing.

As Hahn sees it, the AI track record is good but far from

spectacular. Working AI systems provide some genuine benefits in a **human factors** sense. Scene analysis systems provide relief from fatiguing observation tasks. Cognitive ergonomics has contributed to data base management support and knowledge acquisition. Obviously, robotics relieves humans from performing tasks that are tedious and sometimes even dangerous. Natural language processing has experienced slow progress; progress has been most visible in terms of knowledge-based machine translation.

More potentially relevant than AI per se is the intersection between AI and psychology -- the nexus between smart computers and machines capable of adapting to and anticipating the actions and even thoughts of users. This marriage (known as cognitive science) is important to the fifth generation and will assume even greater centrality with the advent of the sixth generation. Natural language interfaces (and other manifestations of AI) will become more important to UCI design considerations in the future, and system transparency will take on even more importance.

3.2 Next Generation UCI Technology

System operators can often overcome poorly designed display panels and incompatible display-control relationships during routine system operation. However, when time-pressured decisions and responses must be made, the display designs must be optimal if errors are to be avoided. In the present section, we describe

stress resistant display guidelines related to various UCI **dimensions.** These guidelines were selected to illustrate the power of current high technology hardware and state-of-the-art software.

3.2.1 Organizational Metaphor - The system should employ a metaphor which is consistent with the user's view of its function. Lakoff and Johnson (1981) defined metaphor as the use of an example in one domain to provide structure for a second domain. For example, they explained that there is not a clear concept of what an argument is. But if we say that ARGUMENTS ARE LIKE WAR, then we can use what we know about war to see how our opponent is attacking our position, and we are defending it and searching for chances to counterattack. In a study of designers (Dent, Klein, and Eggleston, 1987), we found that metaphors were employed in virtually all the displays we examined. Metaphor is a potentially powerful source of organization in CRT displays in two ways: to the designer as a source of organization in guiding decisions about how to portray information, and to the user in guiding attention to important information needed for skilled action under time pressure. Metaphor is powerful because it uses what is well known and familiar to comment on or depict what is less well known.

With an increase in the use of pictorial displays comes more opportunity for visual metaphor. Indeed, metaphors are pervasive in designs for interfaces in the areas of word processing and animation (Carroll & Mack, 1985). Organizing metaphors which structure a whole display or set of displays, and visual metaphors which can appear in iconic displays, seem to be important tools on which designers of word-processing software and interfaces can draw.

The resemblance between different domains is the basis of the power of metaphor to guide attention to information about real-world objects and events (Verbrugge and McCarrell, 1977; Verbrugge, 1980; Dent, Klein, and Eggleston, 1987). This resemblance, then, supports the transfer of skilled action known well in one domain to action in another domain. The effective metaphors were ones that guide performance by letting the operator access an integrated response sequence from another domain and use it to react to the new domain. The function of metaphor was to guide actions, not to structure the assessment of states.

When used systematically, metaphors can be very powerful in organizing the designer's task and in organizing the operator's use of the display; examples are the flying-in-formation metaphor and the desktop metaphor.

A display which helps the pilot simulate flying-in-formation has been used to help Naval pilots under heavy bombardment. When a pilot has to fly through thick anti-aircraft irradiation, a cockpit display is critical to show the best flight path. There are several forms that this display could take. The pilot can be shown the irradiating cones over each anti-aircraft installation. Alternatively, he could be shown the safe flight path as a river, or pathway in the sky. The Navy has found that a phantom wing leader display can be used to good advantage. The display shows a single aircraft ahead of the real plane. The pilot is told to fly formation with this "wing leader." In this metaphor, the pilot is told to treat the displayed information as he would a real aircraft. Since he is very familiar with flying in formation, this metaphor is a successful one.

Macintosh computers organize their displays in a desktop metaphor. Their word processing program emulates an electronic desktop and filing system to help the operator organize his/her use of the system. Macintosh developers correctly assumed that the user would already be familiar with nonelectronic desktops and filing systems. Hence, the transfer to a computer version was a helpful metaphor.

The power of metaphor lies in its potential to organize displays and guide the operator's interactions with the displays. Domains that are well known to both designer and user can be used as metaphors to coordinate displays and the actions the displays support. This is why the flying-in-formation metaphor and the word-processing-as-desktop metaphors work so well. A metaphor which is less likely to be successful is "aircraft as a human body." It has been proposed that cockpit displays indicate the "health" of various subsystems within the plane. We believe that this metaphor may not be successful because physiology and medical diagnosis are not well known or often used by designers and pilots. Therefore, the power of this metaphor is severely limited by the user's knowledge in that analogy domain.

Because two different domains are involved in metaphor, mismatches or areas of dissimilarity will exist and so the potential to hide certain information also exists. The challenge is to develop guidelines and support material to maximize the effective use of metaphor by designers and to minimize the risk that metaphor could mislead the operator. The cost of the misuse of metaphor may be high; however, the cost of not using metaphor may also be high. The designs that will be used in cockpits sometime in the next 15 years already include metaphors, but they have not been used as consistently or as completely as possible. In addition, alternative metaphors or alternative designs without metaphor have not been tested against those already in use.

One possible organizational metaphor for the proposed decision support system is that of a staff person who provides background knowledge, cases, and rules. This electronic staff person also keeps track of the session, filing and retrieving plans in progress. This metaphor is a good way to depict the uses of this system. The planner's knowledge of support staff roles will guide his use of the computer support system. The staff person metaphor will be fine for the general organization of the planning session. However, we expect that it will not be helpful in time pressured segments where the planner needs to know what action to take. In these segments, the system must serve a more directive role. This role is analogous to the Commander's Directive. Whether the combinations of such different roles within one system will give the impression of "electronic schizophrenia has not yet been determined."

3.2.2 Decluttering - The user should have the option of selecting the detail and the density of information displayed in a single screen or menu.

Eliminating nonessential elements from a display is known as decluttering. Under time pressure, the operator needs to find the critical cues with the least amount of distraction. Edgell and Castellan (1986) have reported that an overload of relevant cues can produce more interference than does an excess of irrelevant information. Because of the ability to add more and more detail to displays, designers are currently preparing strategies to "declutter" a display by reducing the classes of details. However, this can take time. The operator must take time out to call for decluttering. What is needed is a set of techniques that would produce the decluttering as a function of mission requirements, unless over-ridden by the operator.

Decluttering is a particularly appropriate facility for a graphic or a map display. A full planning map will include too much information to allow the operator to concentrate on the segment being planned. For example, the positioning of supply lines can be considered independently of terrain information. Therefore, terrain features can be omitted during this planning activity. However, the movement of these supply lines must again consider terrain features so the operator should be able to toggle the terrain overlay back onto the screen to plan movements. Other examples include highlighting among menu options, and hierarchical menus.

3.2.3 Task Priorities - The system should help the user determine which tasks must, should, and don't have to be accomplished.

Highlighting (foregrounding) the key aspects of a task helps to set task priorities within a display. Figure/ground relationships can help to emphasize the key aspects of the task. We have found that even under routine operations it can be difficult to maintain a proper sense of priorities, to ensure that the most important task is done first. Under time pressure, this need is even greater. Operators can become so immersed in handling noncritical problems that they fail to react to emergency conditions until it is too late. This was the rationale for alarms, but auditory alarms are not the sole source of alerting available to the designer. There is a need to restructure displays to aid the operator in setting task priorities. (Embedded process models satisfy much of this need.)

For example, during extreme conditions such as spins, the F-14 CRT will degrade into a large arrow showing the pilot which way to pull the yoke. There is a general requirement to help the operator focus on the most critical task first. As a second example, instances of figure/ground reversals should be considered here. Pilots need to see where the radars of anti-aircraft batteries are searching, but when these become too extensive there is a need to shift into a display of safe routes through the anti-aircraft irradiating spotlights.

In addition, the system should provide an ordering within any sequence of subtasks necessary to accomplish a goal. If the planning session is under too much time pressure, some tasks will have to be shed. Which tasks should be delegated or just not done? Which must be completed regardless? Which tasks must be completed early in planning either to meet interim deadlines or to allow completion of subsequent steps? A "to-do" list should be designed to keep track of the remaining critical tasks in the planning session.

The concept of "urgency" is built into the need for setting priorities. If A must be known before B can be decided, then the decision support system should be designed to flag A as a necessary prior component. Whenever the planner deals with troop placement before determining the terrain elements, this error should be flagged. If the planner is dealing with air defense, then weather is the necessary requisite factor.

3.2.4 Functional Prototypes - The display organization should be focused on reactions that the operator must make, rather than on simply producing current status descriptions.

The appropriate orientation is on the person's preparations for action rather than on the passive identification of the cues in a scene. In Klein Associates, Inc. fireground research, standard structural prototypes for fires would include residences, apartment buildings, factories, and the like. We did not find much evidence for the use of these structural prototypes. The fireground commanders were oriented around the orders they would have to give, and the prototypes were things like Search & Rescue operations, interior attacks, defensive operations to contain the fire, and the like. The commonality was not in the structural features but in the functional requirements of the fires. Displays that present functional (response) layouts will be easier to use during time pressure.

An example of the misuse of status descriptions is the displays designed for process controllers in a chemical plant. Each CRT screen portrays a separate chemical tank. Unfortunately, during emergencies the operator needs only a small amount of information from each screen and must flip back and forth between them in order to have the information necessary to achieve a rapid and safe shutdown. Ideally, there would be functional screens for the major operations of shut-down, start up, and damage control.

Bateman, Reising, Herron, and Calhoun (1978) have shown the effectiveness of a multifunctional keyboard. They demonstrated that a tailored display had definite performance gains over the standard logic tree. Bateman et al. used some dedicated switches and some multifunction switches. In some conditions, the multifunction switches changed functions according to a branching logic. Under another condition, there was an automatic assignment of functions and legends to switches according to the flight mode of the aircraft. Significant time savings were realized using the tailored logic, which would be predicted by our guideline of functional prototypes.

This decision support system should be organized to allow the user to plan on the basis of the action to be taken. When the planner considers comparison cases, he will be interested in the action that was taken in that case. For example, in defense planning, an historic precedent should be made available via the defense strategy, not on the basis of the environmental cues (weather, terrain, force strength).

3.2.5 Memory Aid - The system must store progress of the planning session and then allow retrieval of this information.

The system must permit fast, easy access to its memory stores. Our experience with battalion level planning indicated that a considerable amount of time was spent in going over the same planning segment several times. At widely separated points in the five hour session, planners would rehash the same material. A support system which retained these completed planning segments and incorporated them in the ongoing planning would be a useful aid. Entire run throughs of some tested or built course of action could be stored for future use. For example, the bridge can be taken out by an air strike and this will divert the enemy. This plan (with all its attendant subplans) can be stored and later accessed when the plans for the general air strike are being made. The memory of tested "what if" plans can be stored and will allow the planners to be more creative in their contingency planning. We believe that planners will like to use these so they should be made easy to use iteratively to tweak plans.

(In the next two sections, we will discuss Deepening and Predictor Displays. These are actually subsets of the general memory aid question. However, they are sufficiently important to warrant their own guideline entry.)

3.2.6 Deepening - The system should encourage the user to retrieve information about his selected option.

Planners, as well as other decision makers, seldom consider two or more response options concurrently. They will select one option and then explore it to see (a) what subdecisions are needed to carry it out, and (b) whether there are any unsurmountable obstacles to the plan. This exploration is called deepening because the planner delves deeper into the option as he verifies its suitability. Clearly, the quality of the plan will be directly effected if the planner does not deepen the option sufficiently to discover fatal flaws in his decision choice.

Some options require deeper exploration than do others. This depth can be determined by which characteristics/comparisons are relevant to executing the action. The behavior of expert planners can be used to guide the design of the system. The deepening paths and the information which must be sought will be domain specific and can be obtained from subject matter experts. Suppose that the planner selects the option: "Slow the enemy by calling an air strike on the upriver bridge." The need for an air strike should always kick in the guidance checklist: "Check weather conditions"; "Contact fire support officer"; "Set radio communications and code," and the like.

If the operator deepens the solution and does not discover any problems, then he will retain the plan as a sufficient solution. He often will not go on to consider alternative options in a search for a optimal solution; the sufficient one may do. The deepening option of the decision support system can be used to fine tune this plan. Instead of quitting with a set battle plan, the computer system will encourage the planner to play "what if" because the deepening aspect of the program makes it easy to consider alterations of one plan.

3.2.7 <u>Predictor Displays - The system must be able to display the time course of planned actions</u>.

This principle has been used to help control airplanes and ships whose order of control or lag were too great for the pilot to steer. The predictor display shows the operator where his craft will be at some time in the future. This prediction is made on the basis of the current control positions. The prediction gives the pilot enough advanced warning to allow a course correction, if needed.

Groups of men and equipment constitute a system with many of the characteristics of a high order control vehicular system with built in lag. Each component moves at its own estimated speed and with its own characteristic responses to the environment. Clouds affect air strikes, but mud affects ground troop movement. It is hard for the planner to see how the various components of the battle plan will interact and where they will be located at future points in time. The computer support system can be designed to provide this predictor information.

For example, fast forward simulation can be used to determine whether an air strike force can arrive enough before a battalion must move into place. The information needed to simulate these movements is the same as that available now to the planner in his own memory or in written tables. However, new UCI technology allows this information to be stored within the system and accessed automatically.

The planner will be more likely to consider alternatives to his initial plan because of the ease with which the fast forward function can display their consequences. Supplemental displays can be used to show the course of an increase/decrease of a resource or a threat. These effects can be shown as a function of time or with a variation of action plans.

3.2.8 Reconsider - A system should prompt the user with doctrine or historical precedent if he does not access them when the situation warrants it.

Under pressure, many planners fail to consider doctrinal solutions which they have studied during their training. This is not a failure of creative solutions, but a failure of memory. The ability to prompt the planner to consider untapped alternatives requires a true AI system to select responses appropriate to the situations as it has been presented to or altered by the planner.

3.2.9 Case Based Reasoning - A user should be able to access prior instances, in whole of in part, of the scenario he is planning.

Decision makers often use previous experiences to guide them in the solution of a new problem. Very rarely is a situation so unique that there are no previous experiences which will be helpful. Military strategy domains (e.g., battle planning or air defense) are rich in the number of prior cases which can be called up as prior instances to help the planner.

These prior instances or battle plans qualify as cases in the terminology of AI developers who have studied the use of experiences in problem-solving. When a person solves a problem or develops a plan by drawing on information from a previous experience, this is known as case-based reasoning. Case-based reasoning (CBR) is a knowledge representation and control methodology which can assist planners in making complex, domain-specific decisions or problem assessments and recommendations based upon previous experiences and patterns of previous experiences. These previous experiences, or "cases" of domain-specific knowledge and action, are used in comparison with new situations or problems; and these past methods of solution provide expertise for use in those new situations or problems the system is built to handle. Schank and Abelson (1977), Kolodner, Simpson and Sycara-Cyranski (1985), Kolodner and Simpson (1985), and others have examined the applicability of developing automated systems for reasoning based upon previous experience.

Their conclusions have sparked an interest in development of these systems (Kolodner and Riesbeck, 1986; Kolodner and Simpson, 1986; Bain, 1986). While the majority of the efforts to date have been in the academic environment, case-based reasoning is just beginning to emerge as a viable method for providing rich, knowledge-intensive foundations for the production and operation of expert systems. For example, a new initiative, led by DARPA, is under way to develop CBR systems for the Defense Department. CBR can be used as a method for structuring appropriate domain knowledge and processing capabilities to provide experiential reasoning in knowledge-based systems and conventional systems. There are two basic conceptual types of CBR: precedent-based and problem-solving (Rissland & Ashley, 1986). Precedent-based CBR seems to be the more appropriate type for making air defense or C^2 planning decisions. In a precedent-based system, past cases are precedents which are interpreted to provide solutions, analyses, and explanations of present cases.

The system must be able to access appropriate comparison cases as starting points for the generation of analogous plans. The expert planner makes most decisions by comparing the current situation/action to a previous case. Beginning with this case as a base, the expert compares and contrasts the present situation to it and plans accordingly. The major question for the development of a computer decision support system is how these cases will be indexed to allow the user/system to access them in response to the operator's/scenario's input.

3.2.10 Description of Probability - The system must be able to describe uncertainty about conditions and outcomes in a manner that the operator can use productively.

Probabilities are not well understood or utilized by most people.

When the planner learns that the chance of rain is 10%, how does he use that information?

He certainly does not use it in an accurate quantitative manner. Extensive research on people's uses of probabilities has shown that they make systematic distortions of these measures. While low probabilities are distorted more than higher ones, none is used very accurately. Furthermore, the use of probability information is influenced by the way in which such odds information is presented. It matters whether the probability is represented as 10%; 1:10; 10:100; or 2:20. There needs to be a better way to tie down this measure of uncertainty so that it can be used more accurately by planners. Recent work with graphical displays has been promising but it is still in the elementary stages.

In addition to the actual representation of probabilities, the user needs to know the driving factors which are producing this uncertainty. Information about the cause of the uncertainty is of critical importance in the meaningful use of these odds. Is it because of the time of year (almanac) or is it influenced by incoming weather patterns? Perhaps the information has not yet been updated and the meteorologists will be able to firm up this probability in the near future, so the planner should postpone this aspect of the planning until better information is available. The computer system should provide this background information with all probability statements.

3.2.11 <u>Deviations From The Expected - The system should highlight any deviations from doctrine being employed by the planner</u>.

Deviations from doctrine may mean that the planner knows a better way, that this is an exceptional instance, or that he is making an error. In any case, these deviations should be highlighted to call his attention to the fact.

Doctrine also dictates specific acceptable outcomes. It is not acceptable to leave an intact bridge in the expected path of the advancing enemy. However, this may be the outcome of a particular plan, which looks good in all other respects. The smart system should highlight this unacceptable outcome. The planner may have to live with the deviation, but at least he should be aware of it.

3.2.12 <u>Alternative Sources - The planner should be able to get additional information upon request. He should not have to log out of the planning system or leave the room</u>.

C^2 or Air Defense planners are no different from engineers, corporate planners, or any other professional who makes decisions by accumulating information. They will reach as far as their desk, bookcase, file cabinet, or computer for that information <u>and no farther</u>. Studies of engineers show that a colleague down the hall is too far away to ask. Our study of C^2 planners showed that they did not radio their Intelligence Officer (G2) for information. This was not a matter of doctrine, but one of immediacy. There was always something more important happening right in the planning room. Yet when the battle was over (and lost) they said, "I wish we had had better intelligence on where they were coming from."

The same principle of availability holds for the access to information within a computer system. When the user must logout of one system and into another, he is much less likely to access that second system. The decision support system must provide information internally or interface to other databases, without having to logout of the main system itself. Compatibility between computer systems or databases is not achieved by happenstance. The program developers of the computer system and its database must plan for this interface by providing some long range developmental planning in this area. Fortunately, when such planning is done, translation codes can usually be built to make this interface compatible.

3.2.13 Contingencies - The system should keep the planner informed of the big consequences of an action he is planning in terms of necessary service/support resources.

As a plan is developed, available resources are used up. He has only so many engineers and they can only be at so many places at once. If he deploys all of his aircraft in rapid, early sorties, he will be unable to use them for later changes in the battle.

The planner needs a constant tally of the resources he has allocated and the ones which he has remaining.

Some of these are matters of timing, as well as total available resources. For example, do you remember that air strike force can not be here unless you slow the enemy long enough to delay engagement by three hours? This is particularly critical when this delay is atypical and has only been created by some aspect of the developing plan. For instance, the planner is now on his third iteration of this morning assault. He had previously been told that he could not plan on air support. Suddenly he learns that he will have air support available. He then alters his plan to include this cover. This is an ideal time for a "stupid" mistake. The planner does not calculate that the need to maintain radio silence until 0800 will mean that his new air support will do him no good unless he is able to delay the enemy by the proposed three hours.

3.2.14 Conclusions - New technologies provide exciting opportunities for user-computer interface developers. The best of these interfaces will be built by specialists with a thorough appreciation of the users and their tasks. Dumas (1988) described UCI guidelines as a convenient way to communicate the accumulated experiences that have been obtained from human factors and related professionals. In this section we have provided guidelines for developing UCI with new technologies. Throughout we have emphasized the importance of understanding the user's task and how a decision support system should be designed to facilitate that task. We have illustrated these guidelines in the domain of military planning. Our experience with tactical planners shows that they must perform a difficult and complex problem solving task. The guidelines presented in this section will aid the development of a better computer support system for military planning.

Further research must be conducted to determine the limits of these display guidelines. They are not yet refined to the point that designers can use them in handbook fashion. Instead, we hope that guidelines such as these will focus research on the special needs of operators making time-pressured decisions.

3.3 Enhanced UCI Technology Options

The above suggests that there are at least two paths toward the design and development of systems with enhanced user-computer interfaces: conventional "human factors" engineering and UCI principles derived from the cognitive sciences. We believe that while the first field provides a whole host of important opportunities, the second holds even greater promise. In fact, it is pointless to think about enhancements from one area and then another, since real leverage lies in the synergism across the areas. Many usage problems can be traced to adequate conventional human factors considerations that fell short of display routines that offered cognitive compatibility with their users.

The guidelines presented in Section 3.1 offer traditional solutions to enhanced UCI. Those discussed in Section 3.2 offer "unconventional" or cognitive solutions to enhanced UCI. This project called for the exploitation of both sets of opportunities.

As we designed and developed our storyboard we called upon research in both areas. The careful reader will find examples of conventional and unconventional human factors engineering. We tried, for example, to use graphics judiciously and appropriately, we tried to develop easy-to-use and flexible input routines, and we tried to develop output routines that would pass human factors "tests." All of these, and a whole lot more, steps were taken with reference to the conventional human factors engineering literature. We also tried to use metaphors, provide for the prioritization of tasks, provide recall and remember capabilities and embedded process models as navigational aids, permit "deepening" via the use of simulated hierarchical knowledge bases, embed "predictors" via the use of "battle calculators" accessed through "simulate" commands, permit analogical reasoning through access to pertinent "cases," provide "probabilistic" graphic displays, and permit planners to determine the extent to which their ideas are consistent with Army doctrine. All of these steps were taken with reference to the unconventional or cognitive literature. The steps from both sources constituted our "master" UCI option set. We then extracted a set of "high priority" options based on assumptions and hypotheses about which conventional/unconventional sets would provide the greatest enhancements to UCI in the domain in question. The results appear in the system concept for enhanced UCI which, in turn, comes alive in the storyboard prototype.

4.0 THE ENHANCED USER-COMPUTER INTERFACE (UCI) STORYBOARD

4.1 The Storyboarding Process

A good requirements analysis will permit the design and development of a useful prototype. One answer to the question, "why go through all of the trouble?", then, is that insightful requirements analyses yield diagnostic prototypes. In this case, the substantive and UCI (display/interaction) requirements (see Section 2.0 of this report) are the springboard to a useful prototype. There are a variety of prototyping methods, tools, and techniques. While this report stresses the utility of storyboard prototypes, there are other kinds that can, under the right systems design circumstances, yield coherent and verifiable requriements. Some of these methods include narrative descriptions of system functions, conceptual and system flowcharts of system operations, and working models of the evolving system-to-be.

It comes as no surprise that an unreasonably high number of our systems do not serve their intended users well. Part of the problem can be traced to acquisition problems, part to the nature of the support itself, and part to a failure to identify and validate system requirements before the system goes to the field.

This section of the report discusses a new technique for interactive systems design and requirements validation. Anchored in the prototypting strategy (Boar, 1984), the technique calls for the modeling of systems functions immediately after the requirements analysis has been completed. There are a variety of modeling techniques available to the systems designer -- such as narrative descriptions and flowcharting -- but none of these "conventional" techniques serves the application prototyping strategy or verifies requirements via actual screen displays of how the system will operate -- particularly if UCI issues are predominant, as they were for the research reported here.

Storyboarding is a technique that has become popular in the production of motion pictures over the past decade. Cost-conscious directors prefer "seeing" scenes before they are shot, just as producers applaud the cost-effectiveness of "try-before-buy" procedures. Storyboarding is also widely used in the production of animated features, cartoon strips, and even in the production of children's books.

Only recently has bona fide storyboarding become part of the interactive computer-based systems design and development process. For years designers have developed paper screen dumps of intended displays that have proven effective with users, though not nearly as effective as computer-based storyboards.

4.1.1 Storyboarding in the Design Process - The systems design process consists of the following steps (see Figure 4.1):

- System/Problem Feasibility Analysis;

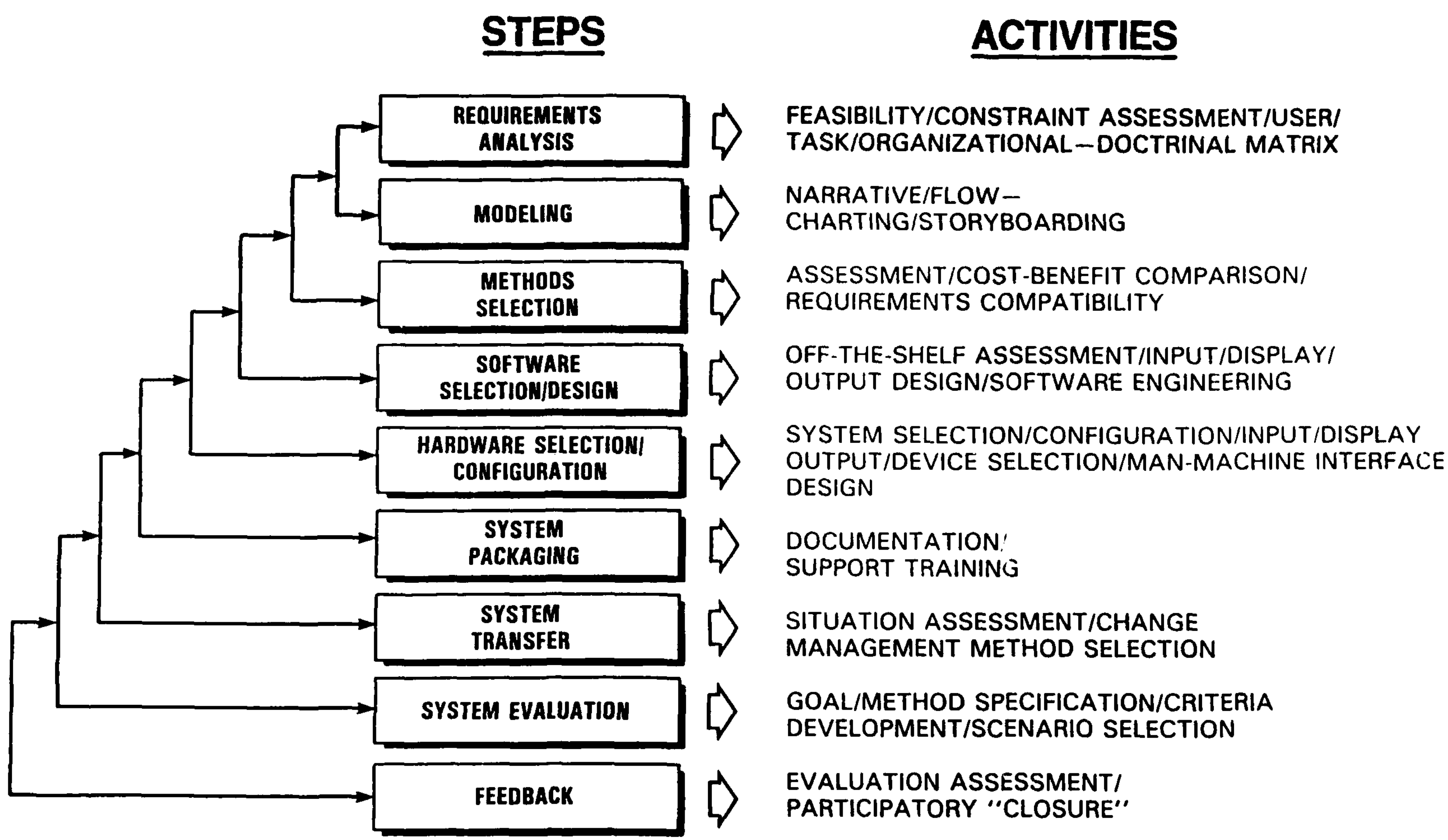


FIGURE 4.1: Prototype System Design Process

- **User, Task,** Organizational-Doctrinal Requirements Definition Development;
- System Modeling (and Re-Modeling);
- Analytical Methods Selection;
- Software Design Specification;
- Hardware Configuration Development;
- System Engineering;
- System Packaging and Documentation;
- System Evaluation and Testing; and
- System Transfer.

The underlined systems modeling step is where the storyboarding technique is first applied. It is also assumed that the entire development process will be iterative, and that requirements will be redefined as the process evolves (Andriole, 1983, 1988; Boar, 1984). None of the above steps should thus be regarded as perfectly sequential; good systems design and development may require all of the steps to be taken several times.

Storyboarding is a modeling technique that borrows from requirements analysis and simulation methodology. A storyboard is a sequence of displays that represents the functions that the system may perform when formally implemented. But unlike paper screen dumps, the modern storyboard is computer-based. When well done, it communicates to intended users system functions that could only be described piecemeal in static paper displays or words.

Storyboards are "live" tools. They are designed to verify requirements definitions, help "size" systems design specifications, and serve as compasses to software engineers. They are dynamic, subject to user and technical review. They are also intended to consume relatively little of the systems design and development budget, while protecting that same budget from false starts, inaccurate requirements definitions, and over-eager programmers. Storyboards are especially suited to UCI research, where hypotheses about enhanced UCI can be tested via interactive storyboards of alternative displays, interaction sequences, dialogues, and the like.

Figure 4.2 suggests the essence of the storyboarding technique, which assumes that great cost-effectiveness can be gained from developing working models of interactive systems before any programming commitments are made.

Figure 4.3 suggests that the the screen displays of general person-vehicle (or system) interfaces, expert systems, natural language processors, and conventional data bases can be used to validate requirements, that system "sizing" can be accomplished via the storyboards, and that low-cost simulations of system capabilities can be developed around interactive storyboards.

4.1.2 Storyboard Development - Assume for the moment that a thorough requirements analysis has been conducted and profiles exsit for the users, tasks, and organization-doctrine that the

- **Application Prototypes**
  - "Working Models" of System Functions → Proof of Concept Demonstrations
  - Support "Sizing" for Application Systems Development
  - Particularly Suited for "Analytical" Computer-Based Systems Design & Development
  - Cost-Effective

FIGURE 4.2: Applications Prototyping Specified

- **Animated Storyboards**
  - Computer-Based Screen Displays of PVI, Expert Systems, Natural Language Processors for Requirements Validation
  - Extremely Low-Cost/High Payoff Alternative to Conventional Functional Modeling
  - Support System "Sizing", i.e., Identification of Size & Nature of Data/Knowledge Bases, Selection of Analytical Methods, & Specification of PVI
  - Support Low-Cost Simulations of System Capabilities
  - *Early* Demonstration of System Capabilities...

FIGURE 4.3: Animated Storyboards Specified

system is intended to serve. Conventional systems designers might immediately convert these profiles into logic flowcharts for the software engineers. These charts may or may not be passed by the users, though even if they were chances are that the users would not know how to interpret them. Other designers might describe what the system will do in a narrative, though narratives also fail to capture the essence of intended system operation and performance.

Perhaps the best approach requires the design team to develop a set of displays that represent each and every path users might take once the system is developed. An even better approach involves computerizing the displays to run interactively in a realistic (scenario-based) setting.

The procedure itself is straightforward. First, assemble members of the design team as well as users (ideally, users will already be members of the team). Discuss the storyboarding concept, anchor it in the results of the previously conducted requirements analysis, and nominate one team member to develop a strawman paper storyboard. This strawman will accelerate progress, since many designers cannot work productively in the abstract. Require team members and users to revise the paper storyboard by actually redoing individual or groups of displays. When rough consensus emerges then convert the paper storyboard into software.

Depending on the problem, storyboards can range anywhere from fifty to five hundred displays. It is important to keep the burden on your user review group as light as possible. Good storyboards may not display every possible housecleaning option in a proposed system; the purpose of storyboarding is to represent all of the important functions that the system will perform as well as the output that it will generate. Input requirements will naturally fall out during the evaluation of the storyboard.

4.1.3 Storyboard Implementation - One preferred storyboard configuration is an Apple Macintosh with an external disk drive (or, ideally, a hard disk) connected to either an Apple Imagewriter or Laserwriter printer. The Macintosh is flexible, has extremely high resolution, and can exploit a variety of tools that seem to have been tailormade for storyboarding (in fact, some of them have). It should be noted immediately that there are several other hardware/software configurations capable of supporting the development of interactive storyboards. At the most basic level, there are software systems that permit the design of screen displays of evolving systems, IBM PC-based tools for storyboarding, and tools available on much more expensive LISP-based systems, like the Symbolics, TI Explorer, or LMI series of development workstations. Our experience has been with the Apple Macintosh, PCs, and the LMI systems; this report is confined to a discussion of Macintosh applications and, in subsequent sections, with how some Apple Macintosh-based tools have been used to design, develop, and demonstrate storyboards.

There are a variety of tools available for storyboarding on the Macintosh. (Storyboarding on the Macintosh can be achieved on a Macintosh 512K, a Macintosh Plus, a Macintosh SE, and/or a Macintosh II [when color is appropriate].) They include the following:

- MacDraw;
- MacPaint (FullPaint, SuperPaint, and the like);
- Storyboarder;
- Videoworks II;
- Slide Show Magician; and
- Hypercard.

These tools permit the development of the screens themselves and arranging them in sequence for presentation to prospective users. In our experience the key programs include Macdraw, Macpaint (or equivalent), Slide Show Magician or Storyboarder, and -- for the adventurous -- VideoWorks II. Some others have more special purpose functions that may or may not satisfy your needs. Of special current interest is the new Apple Hypercard* system. Hypercard combines many of the features of several of the above programs and may yet become the premier storyboarding tool.

It is possible to build storyboards with just a Macintosh and Macdraw/Macpaint/Slide Show, but there are faster ways.

---

*Hypercard is a trademark of Apple Computer, Inc.

Given that many storyboards will have graphic displays, it might make sense to invest in some tools that make it fast and easy to input maps, photographs, and the like for subsequent manipulation. Some of the tools that make this possible are listed below:

- MacVision;
- Thunderscan;
- Micro-Imager;
- Mac Private Eye; and
- Omni-Reader.

These tools make it possible to enter all kinds of data and information (as MacPaint and/or MacDraw files), thereby freeing the storyboarder from having to rely on Macpaint or Macdraw to re-create the data/images manually.

After the storyboard is completed but before users are invited in to evaluate the mock system, several products might be considered that permit large screen display of the storyboard (to large or working groups, for example). The products below support large screen projection of Macintosh images:

- Big Mac Monitor;
- Project-A-Mac;
- Composite Video Adaptor; and
- CineMac, among others.

4.1.4 Storyboard Testing - When the storyboard is put to the test make sure that paper copies of the screen displays are available for annotation. The experienced user can actually annotate on the screens themselves, but to avoid clutter it makes practical sense to give each user time on the Macintosh and their own hard copy of the the mock system's output. It may also be useful to tape record their comments as they proceed throught the system and later compare the utterances of all of the users.

Note again that the storyboard is "living." After the test runs it will be necessary to make changes and conduct more tests -- and so on, until consensus emerges about what the system should do and how it should do it. Even after the modeling phase is passed, the storyboard will serve double duty as a software design tool, especially from a human factors/user-computer interface perspective.

4.1.5 A Storyboarding Case Study - The storyboards that follow this section were extracted from a tactical planning system concept developed for the United States Army's Ballistic Research Laboratory (BRL; Andriole and Hopple, 1987).

The storyboards that follow suggest how a planner would use some advanced analytical techniques to infer and display adversary intentions and how he might respond to a "predicted" adversary course of action. The boards in the sequence were extracted directly from the master storyboard which consists of one hundred thirty separate displays that run interactively on a Macintosh Plus.

The board was demonstrated to users and their reactions were recorded on paper copies of the screens; adjustments to the system concept were made based upon these comments and suggestions.

Our experience with storyboarding suggests clearly that it represents a clear path toward requirements validation and verification. Storyboards may serve as throwaway <u>or</u> evolutionary prototypes. New storyboarding tools like Hypercard will permit the design and development of storyboards and the evolution of the prototype as requirements are refined -- all within the same Hypercard programming environment.

This report suggests some new approaches to requirements analysis, verification, and validation. The preferred technique is derived from the larger value ascribed to prototyping; the specific approach to prototyping is the design, development, and testing of interactive, computer-based storyboards. Our use of storyboard prototypes has proven extremely successful. Complex requirements have been modeled and validated via storyboarding; users have been integrated into the design process via storyboarding; and programmers have been able to "size" subsequent software engineering projects accurately via storyboarding (see Section 5.0 for details on the sizing of the Human Engineering Laboratory [HEL] storyboard). New tools that

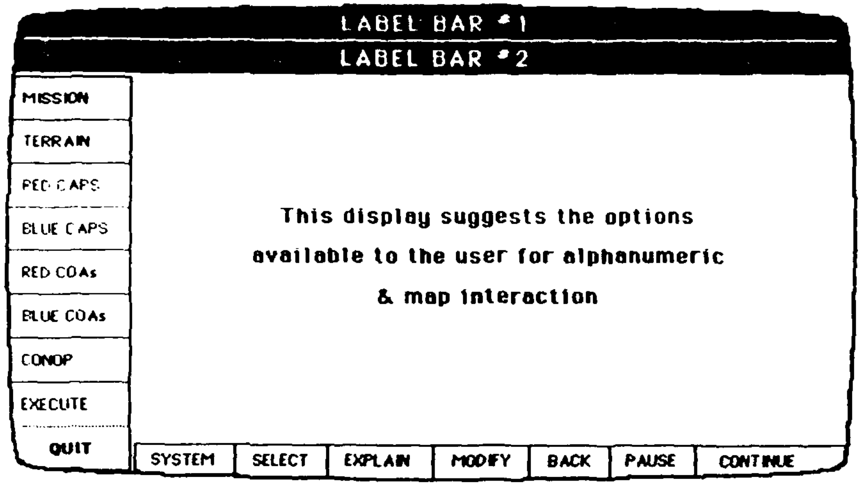
INFERENCE - MAKING &
OPTION SELECTION IN
TACTICAL PLANNING

SYSTEM CONCEPT
STORYBOARD

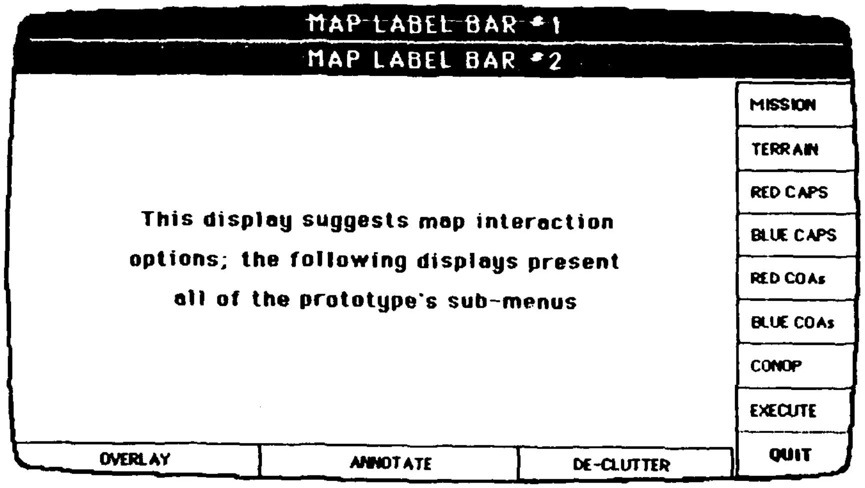
The Storyboard/system concept is
based upon :

The Airland Battle Management Program Corps Operations Scenario prepared by the BDM Corporation for DARPA (BDM/R-85-0987-TR, October 31, 1985)

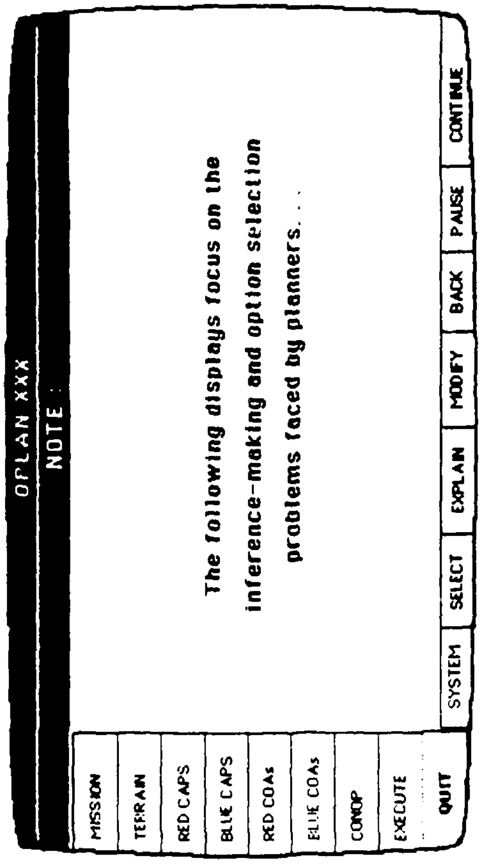
START

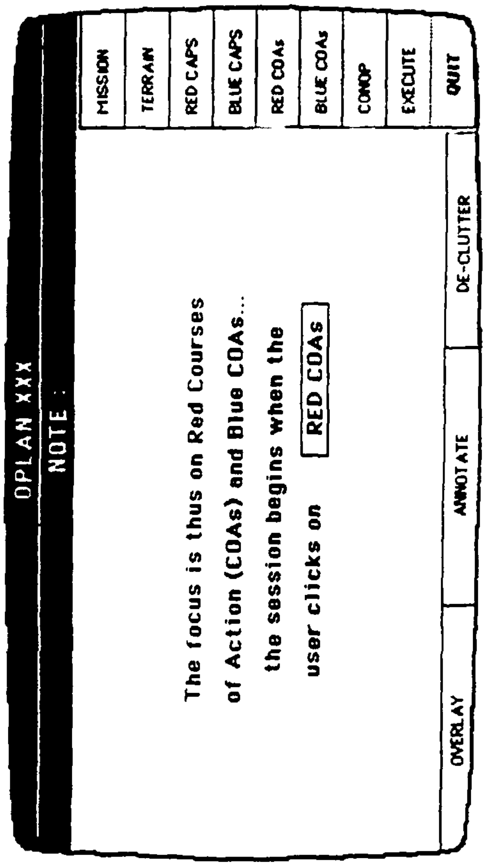


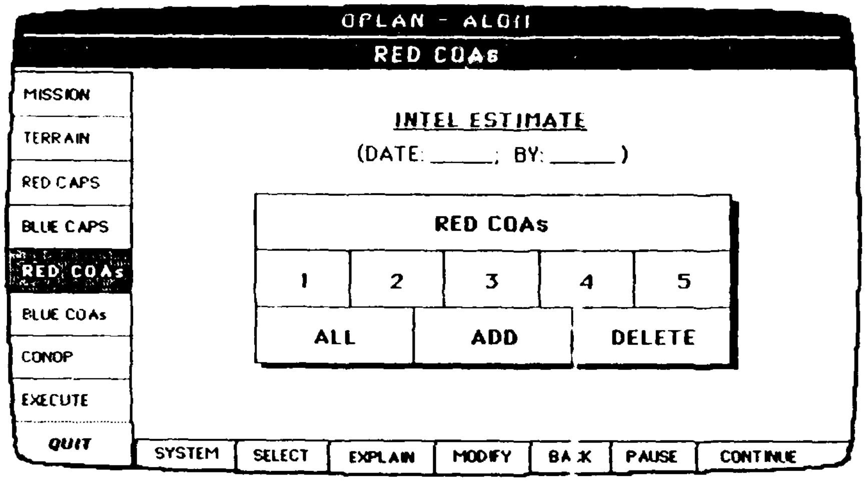
The options along the left hand side and across the bottom of the screen suggest how the planner might execute planning functions; notice the "elements" of tactical planning along the left and "system" functions across the bottom. Any of these functions can be executed at any time; the ordering of the options in the vertical and horizontal menus does not suggest sequencing . . .



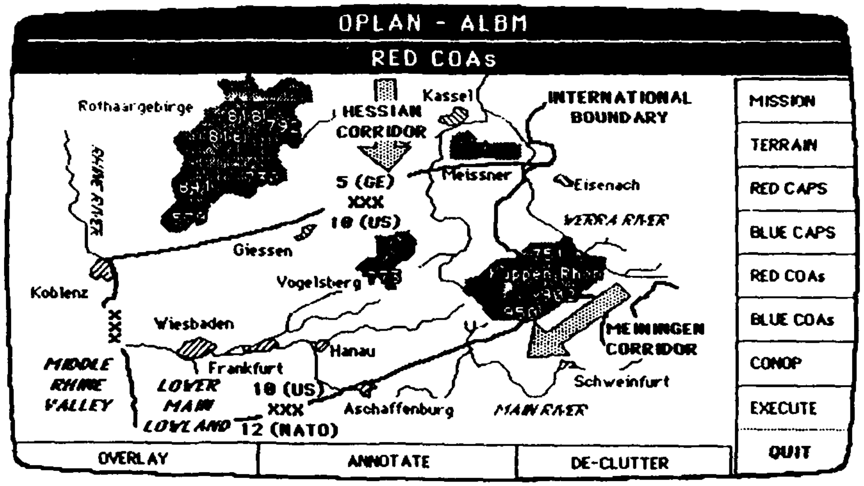
These options are for the display screen. Note that the system concept here calls for a dual screen display configuration with standard alphanumerics on the left (with some graphics) and a map display on the right (these positions can be reversed depending on the preferences of the user). The "overlay" option allows the planner to "mix and match" overlays; the "annotate" function permits the planner to write directly onto the screen, while the "de-clutter" option permits the planner to remove overlays (either system or user-generated) from the screen.


OPLAN XXX
NOTE :
The following displays focus on the
inference-making and option selection
problems faced by planners. . .
MISSION
TERRAIN
RED CAPS
BLUE CAPS
RED COAs
BLUE COAs
CONOP
EXECUTE
QUIT
SYSTEM
SELECT
EXPLAIN
MODIFY
BACK
PAUSE
CONTINUE

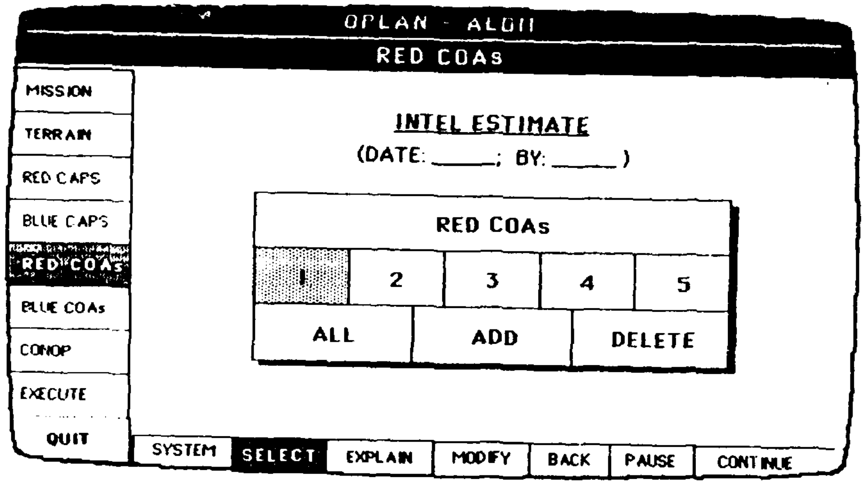

OPLAN XXX
NOTE :
The focus is thus on Red Courses
of Action (COAs) and Blue COAs...
the session begins when the
user clicks on
RED COAs
MISSION
TERRAIN
RED CAPS
BLUE CAPS
RED COAs
BLUE COAs
CONOP
EXECUTE
QUIT
OVERLAY
ANNOTATE
DE-CLUTTER



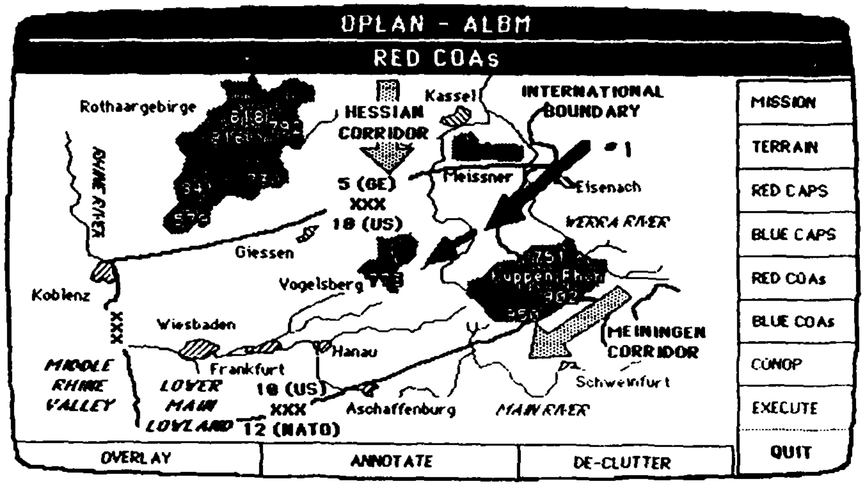
This display suggests that when the planner "clicks on" "RED COAS" a display appears that reflects input from G-2. In this hypothetical case five RED COAs are suspected by G-2 as possible. The system gives the planner the option to see one, two, three, four, or all of the COAs. It also permits hims to click on "ADD" where he can add a sixth or seventh hypotheses to the estimate. Note that we are not here suggesting that the G-3 perform the G-2's job; rather, the assumption is that G-3 personnel are interested in the elements of the estimates on which they will base their operational decisions . . .



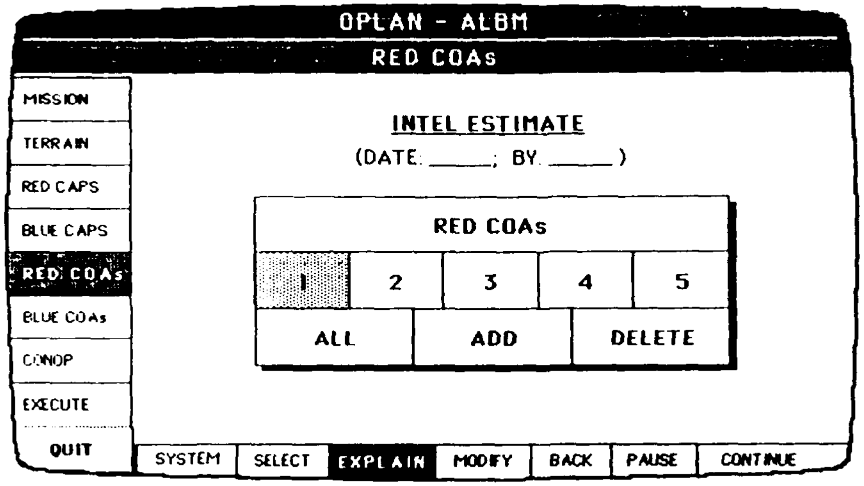
The map appears for the first time. There is something for the system to "say" since a specific request has been made. Note that we entered the planning process in the middle since our focus is on inference-making and option selection; at this point in a full process, the planner would already be familiar with the map . . .



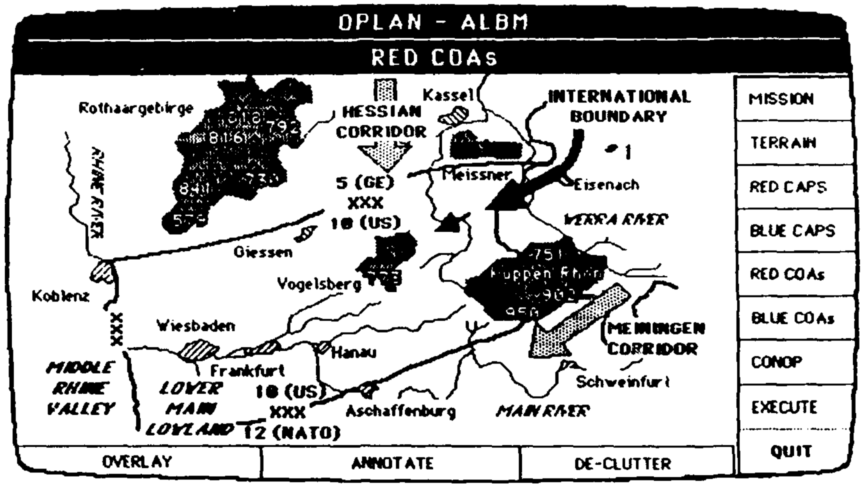
The planner clicks on "1" . . . and also on "SELECT" though both clicks would not be necessary . . . note also that a quick look at the screen suggests to the planner exactly where he is in the planning process. The top labels suggest he is building (a fictitious ALBM) plan and that he is looking at RED COAs. The left elements options also lists "RED COAs" and the bottom option illustrates that he has selected "1" in the middle of the display. *This redundancy and navigational aspect of the display is important, especially to the novice user of a system such as this . . .*



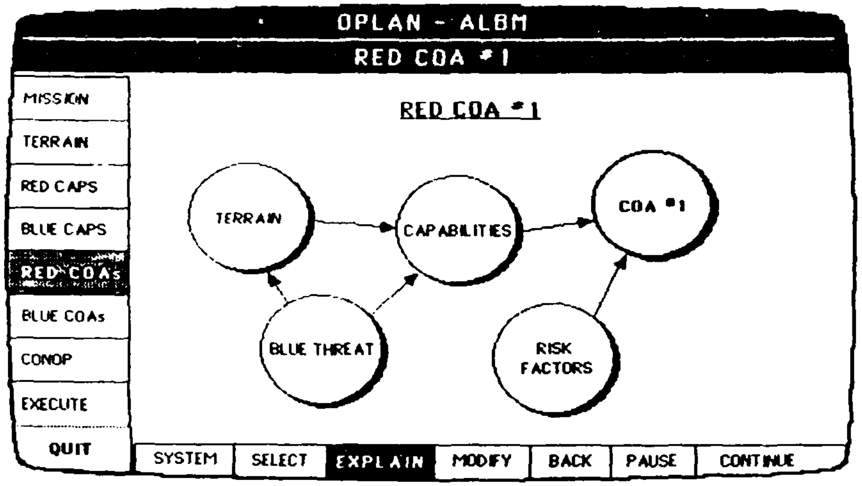
The map displays RED COA #1 . . .



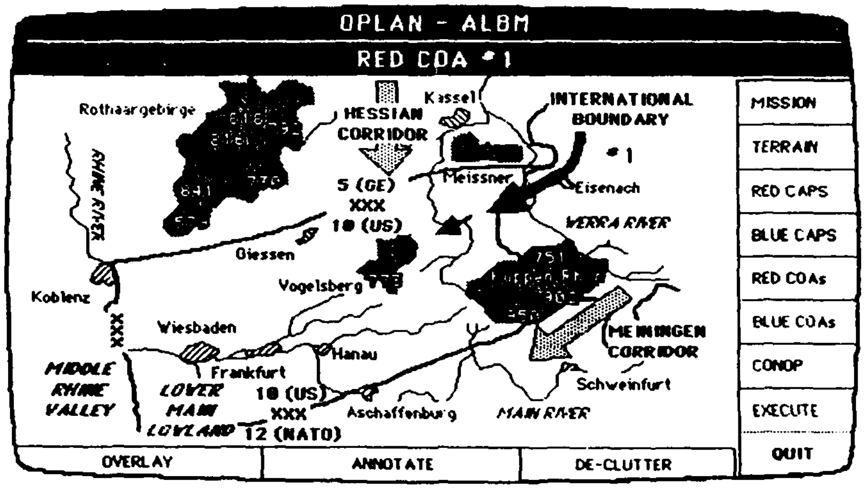
The planner returns to COA #1 since it strikes him as either the most likely or most interesting . . . and asks the system to "explain" it . . .



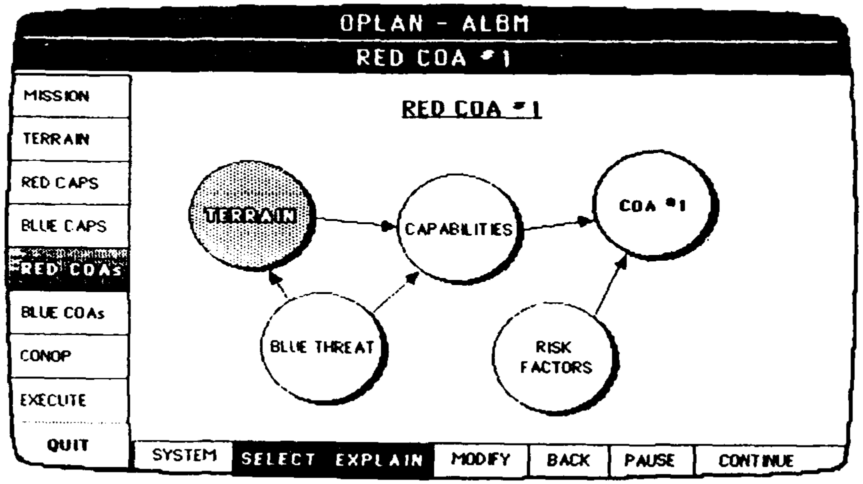
The system displays only COA #1, as the planner requested and sets up its explanation . . .



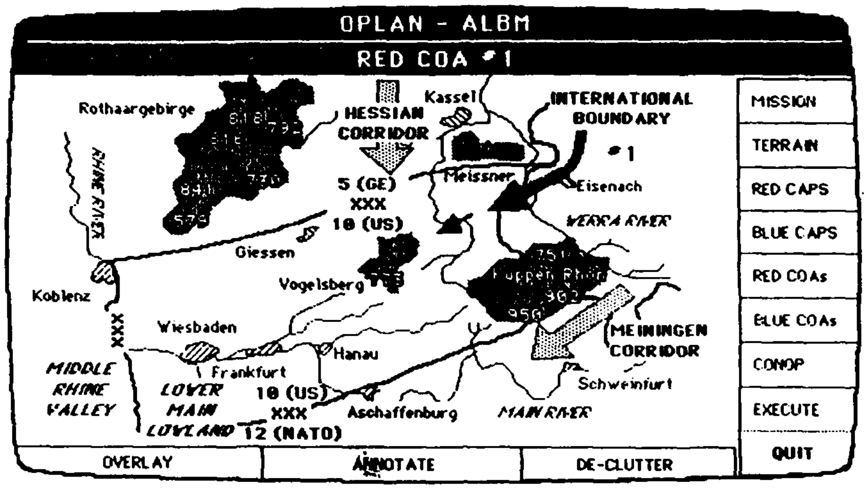
The system generates an **influence diagram** that explains the inferencial relationships among the variables that it assumes are "driving" the likelihood of COA #1. This influence diagram is embedded in the system, it was created by expert planners who when confronted with similar planning problems identified these variables and their interrelationships



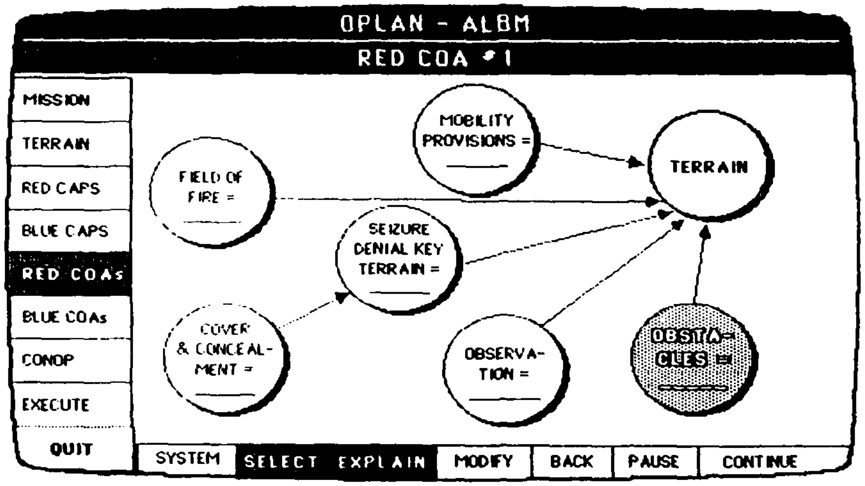
The map display remains the same . . .



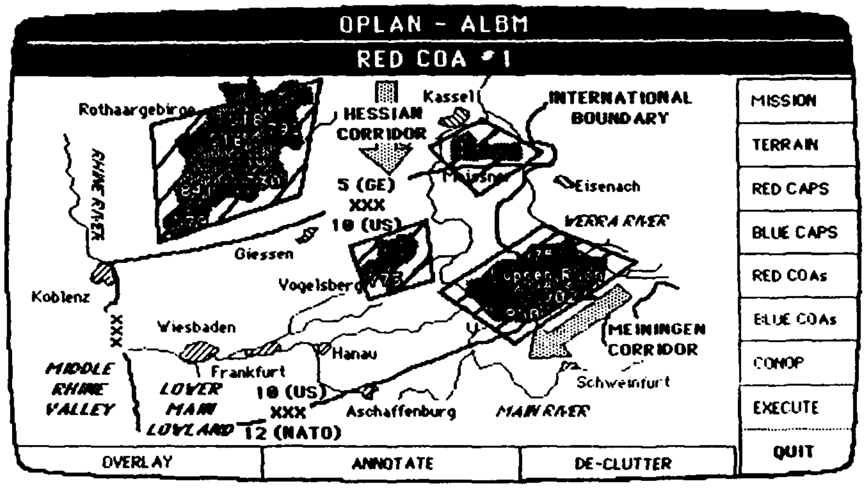
The planner, seeking more detail about the inferencial relationships, clicks on "TERRAIN" . . . *In effect, he is descending hierarchically into the* inference structure looking for more data to support the likelihood hypothesis . . . he wants to know in this specific instance precisely what it is about "*TERRAIN*" that is exerting a "causal" impact on capabilities and Blue threat . . .



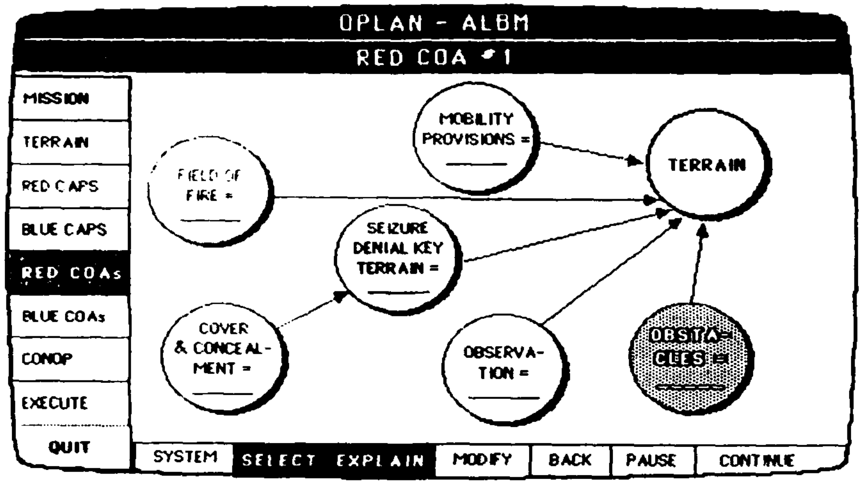
The map does not change . . .



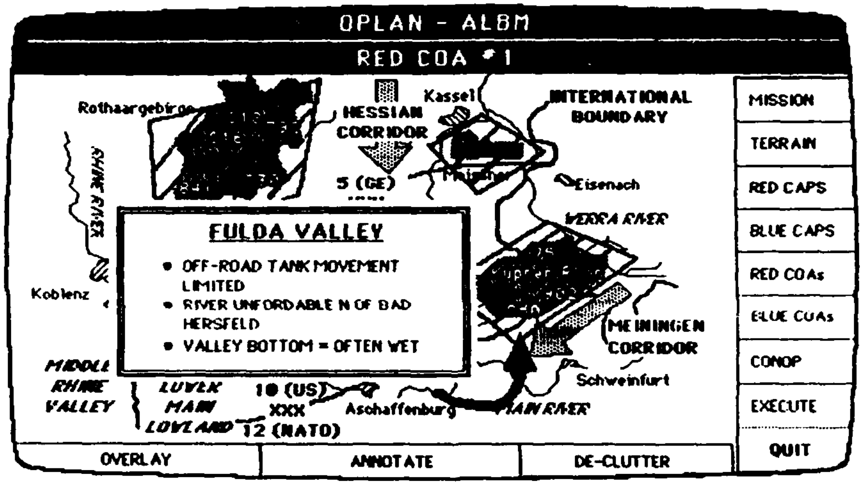
Another level of inference-making is displayed to the planner. Now he sees that there are a number of variables that comprise terrain . . .he looks over the variables and decides to find out more about "obstacles" ; note that he is still within the RED COA primary function and is now inquiring about but one of the variables that the system assumes are driving terrain (which in turn is driving capabilities and Blue threat . . .



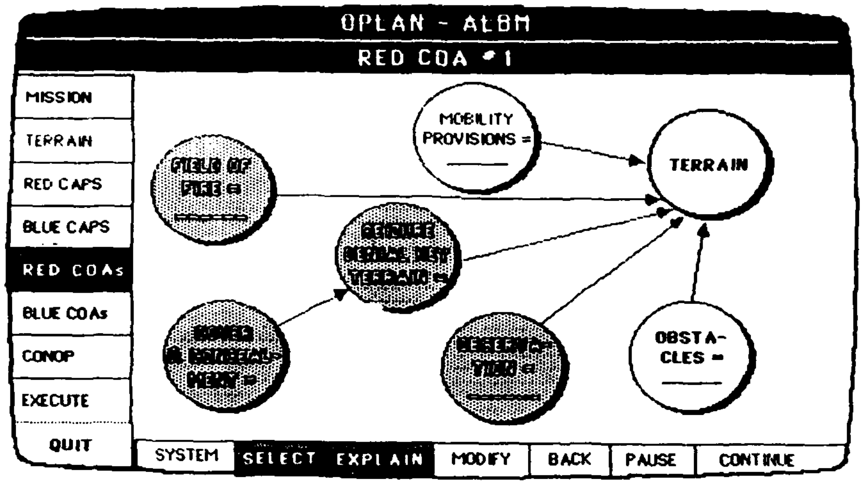
The map display shows him the obstacles . . .



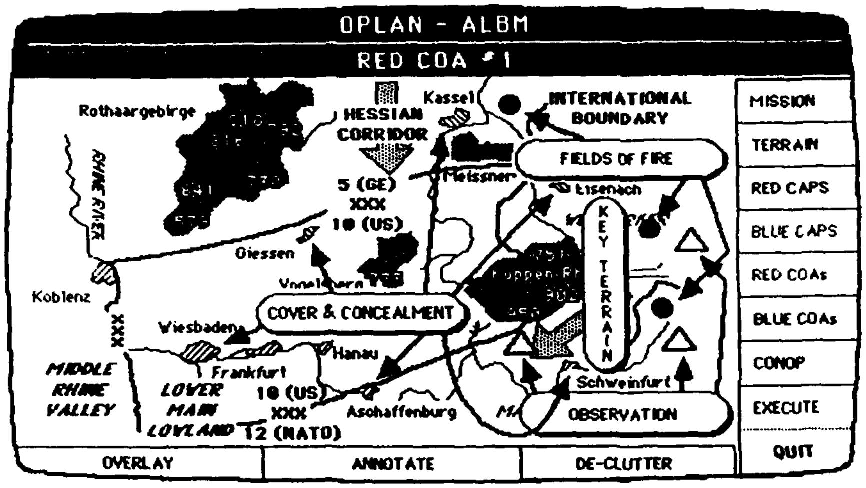
The planner again clicks on explain. Here the hierarchical explanation capability is demonstrated. As long as the user clicks on a function and as long as there is data or a sub-function below it, it will respond . . .



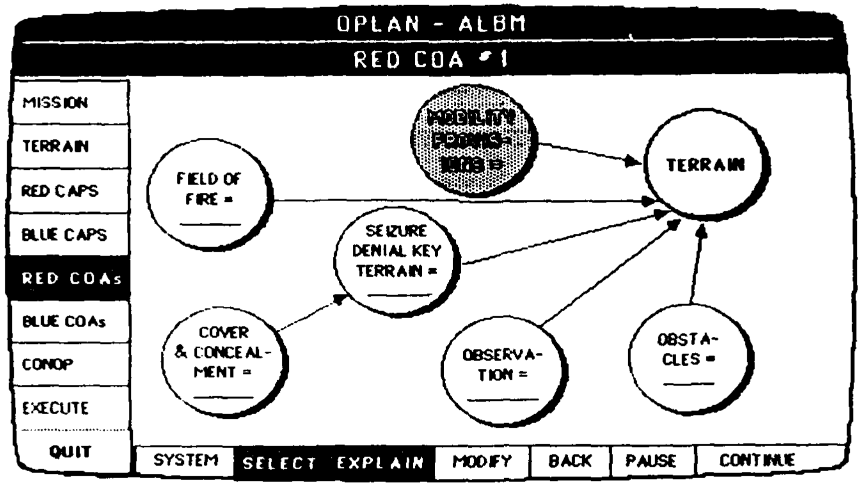
The system responds with a window text explanation on the map display . .



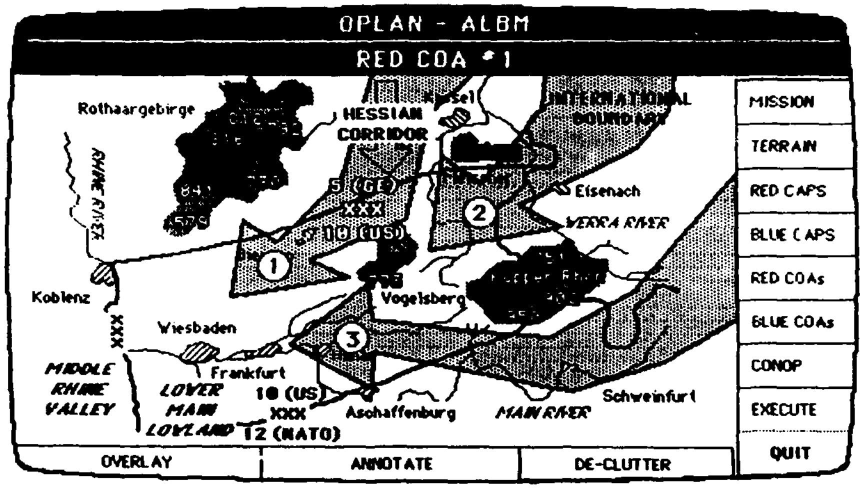
Now the planner wants to see a variety of the terrain components . . . he clicks on four of the six elements . . .



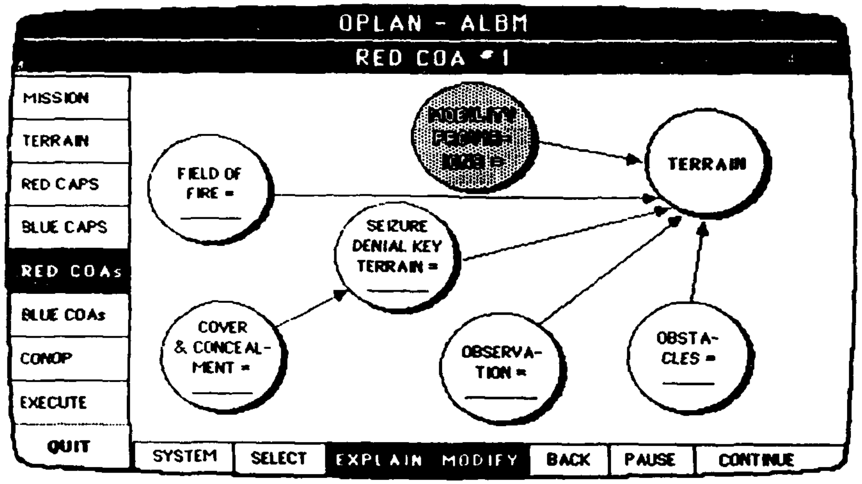
The system displays all of the terrain features . . . note that the planner can now de-clutter the image and see only, for example, "cover and concealment" or "observation points" . . .



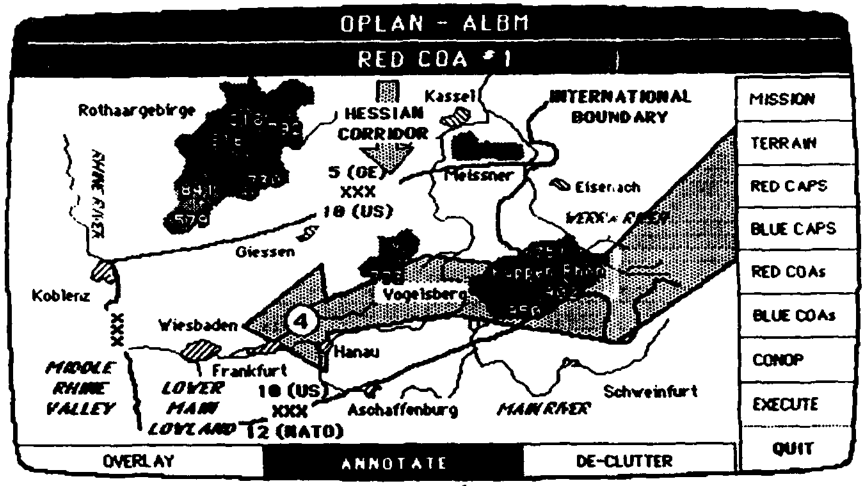
Now the planner wants to see yet another aspect of terrain, :mobility provisions" . . . he clicks on the bubble and waits for the display to change.



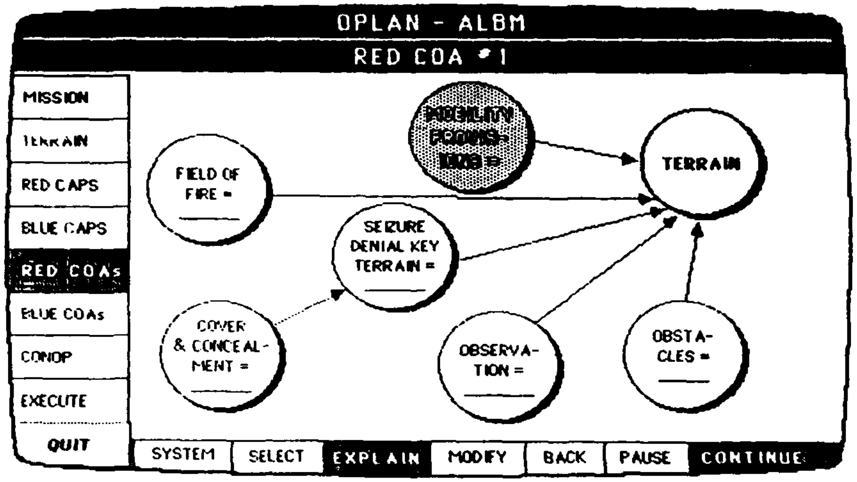
The map displays the corridors most likely connected with RED COA #1 . . .



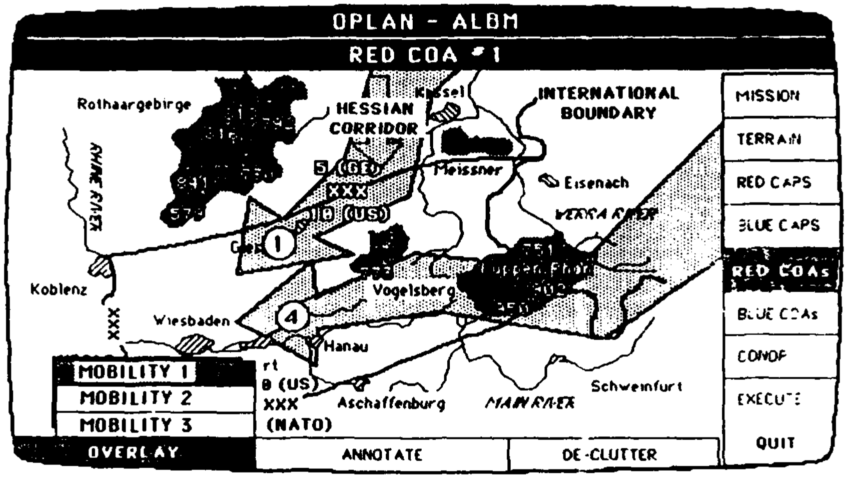
The planner clicks again and receives the next most likely mobility corridor . . .



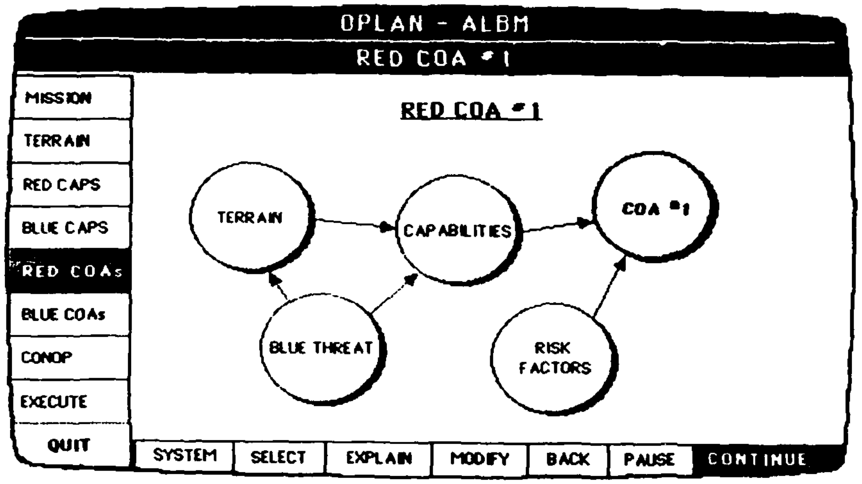
The map responds accordingly . . .



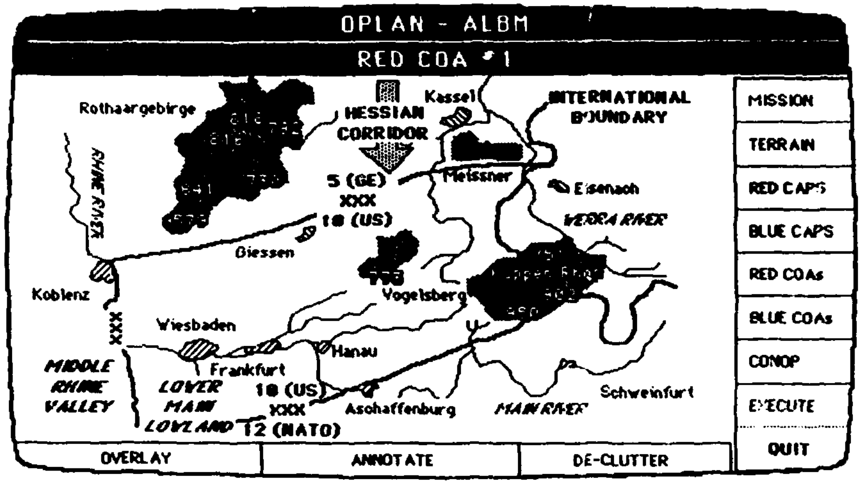
The planner stays with "mobility" and requests the system to hold while he does some comparing (see the map display) . . .



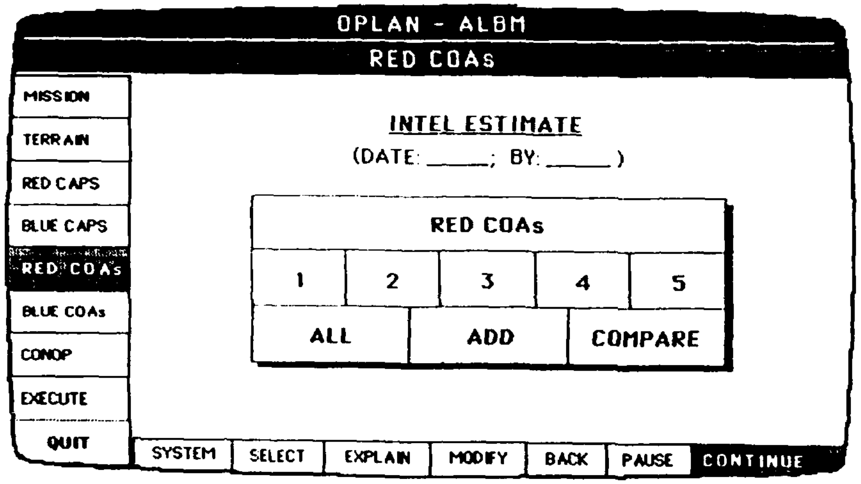
The planner decides to compare MOBILITY 1 with MOBILITY 4 . . .



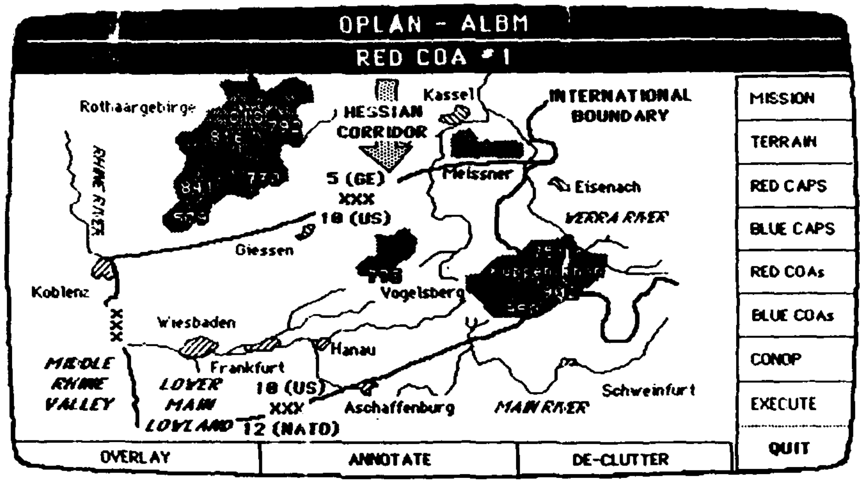
The planner backs out of the terrain function by clicking on "continue" . . .



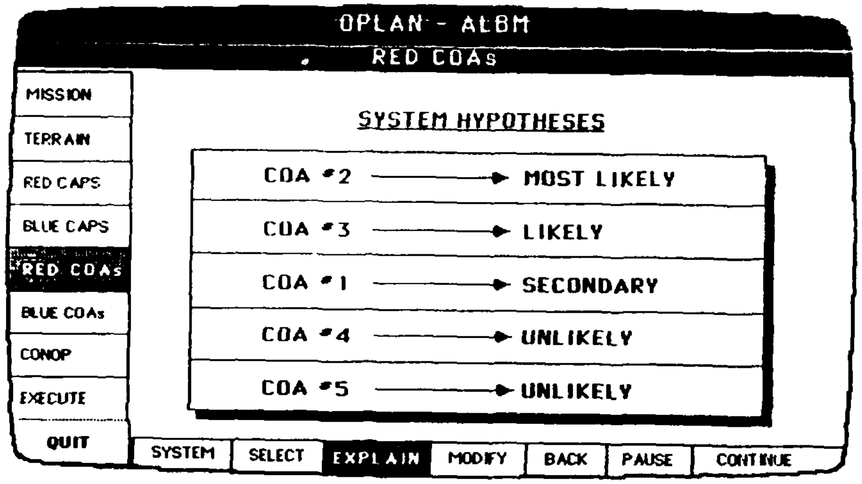
The display returns to its initial state . . .



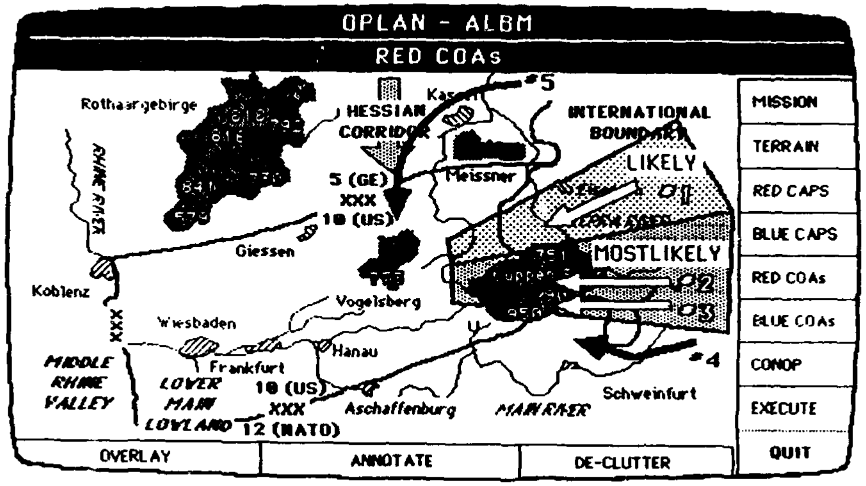
The system then ascends to the original display regarding RED COAs . . .



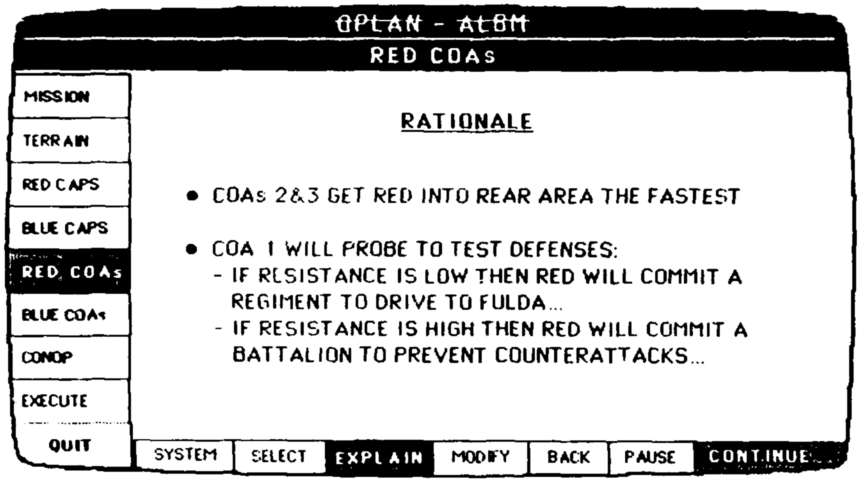
The map display also returns . . .



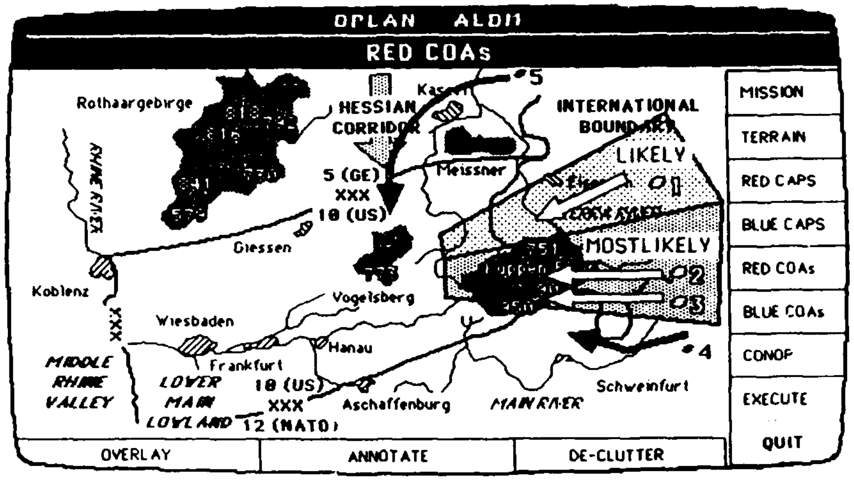
The planner then again clicks on "explain" and, since he has just cycled through an explanation, the system generates a list of the most and least likely COAs as proposes them as system hypotheses. This is an ex .mely important aspect of the interaction sequence, since the system, via its intelligent and embedded influence diagrams (and, presumably, its more sophisticated hierarchical inference structures), is now suggesting to the planner where RED will come from and how . . .



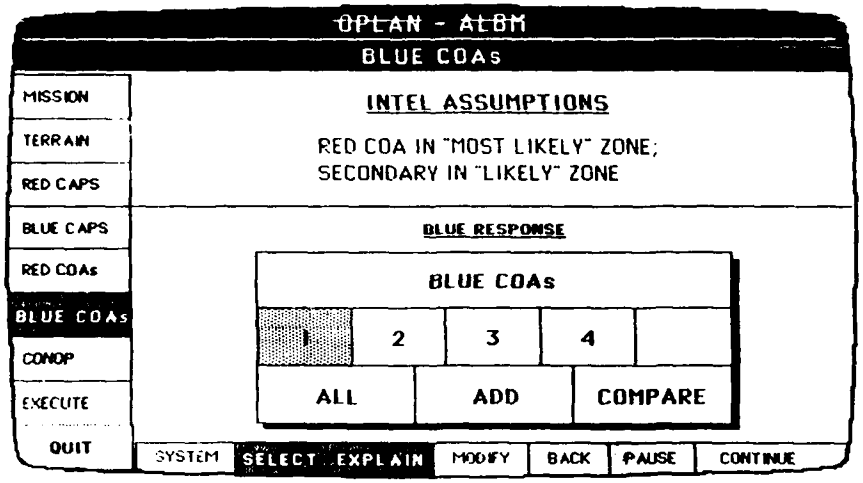
The system displays "zones" of likelihood. This is an important display since it suggests how one might use "analytical graphics" to express essentially qualitative concepts like "likelihood," "risk," and "opportunity" . . .



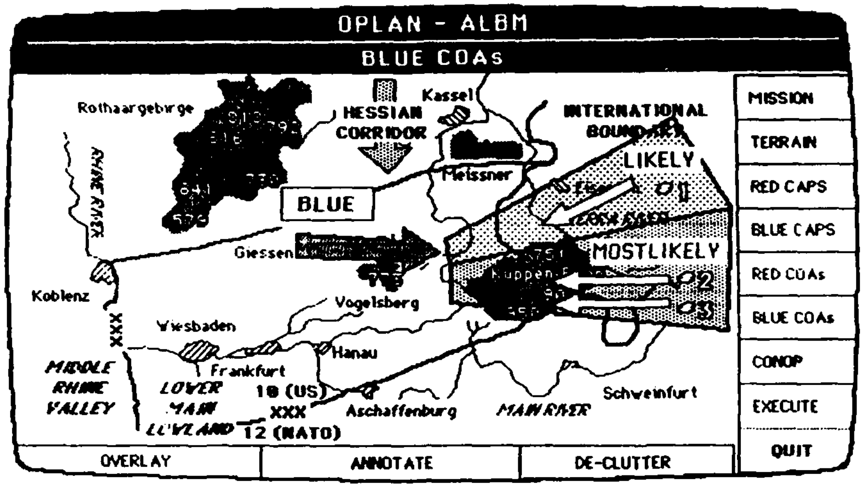
Now the planner wants more detail about why the system feels that COAs #2 and 3 are more likely than the others . . . the system responds with an explanation.



The map display does not change . . .



Satisfied, the planner clicks on BLUE COAs on the left menu and waits for the system to respond . . . the same kinds of options he saw regarding RED COAs now appear for BLUE COAs . . . he clicks on "1" . . . he also clicks on "explain" . . .



The map displays BLUE COA #1 . . .

run on Apple Macintosh's (and IBM PCs) have made the use of storyboards cost-effective. As we learn more about the design, development, and testing of storyboards we expect their utility to grow.

This project's use of interactive storyboards -- reported in detail in the next section of the report -- has added to our experiencial database about the utility of storyboarding. Given the unique UCI hypotheses tested via the storyboard, the results were very positive (see Section 5.0).

4.2 The Enhanced User-Computer Interaction (UCI) Storyboard

The storyboard developed for this project was implemented on an Apple Macintosh Plus with MacDraw*, MacPaint*, Thunderscan**, and the Slide Show Magician***. Paper copies of each board and each interactive sequence were developed prior to conversion to the Macintosh. The paper storyboard was based upon (a) the substantive and UCI requirements hierarchies and the (b) conventional and unconventional UCI technology opportunities that we identified during the course of our research.

Once the paper storyboard was completed it was converted into

---

*MacDraw and MacPaint are trademarks of Apple Computer, Inc.

**Thunderscan is a trademark of Thunderware, Inc.

***Slide Show Magician is a trademark of Magnum Software, Inc.

Macintosh-based equivalents. The board runs interactively; only the mouse is required to run the prototype.

4.2.1 The Storyboard Scenario - As suggested throughout this report, the domain that served as the substantive basis of the project was Corps tactical planning. In order to lend realism to the domain, and to permit the evaluation of the enhanced UCI concepts, we selected a credible scenario, that is, one that would permit us to test new ideas in a context that would permit us to draw some operational conclusions from our work. We selected the Letort scenario, a Corps-level scenario developed at the Army War College in Carlisle, Pennsylvania. The scenario is a familiar one: an impending Warsaw Pact attack into Western free Europe. The simulated Corps is given seventy-two hours to develop a defensive plan. Figures 4.24 and 4.25 present the maps from the Letort scenario (Army War College, 1984, 1985). The "Blue" Corps Commander -- of the fictional U.S. XI Corps (part of the Middle Army Group, or "MIDAG") -- is given a defensive mission at a point in time and asked to defend in sector, re-establish the international boundaries, and then move on into Berlin (if necessary and possible). The scenario begins with "Red" movement and the Corps Commander's development of a tactical plan. Intelligence (G2) information and estimates are part of the scenario and the Corps Operations (G3) staff are tasked with the identification and evaluation of alternative defensive courses of action.

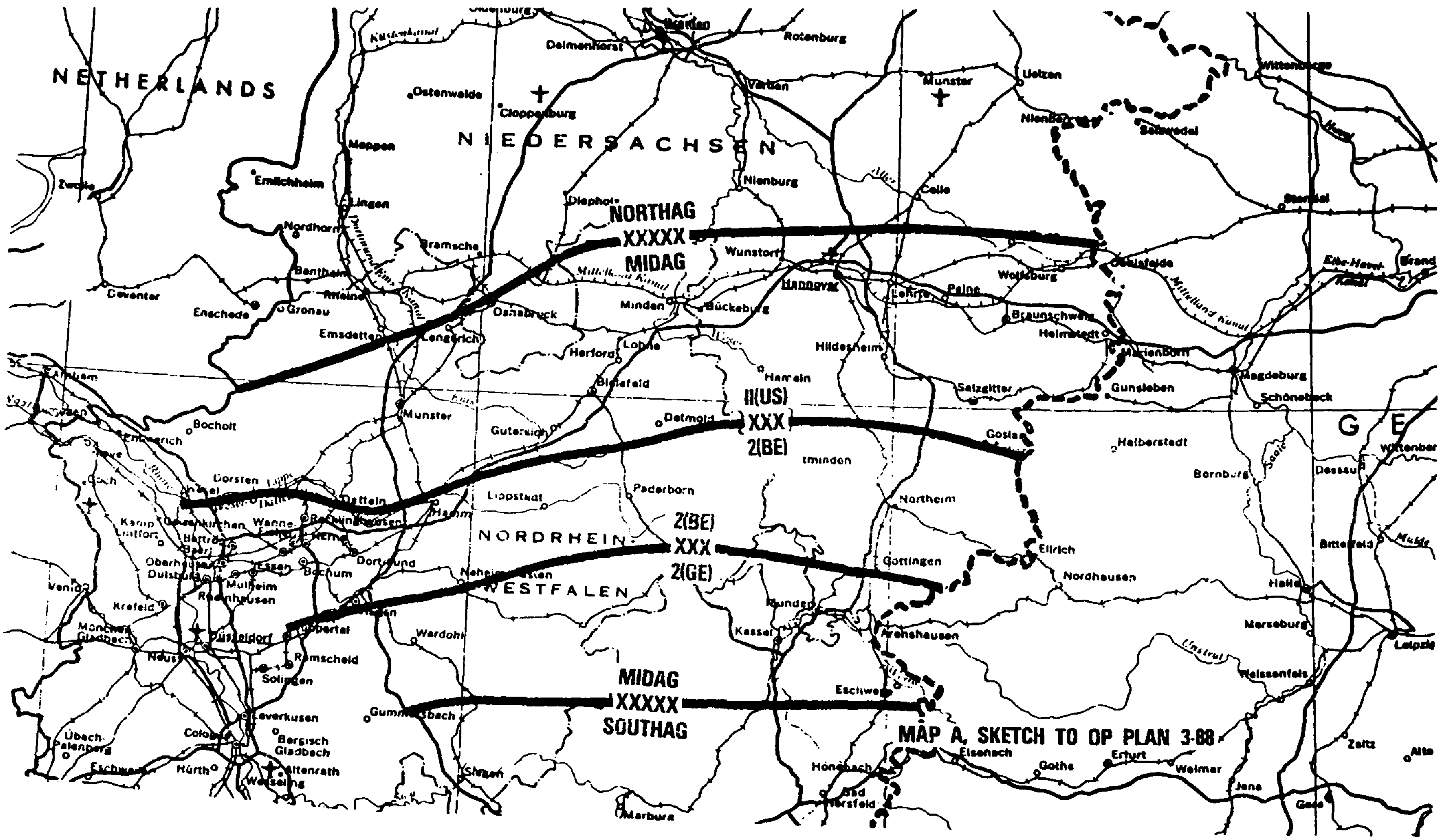


FIGURE 4.24: Blue Area of Operations Map

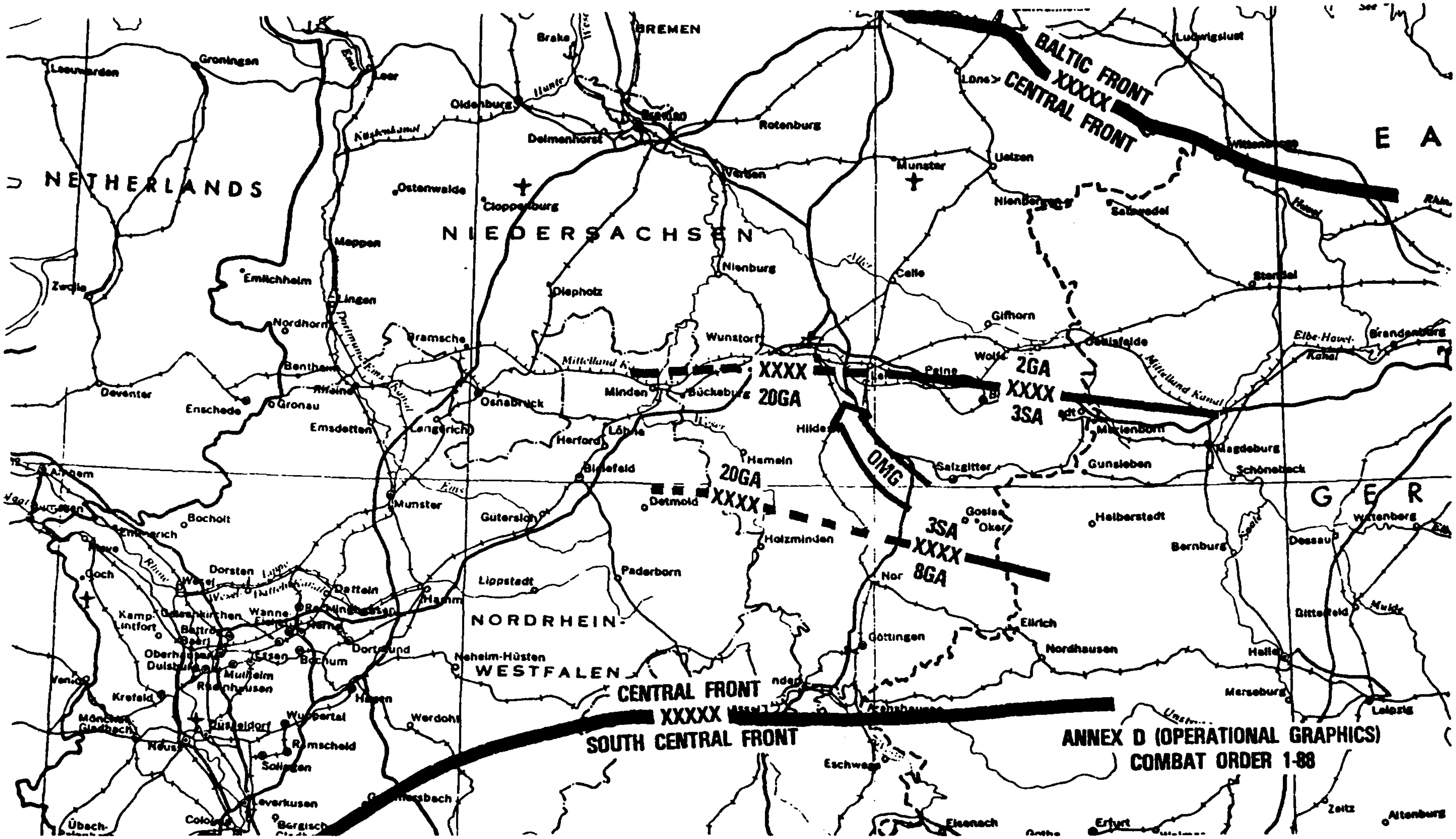


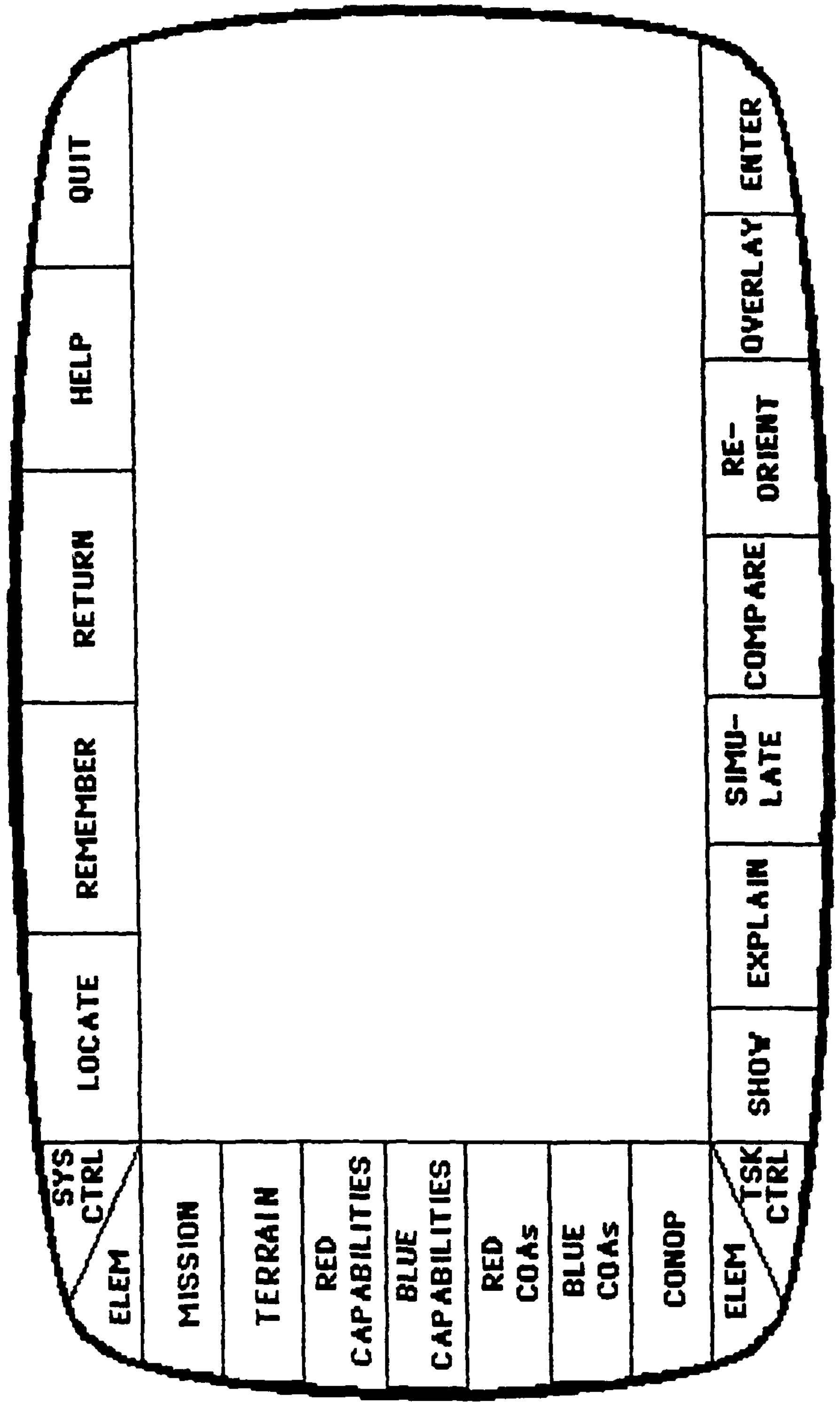
FIGURE 4.25: Red Area of Operations Map

The data and information in the storyboard is taken at times directly from the unclassified Letort scenario. In addition, judgments about Red and Blue courses of action (COAs) were also taken from actual estimates about feasible COAs elicited from real tactical planners (Andriole, 1984; Andriole, et. al., 1986).

This "realism" is important to the design, development, and testing of the advanced UCI system concept. It also permitted us to conceive of an interface that we knew would "play" in the operational community, at least to a significant extent.

4.2.2 The Master Menu/Interface Structure - Figure 4.26 presents the master menu structure. This structure is extremely important to our overall system concept. We have made a number of assumptions about what a functional, easy-to-use, easy-to-learn interface should look like. As suggested above, the system concept is anchored in our requirements data (see Section 2.0 of this report). It is also an outgrowth of our previous work in the area.

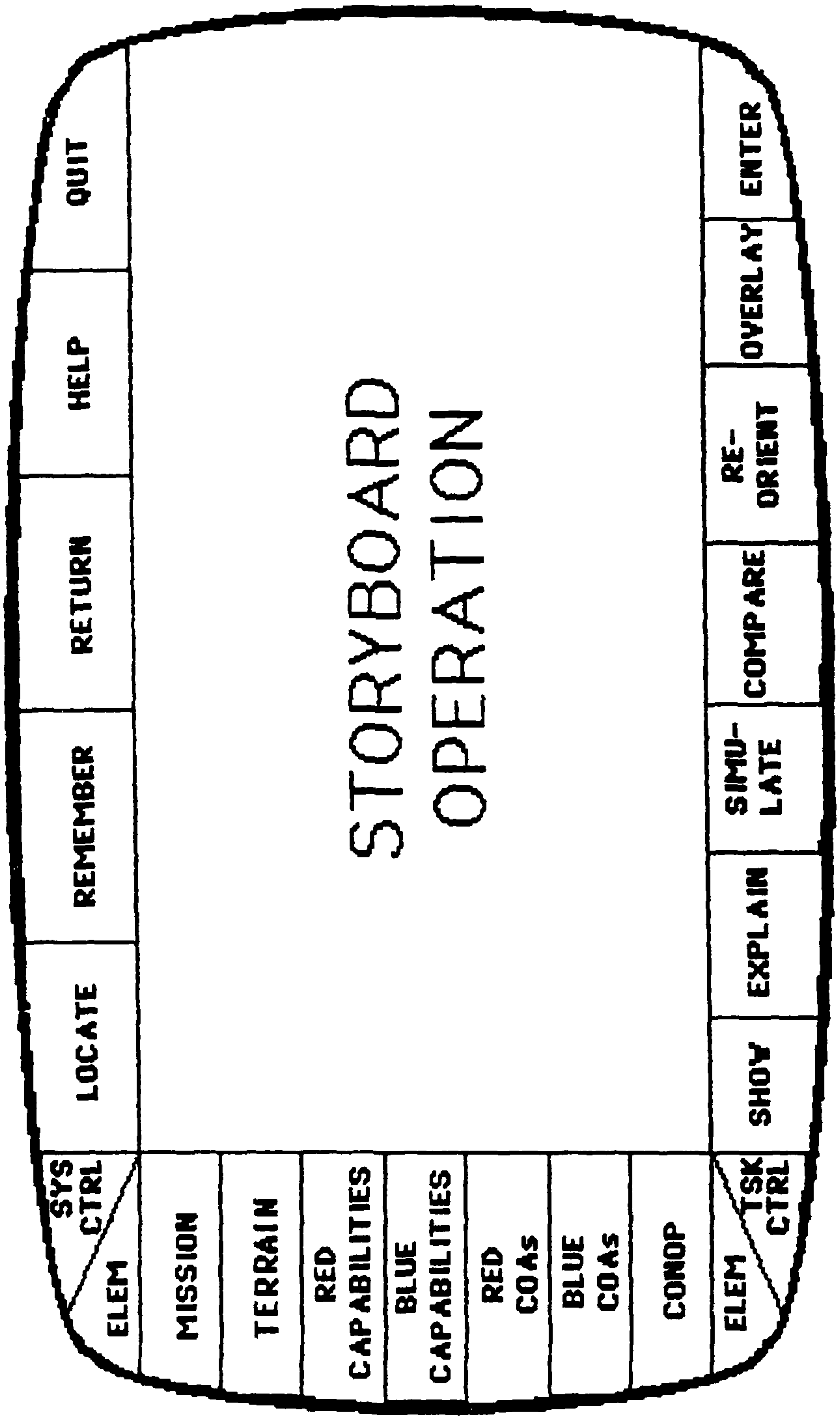
We believe that "naive" computer users will benefit from stationary on-screen menus. We believe that naive users will not invest much time in learning how to use a complex system. We believe that users want ease-of-use, intuitive command structures, and -- perhaps most of all -- "cognitive compatibility." We believe that users should not concentrate on how the system behaves but on what it is intended to do. We

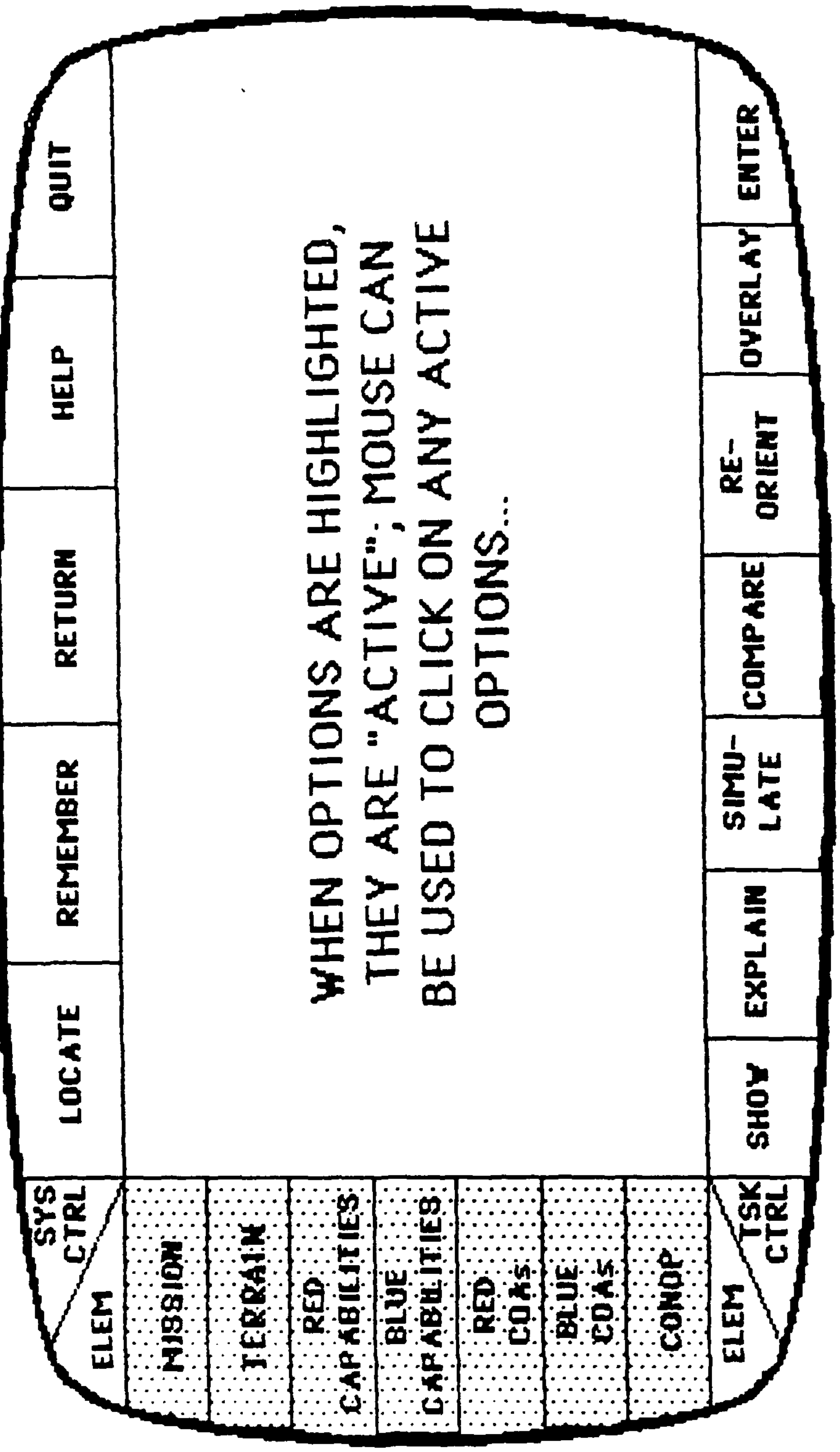

SYS CTRL
LOCATE
REMEMBER
RETURN
HELP
QUIT
ELEM
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

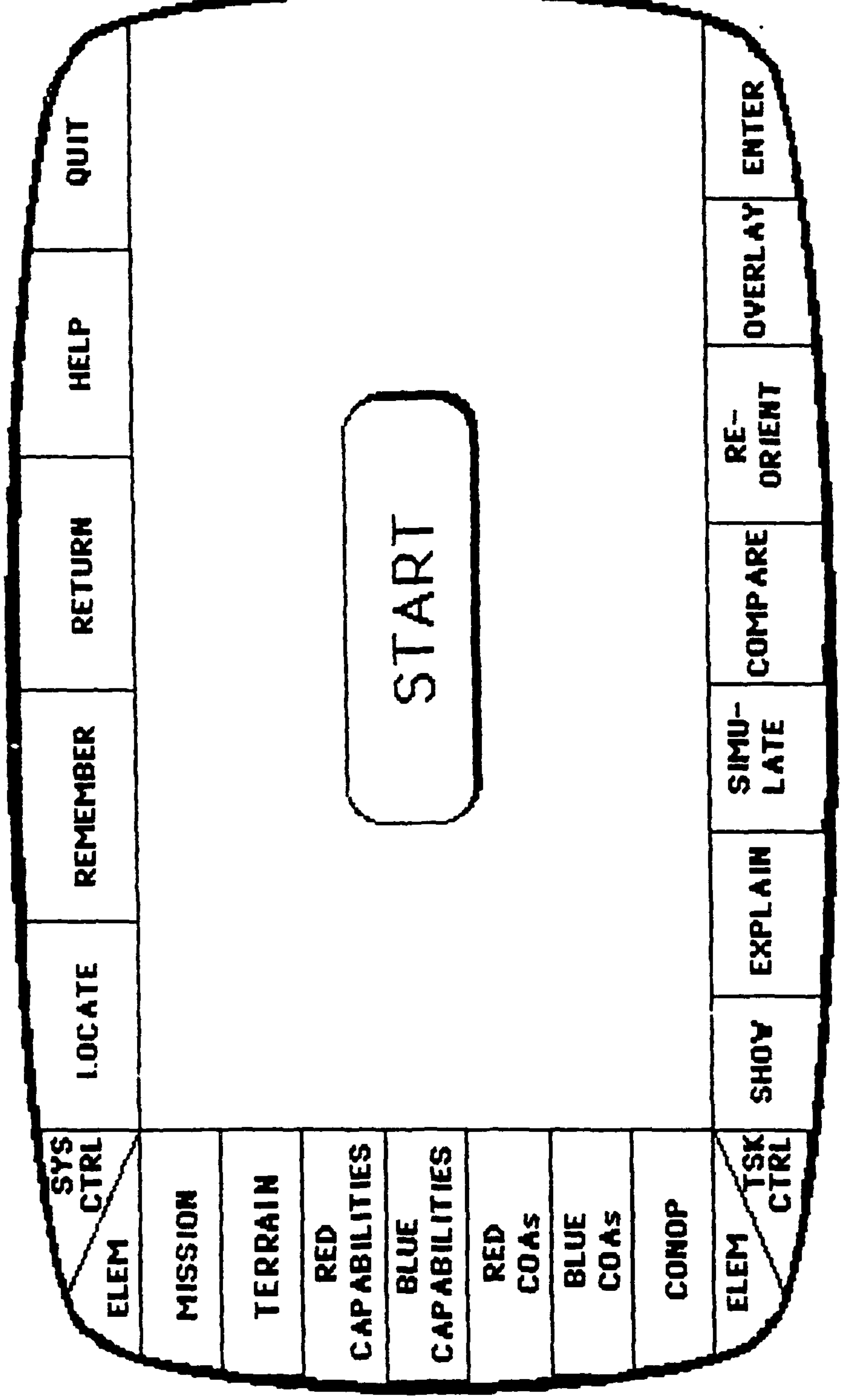
believe that users want to directly manipulate system functions and processes. We believe that planners would much prefer to "see and point" than "remember and type." We believe that the fields of "analytical graphics" and "visual cognition" hold great promise for UCI design. We believe that metaphors should be used to guide and direct user-computer interaction, and that analogs can play an important role in the problem-solving process. We also believe that all of these ideas will work within the domain of tactical planning and for the kinds of users and tasks the system concept is intended to serve.

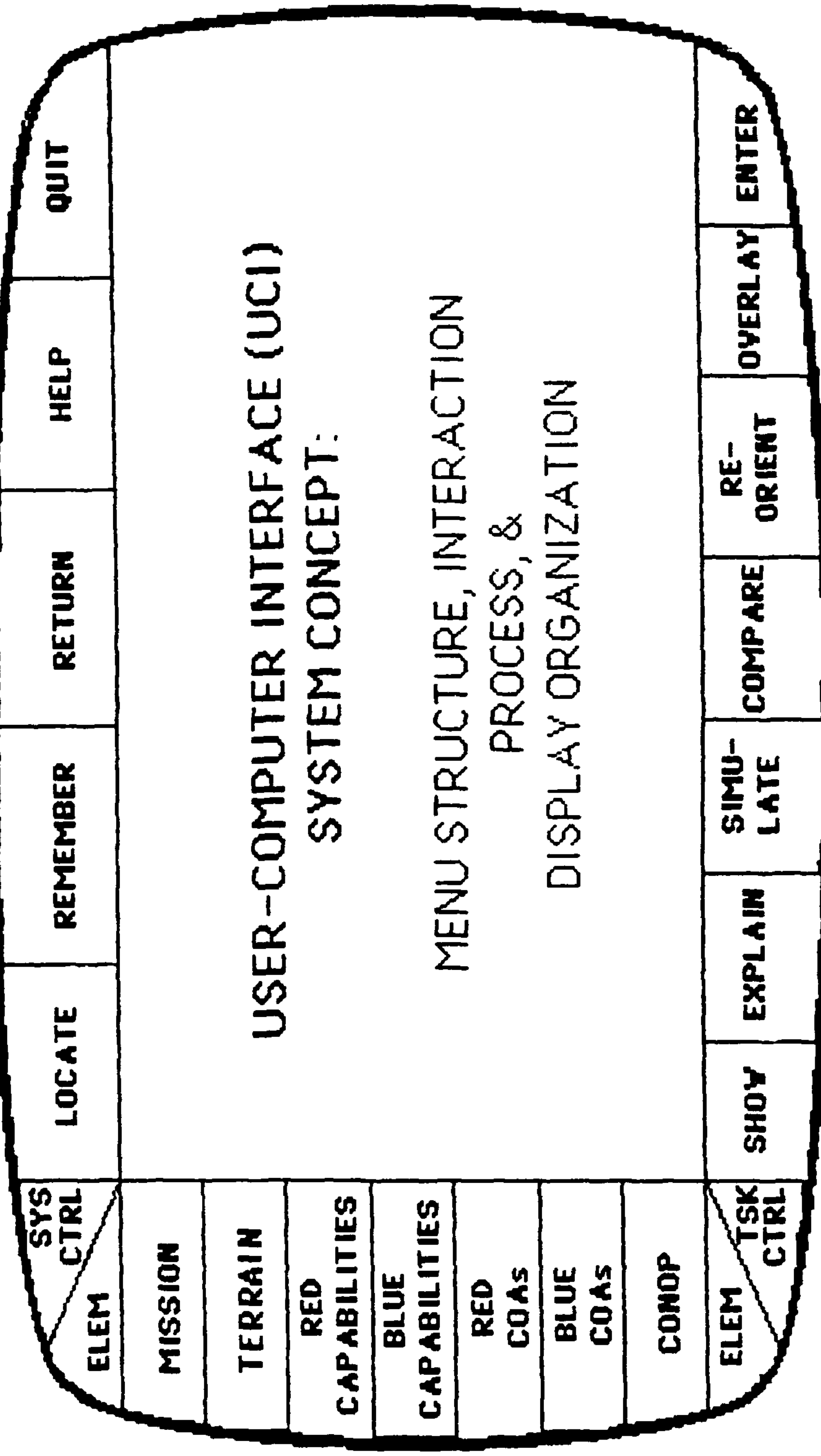
The figures that follow suggest our system concept derived from all of the above assumptions. We have designed an interface that is simultaneously "standard," flexible and powerful. We conceived of an interaction structure that would permit users to perform complex tasks, receive detailed data, information, and knowledge, and solve a specific problem without using a keyboard, without having to learn a command language, and without having to un-learn old or learn new problem-solving processes. (Past decision aids and support systems for tactical planning often required planners to learn a new methodology before operating the system -- a clear violation of the "compatibility" issue.)

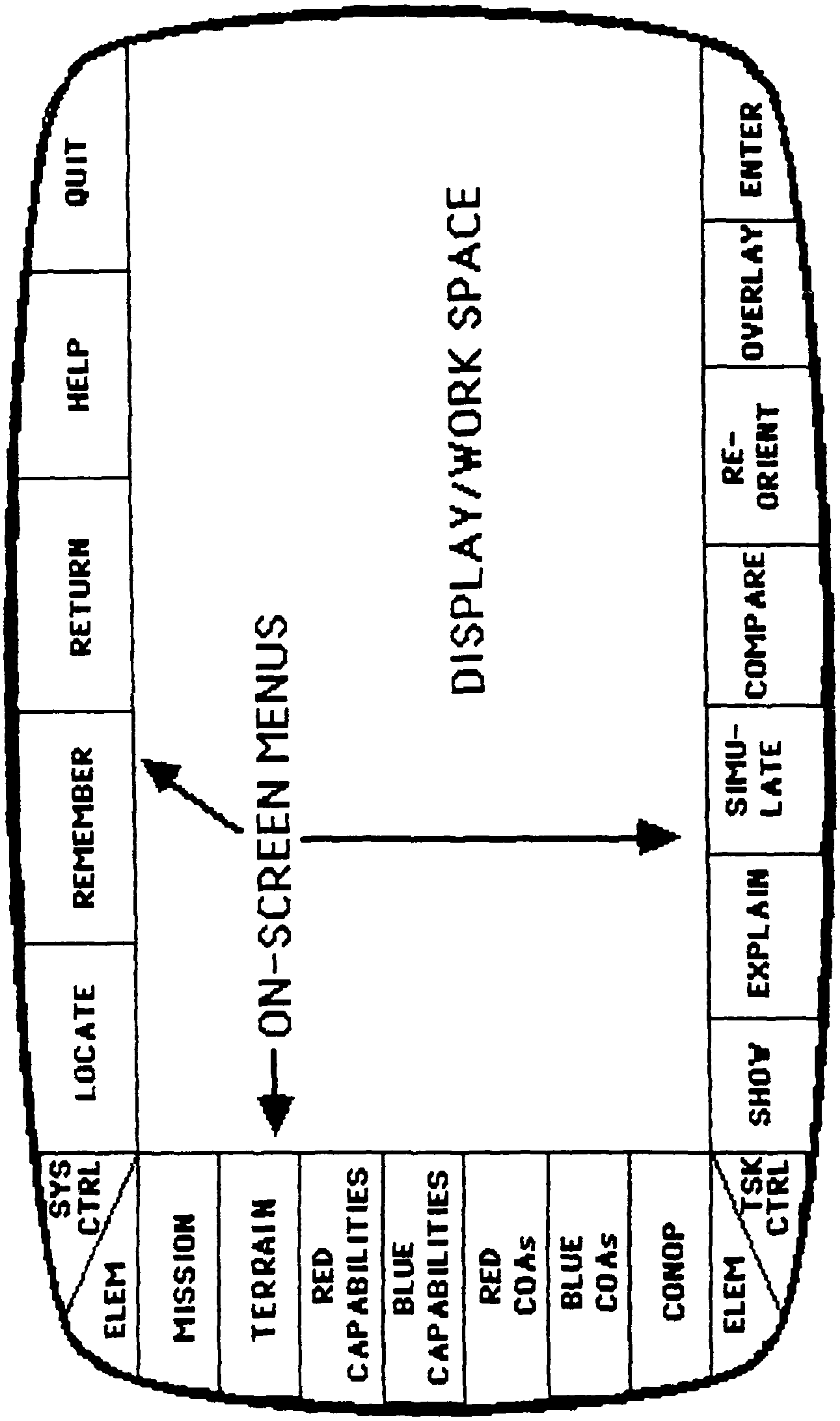
4.2.3 The Interactive Storyboard - The storyboard itself follows this section. Note that the displays can be run in sequence or randomly. The left-hand task options -- "Mission," "Terrain," "Red Capabilities," "Blue Capabilities," "Red Courses

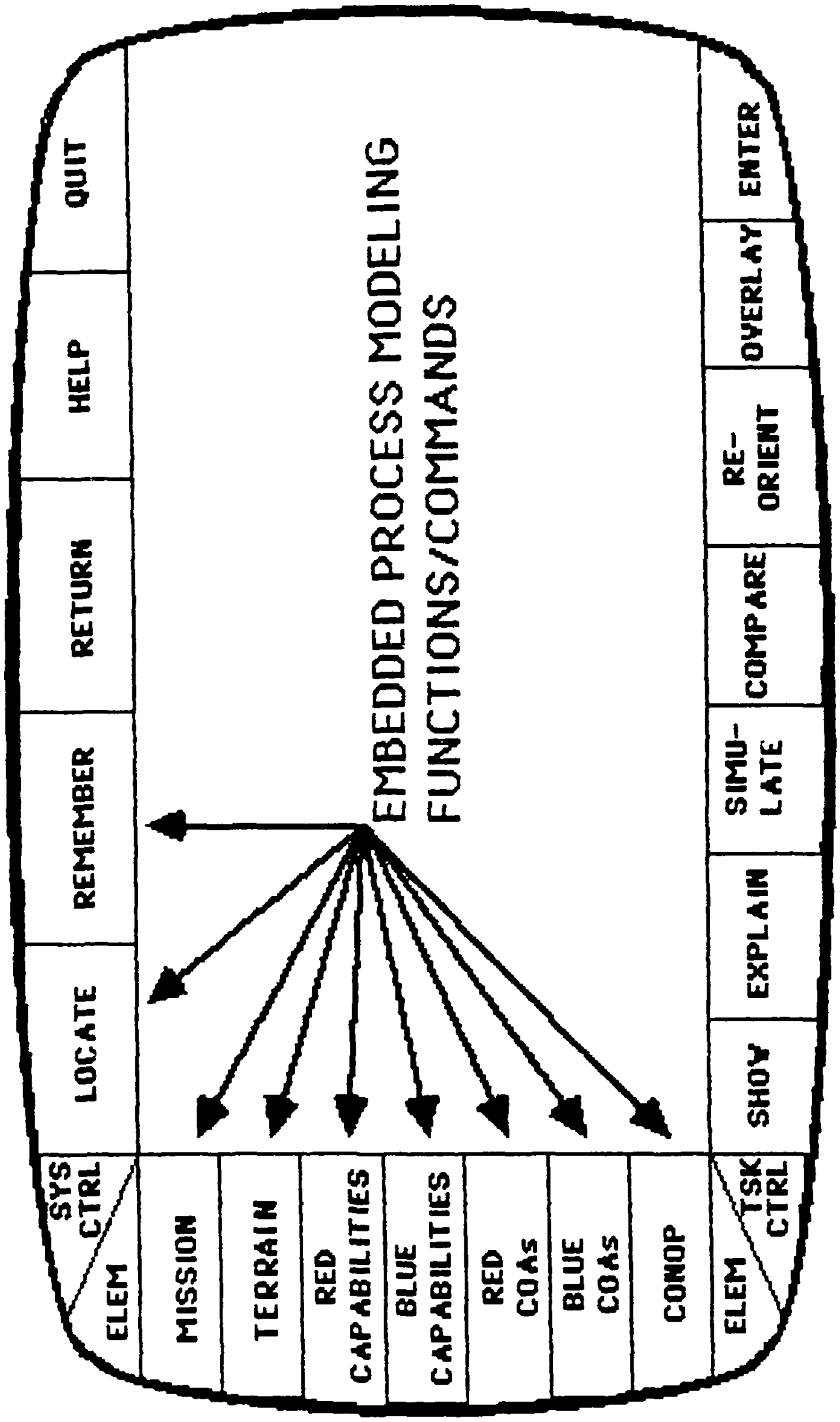

STORYBOARD OPERATION
SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
WHEN OPTIONS ARE HIGHLIGHTED, THEY ARE "ACTIVE"; MOUSE CAN BE USED TO CLICK ON ANY ACTIVE OPTIONS...
TSK CTRL
ELEM
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

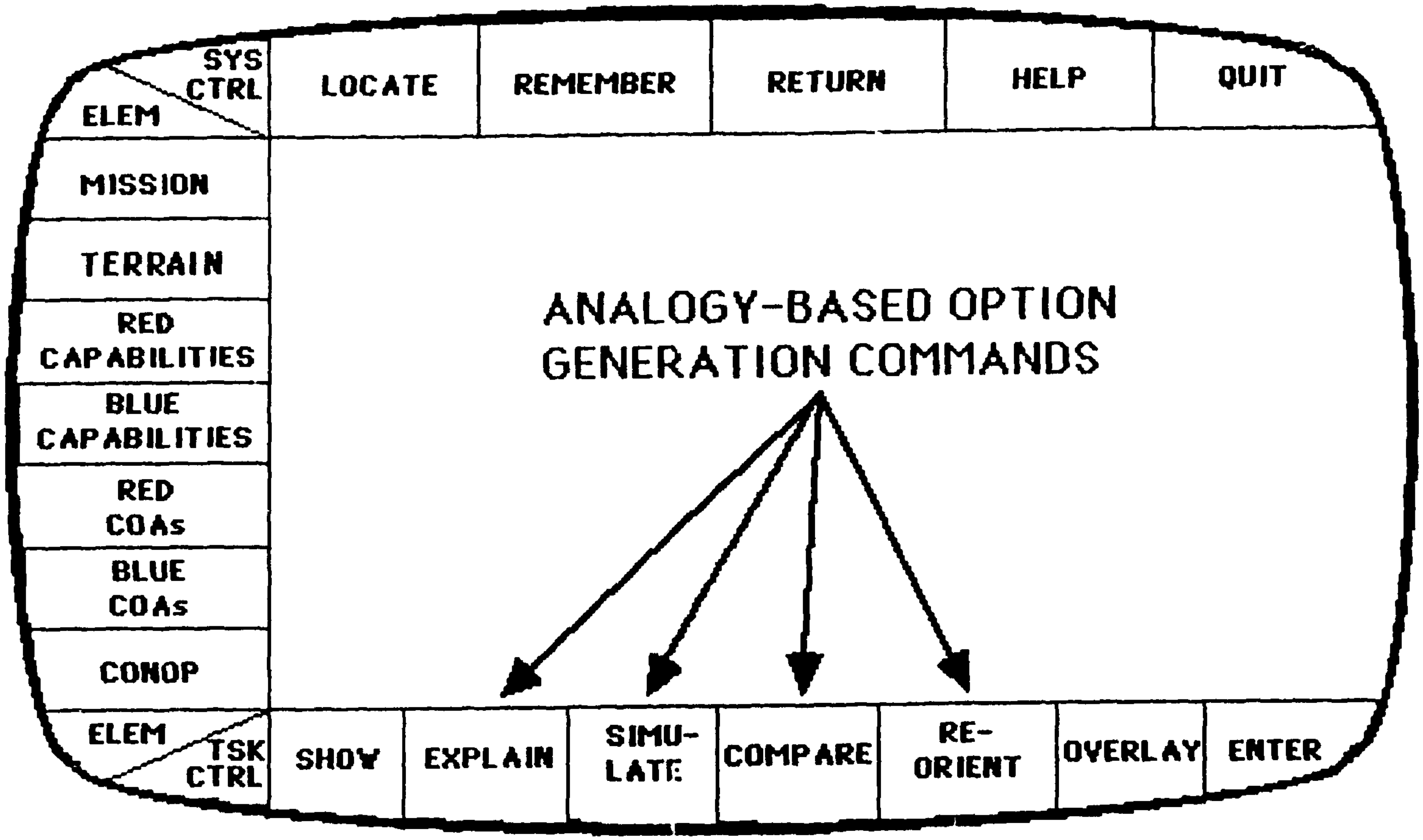

START
SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
TSK CTRL
ELEM
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

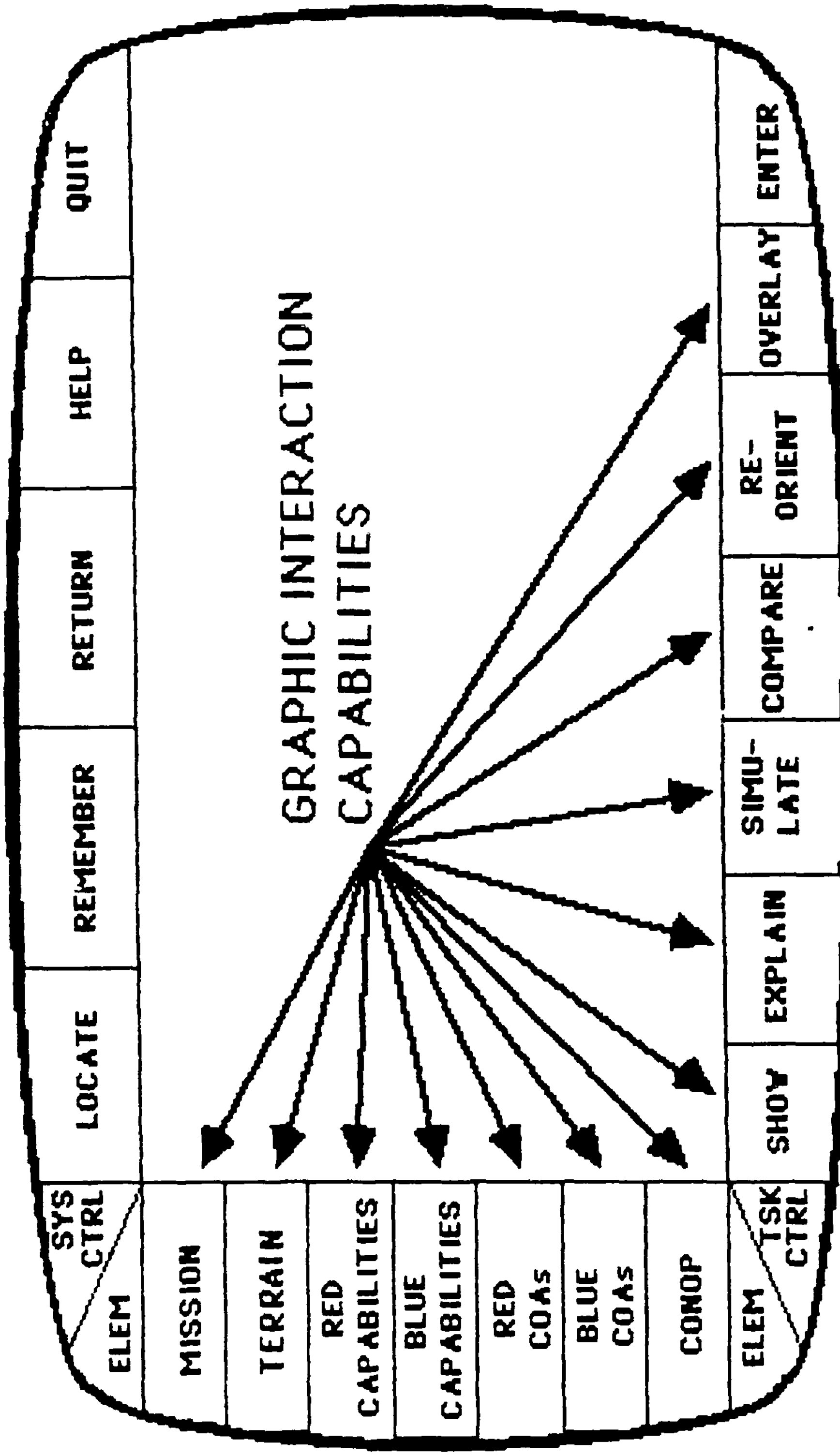

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER
USER-COMPUTER INTERFACE (UCI)
SYSTEM CONCEPT:
MENU STRUCTURE, INTERACTION
PROCESS, &
DISPLAY ORGANIZATION

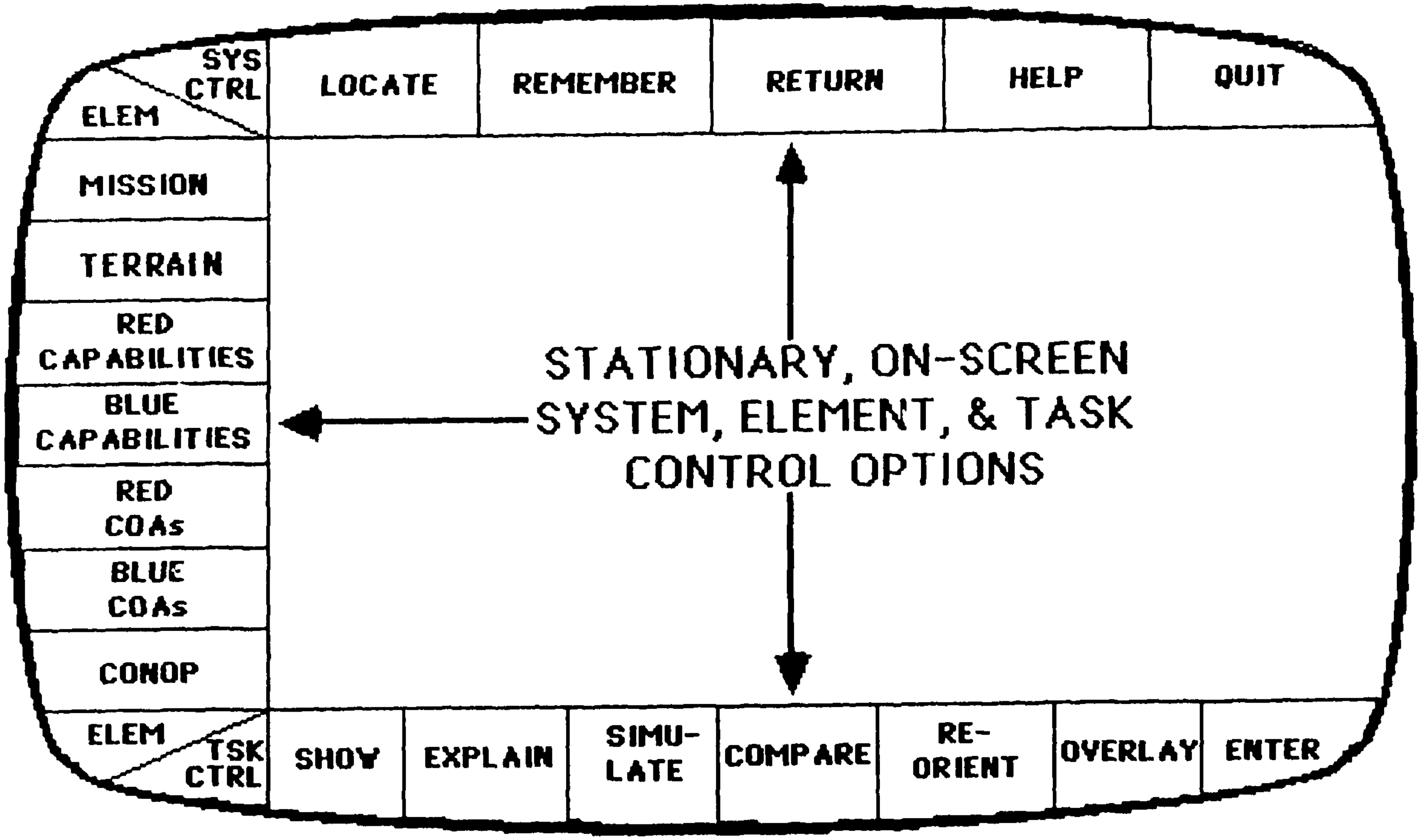

SYS CTRL
ELEM
QUIT
HELP
RETURN
REMEMBER
LOCATE
ON-SCREEN MENUS
DISPLAY/WORK SPACE
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
TSK CTRL
ELEM
ENTER
OVERLAY
RE-ORIENT
COMPARE
SIMU-LATE
EXPLAIN
SHOW

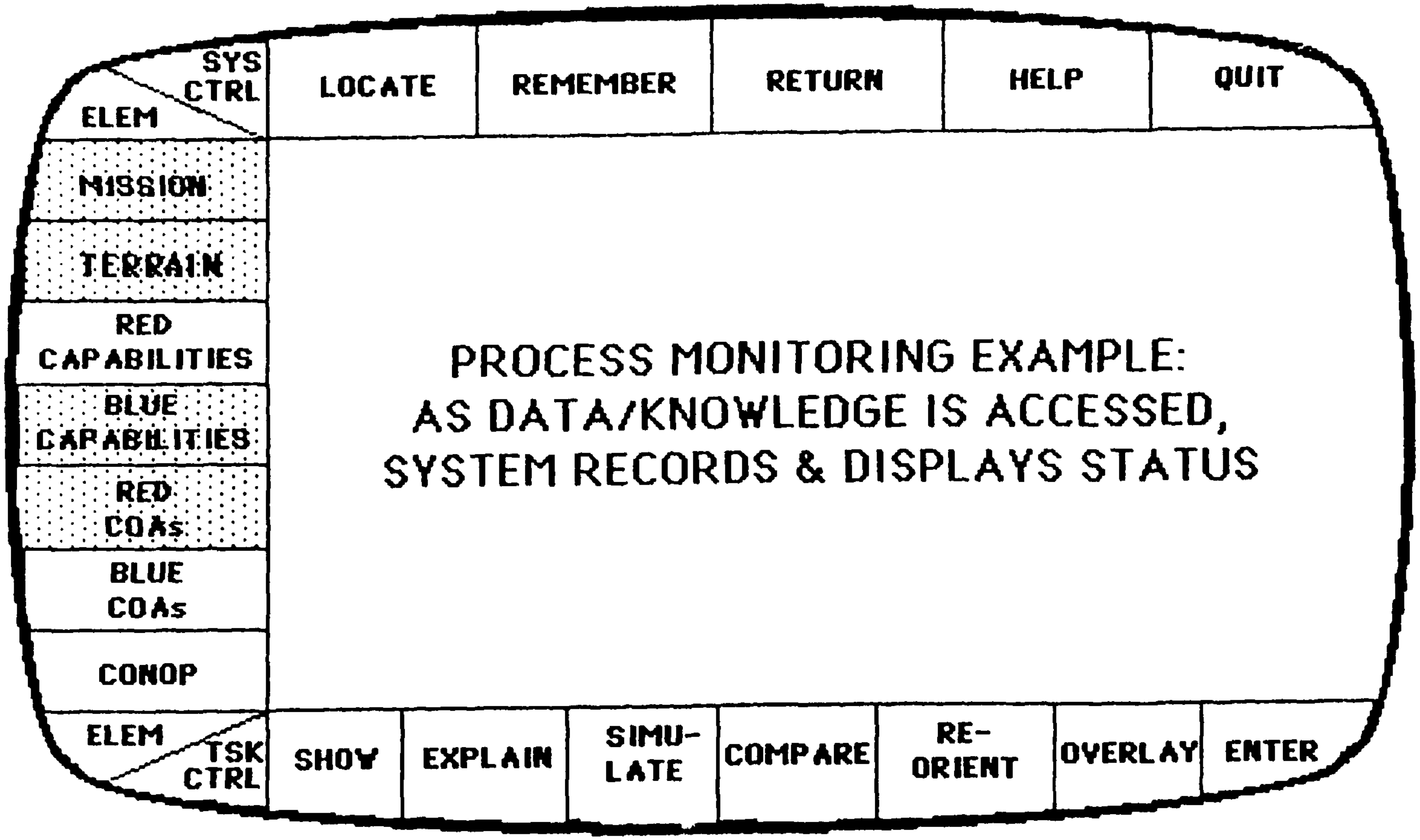


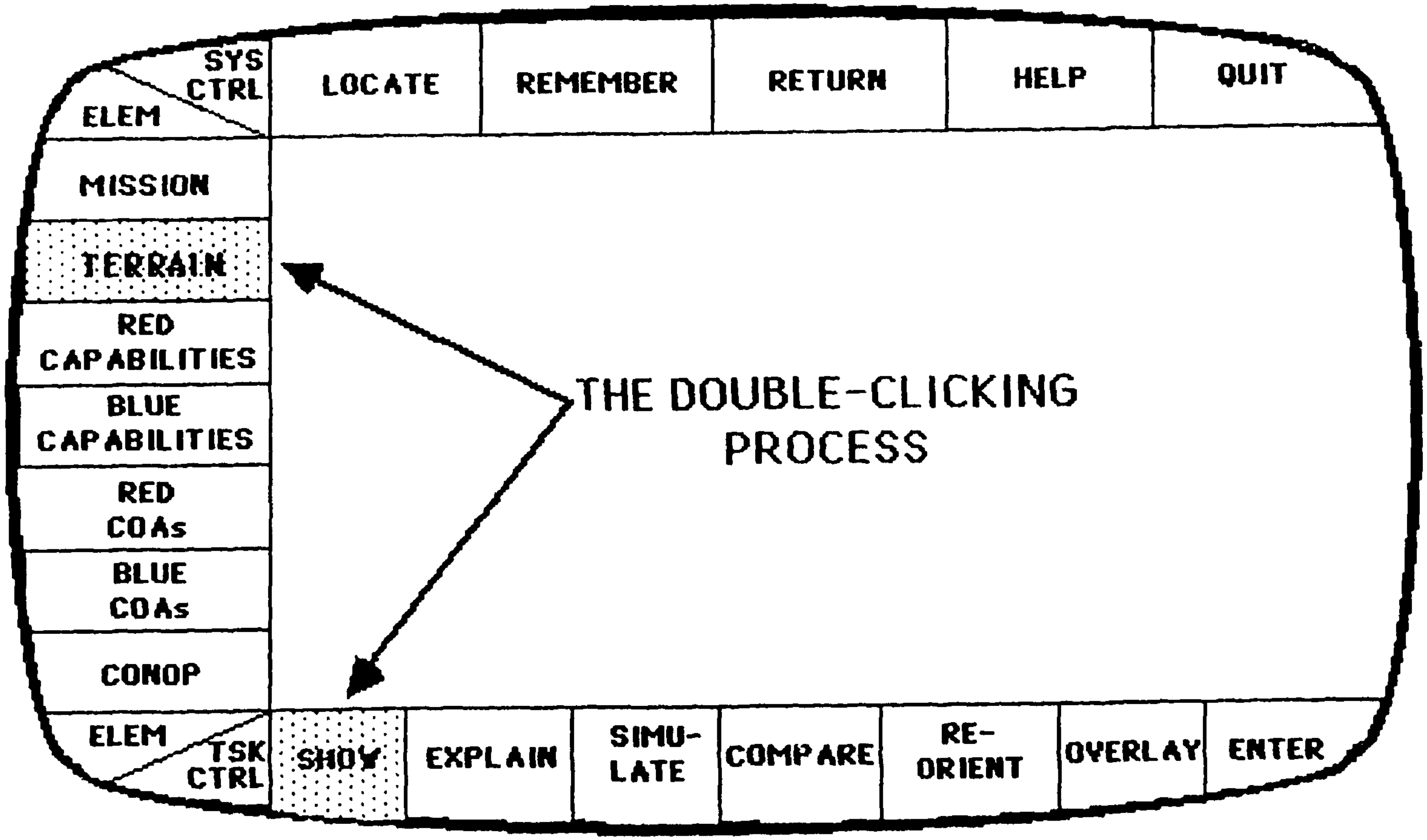
QUIT
HELP
RETURN
REMEMBER
LOCATE
SYS CTRL
ELEM
EMBEDDED PROCESS MODELING FUNCTIONS/COMMANDS
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

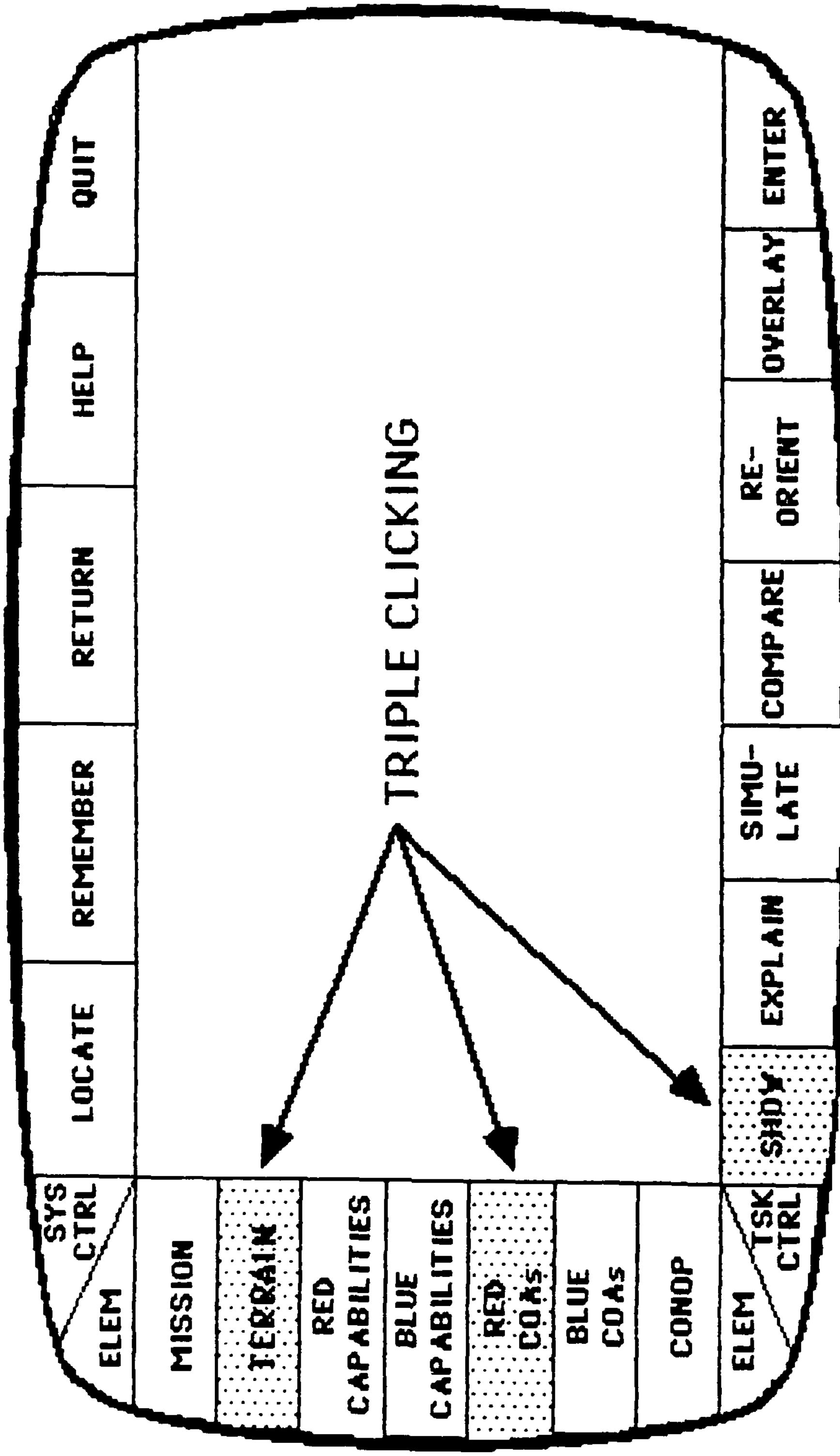

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ANALOGY-BASED OPTION GENERATION COMMANDS
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
GRAPHIC INTERACTION CAPABILITIES
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

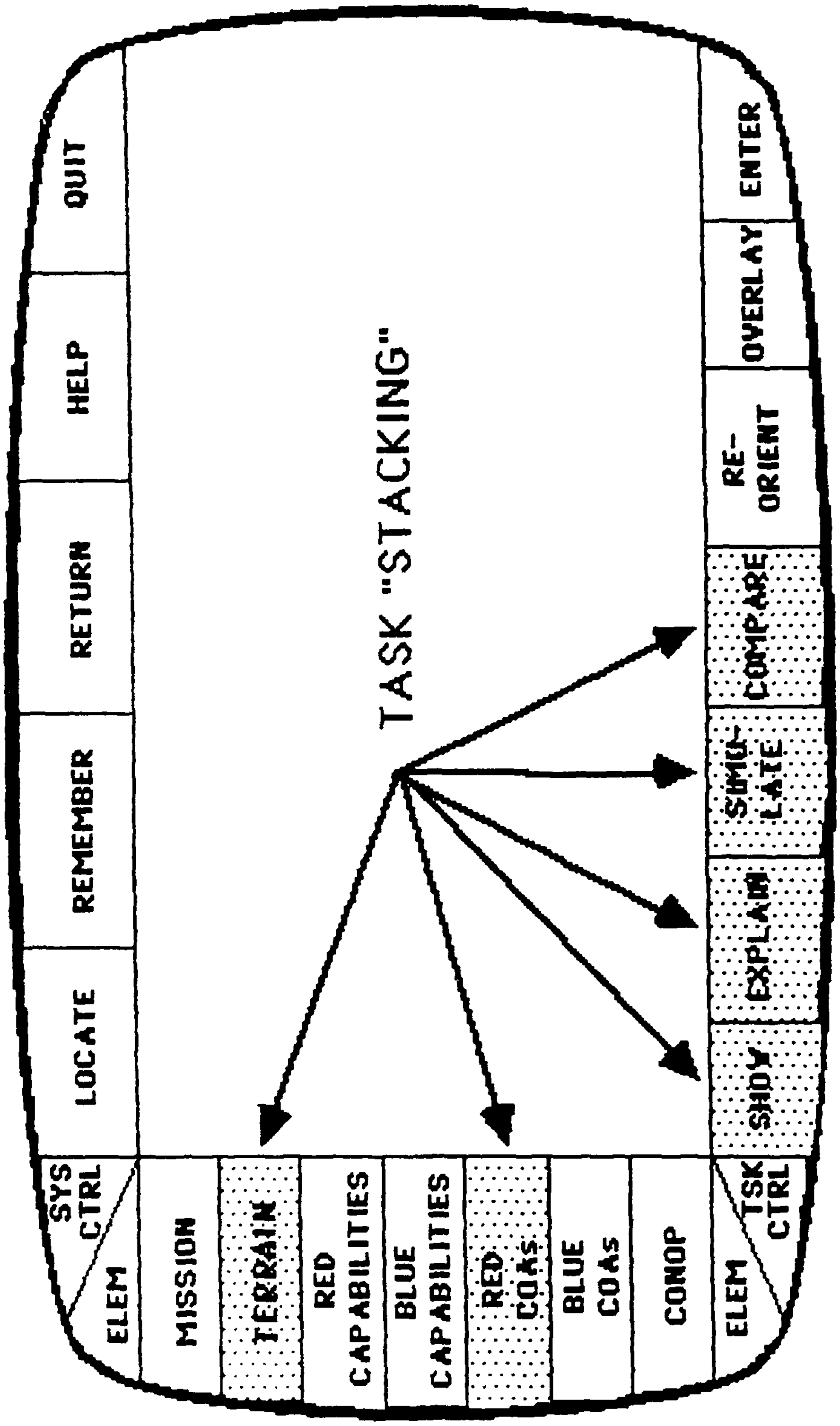

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
STATIONARY, ON-SCREEN
SYSTEM, ELEMENT, & TASK
CONTROL OPTIONS
SHOW
EXPLAIN
SIMU-
LATE
COMPARE
RE-
ORIENT
OVERLAY
ENTER


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
PROCESS MONITORING EXAMPLE:
AS DATA/KNOWLEDGE IS ACCESSED,
SYSTEM RECORDS & DISPLAYS STATUS
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

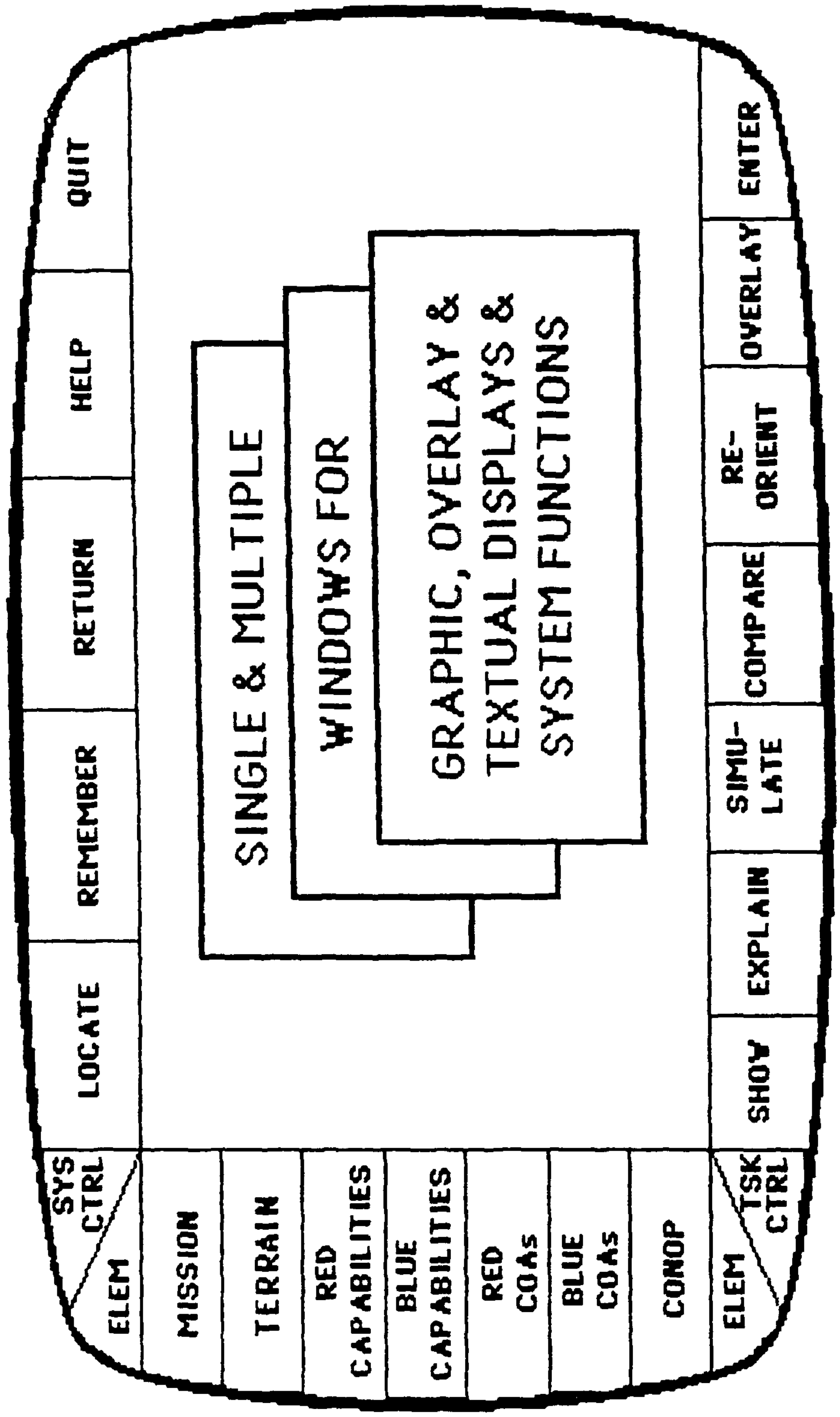

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER
THE DOUBLE-CLICKING PROCESS

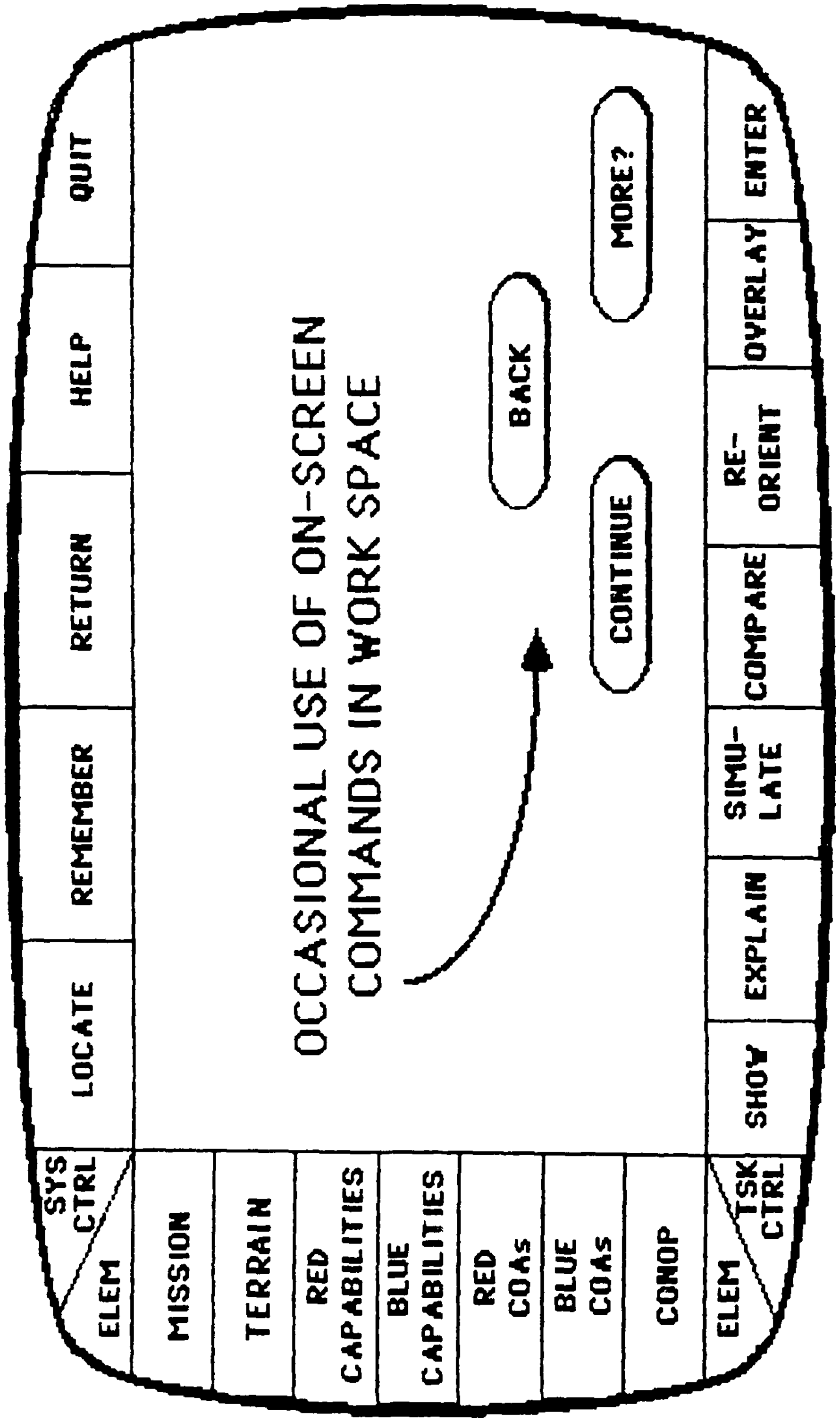


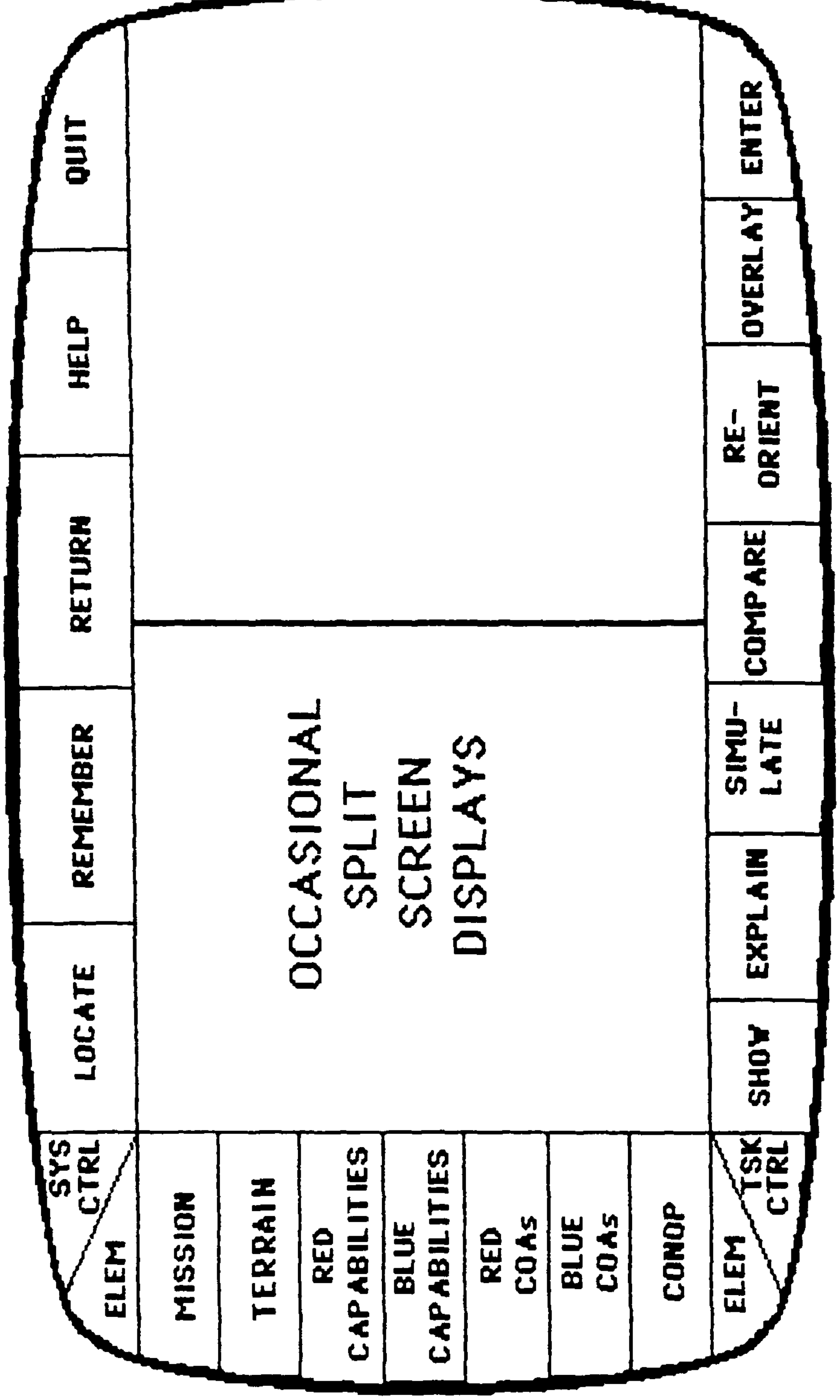
SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
TSK CTRL
ELEM
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER
TRIPLE CLICKING

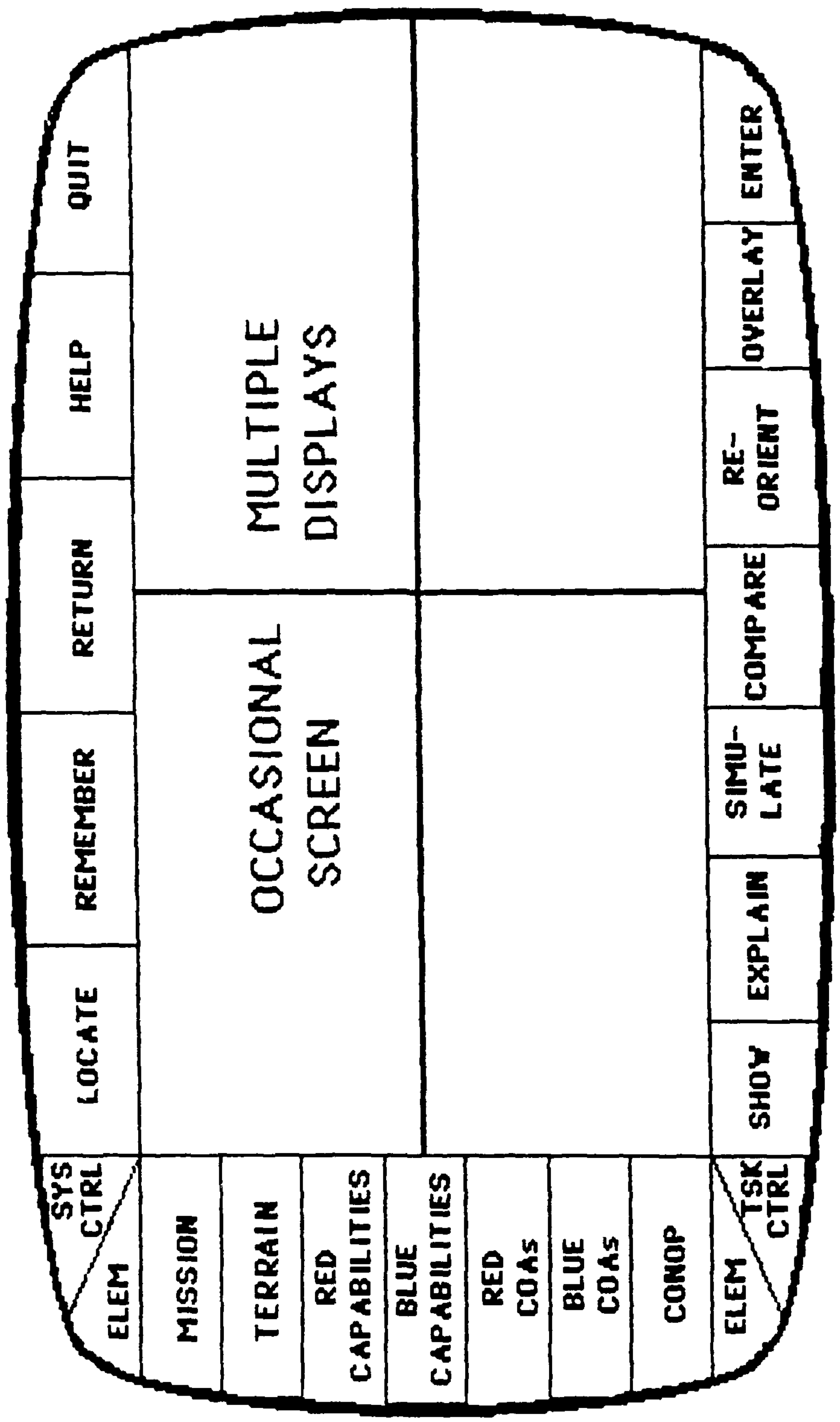


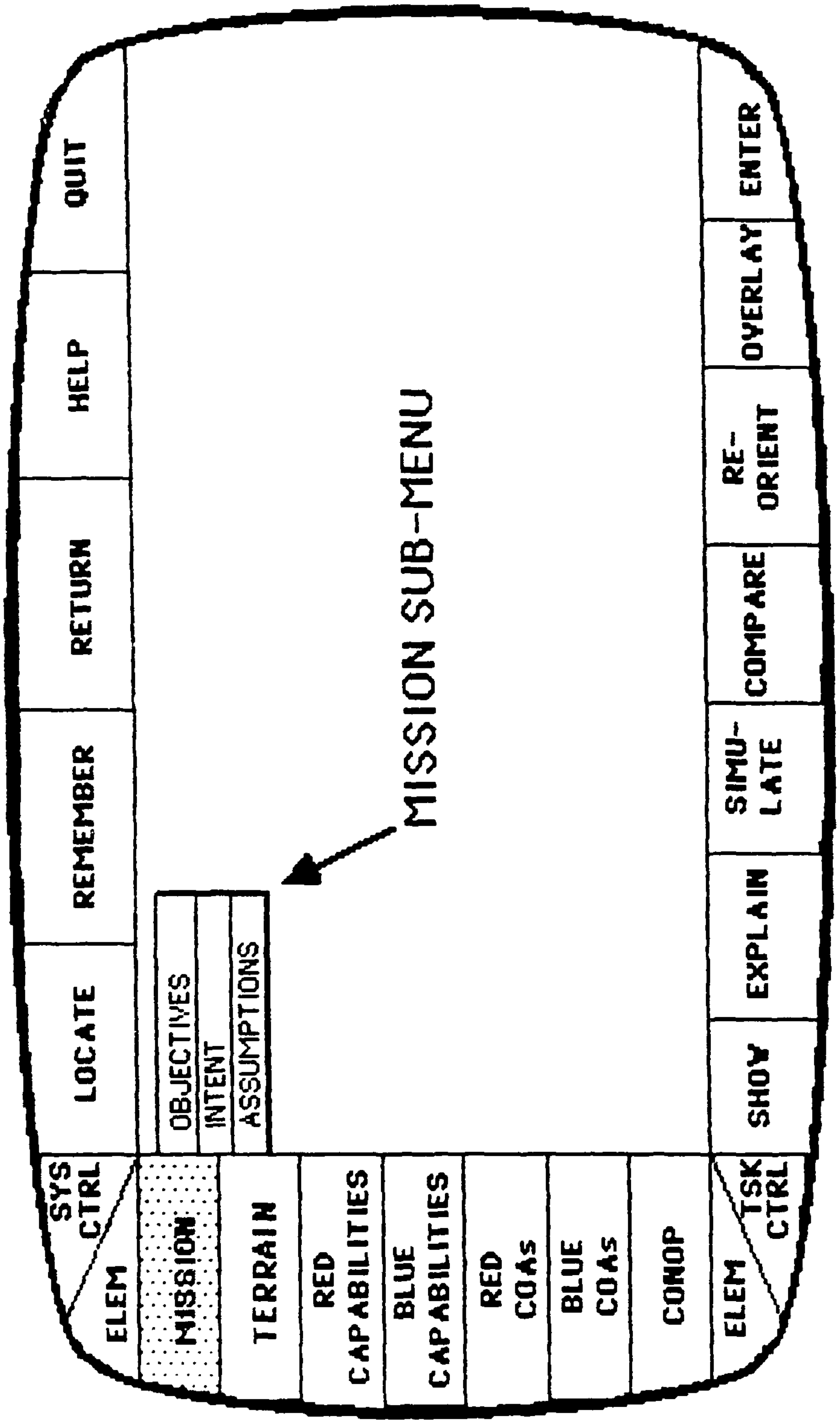
TASK "STACKING"
SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

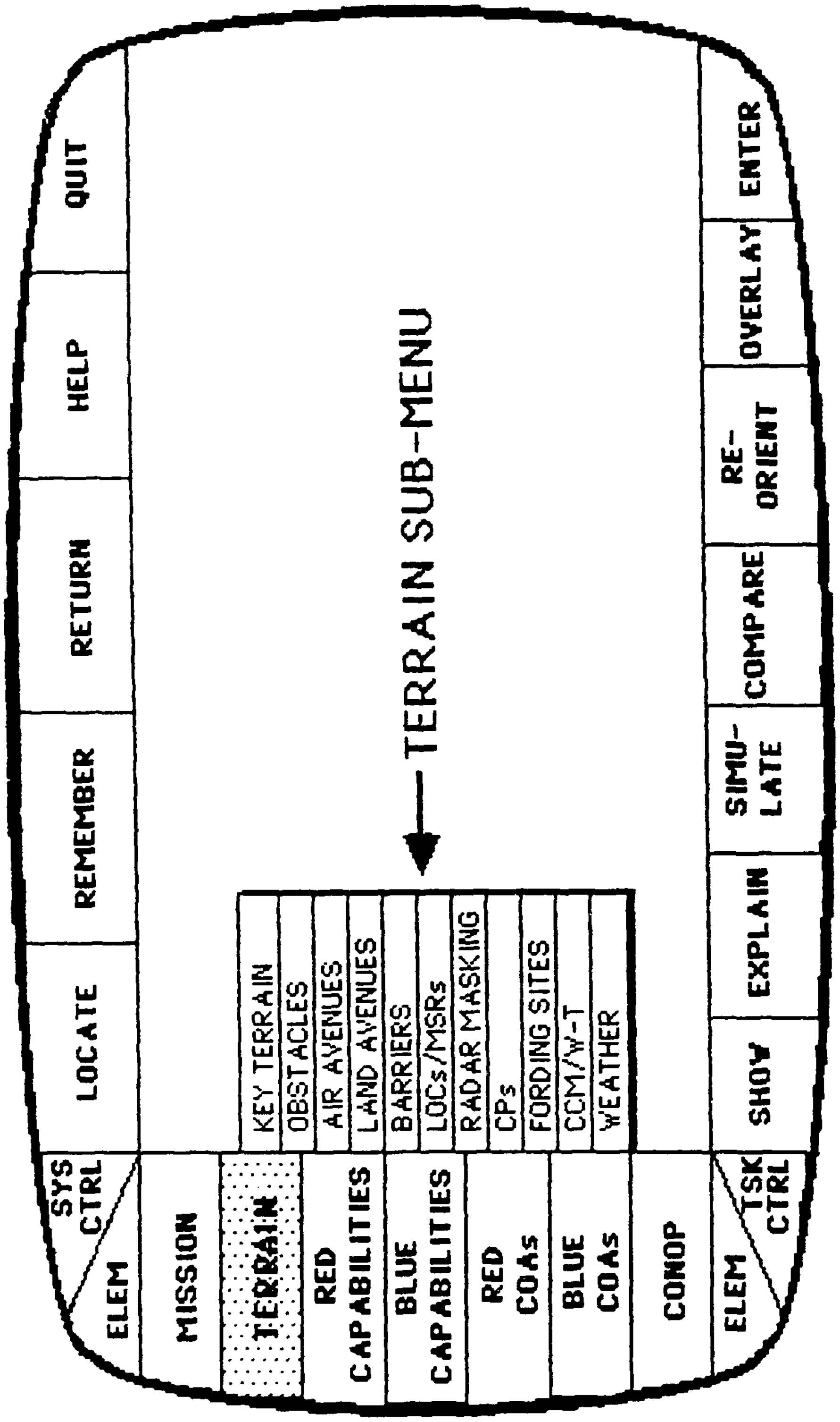

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
SINGLE & MULTIPLE
WINDOWS FOR
GRAPHIC, OVERLAY & TEXTUAL DISPLAYS & SYSTEM FUNCTIONS
TSK CTRL
ELEM
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

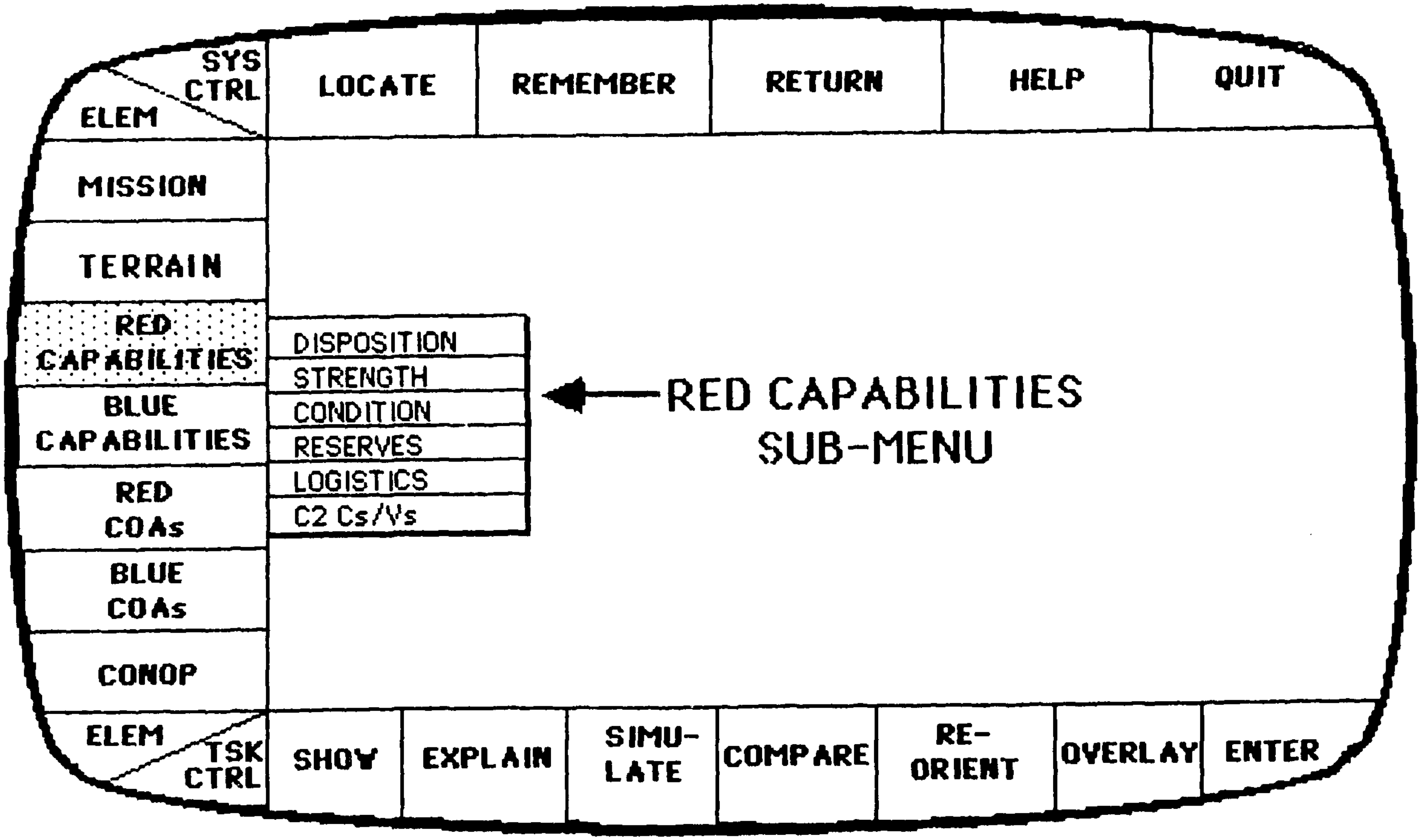

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
OCCASIONAL USE OF ON-SCREEN COMMANDS IN WORK SPACE
BACK
CONTINUE
MORE?
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

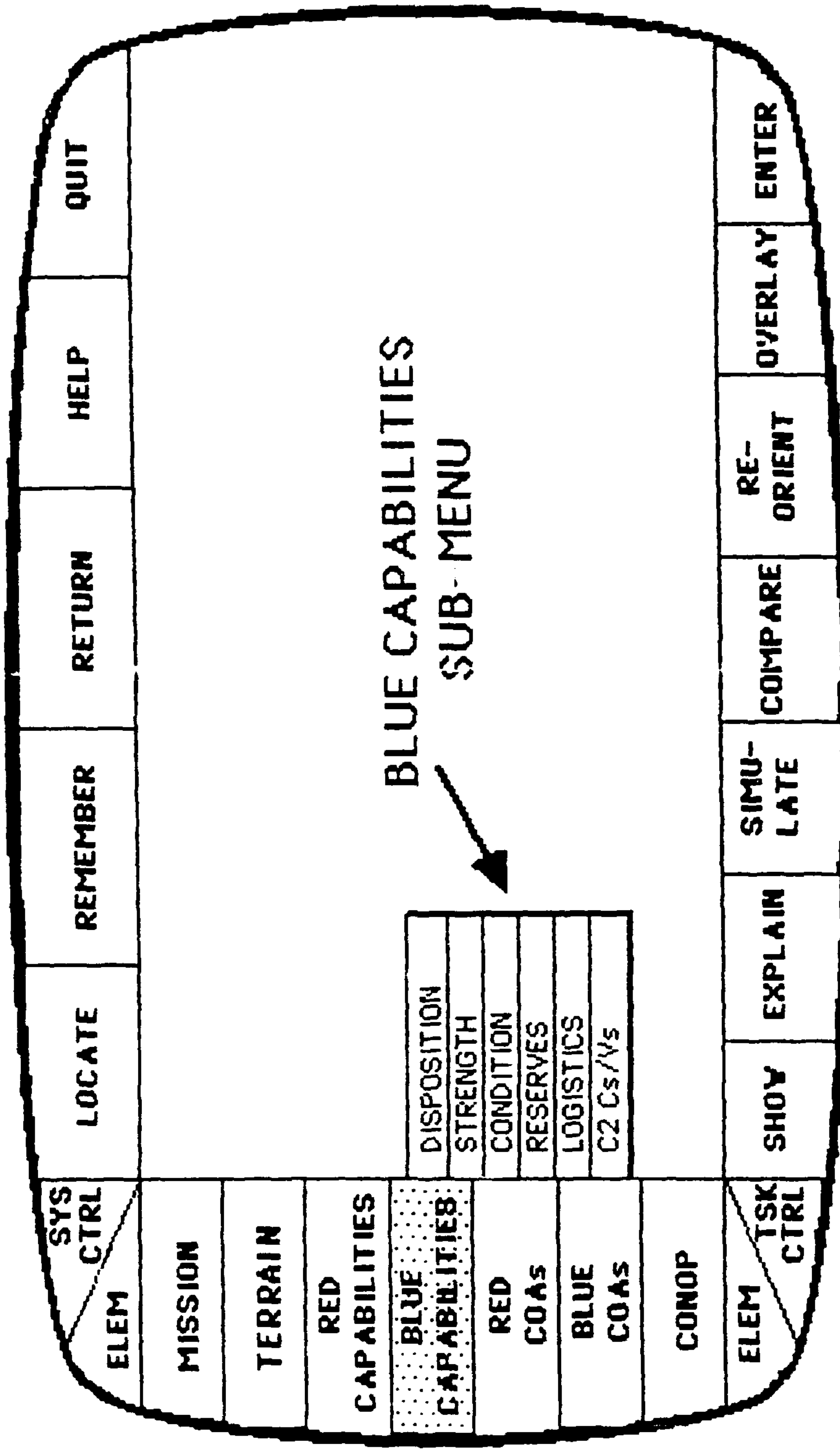

SYS CTRL
ELEM
QUIT
HELP
RETURN
REMEMBER
LOCATE
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
OCCASIONAL SPLIT SCREEN DISPLAYS
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU- LATE
COMPARE
RE- ORIENT
OVERLAY
ENTER


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MULTIPLE DISPLAYS
OCCASIONAL SCREEN
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

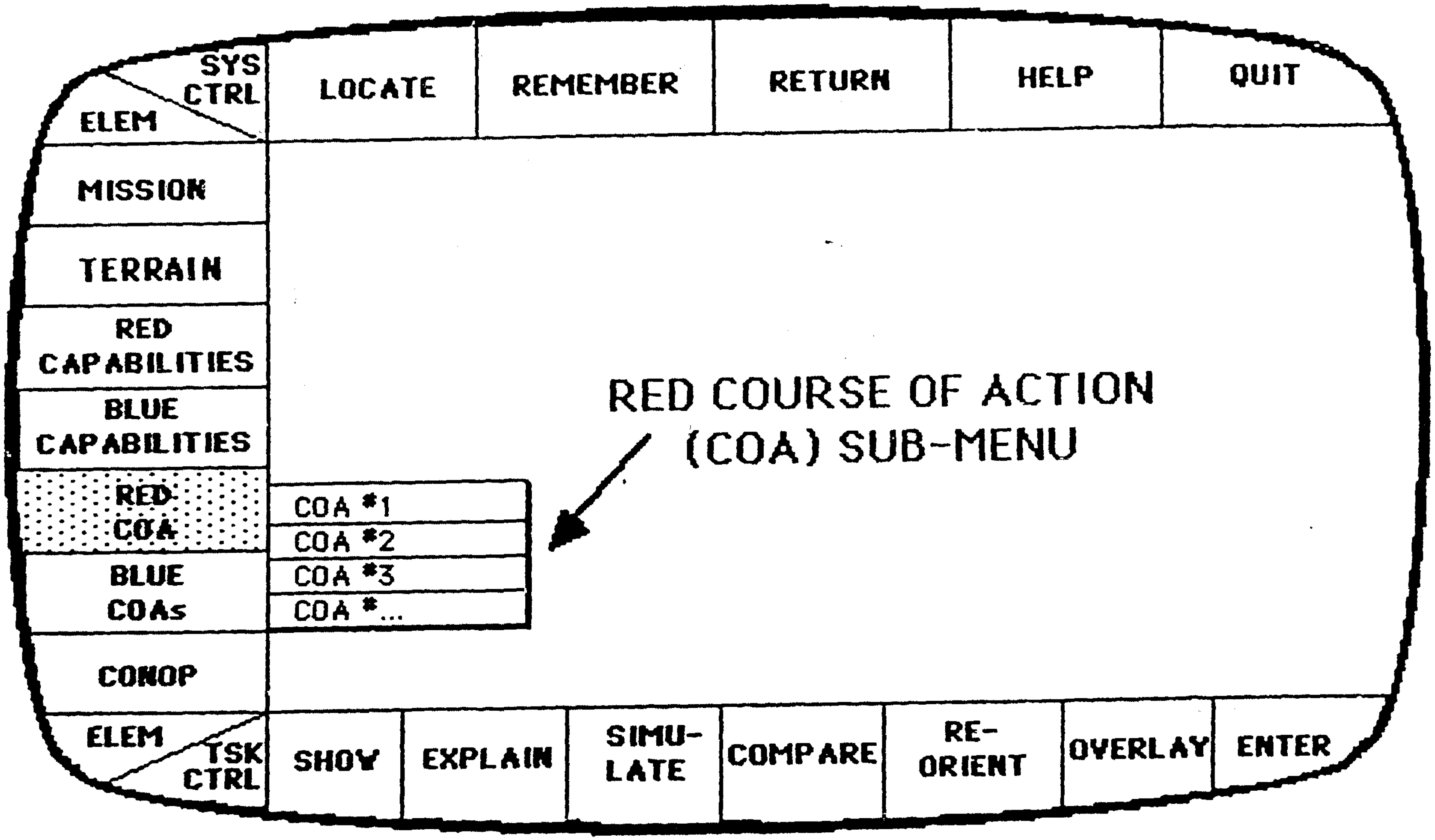

SYS CTRL
ELEM
QUIT
HELP
RETURN
REMEMBER
LOCATE
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
OBJECTIVES
INTENT
ASSUMPTIONS
MISSION SUB-MENU
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

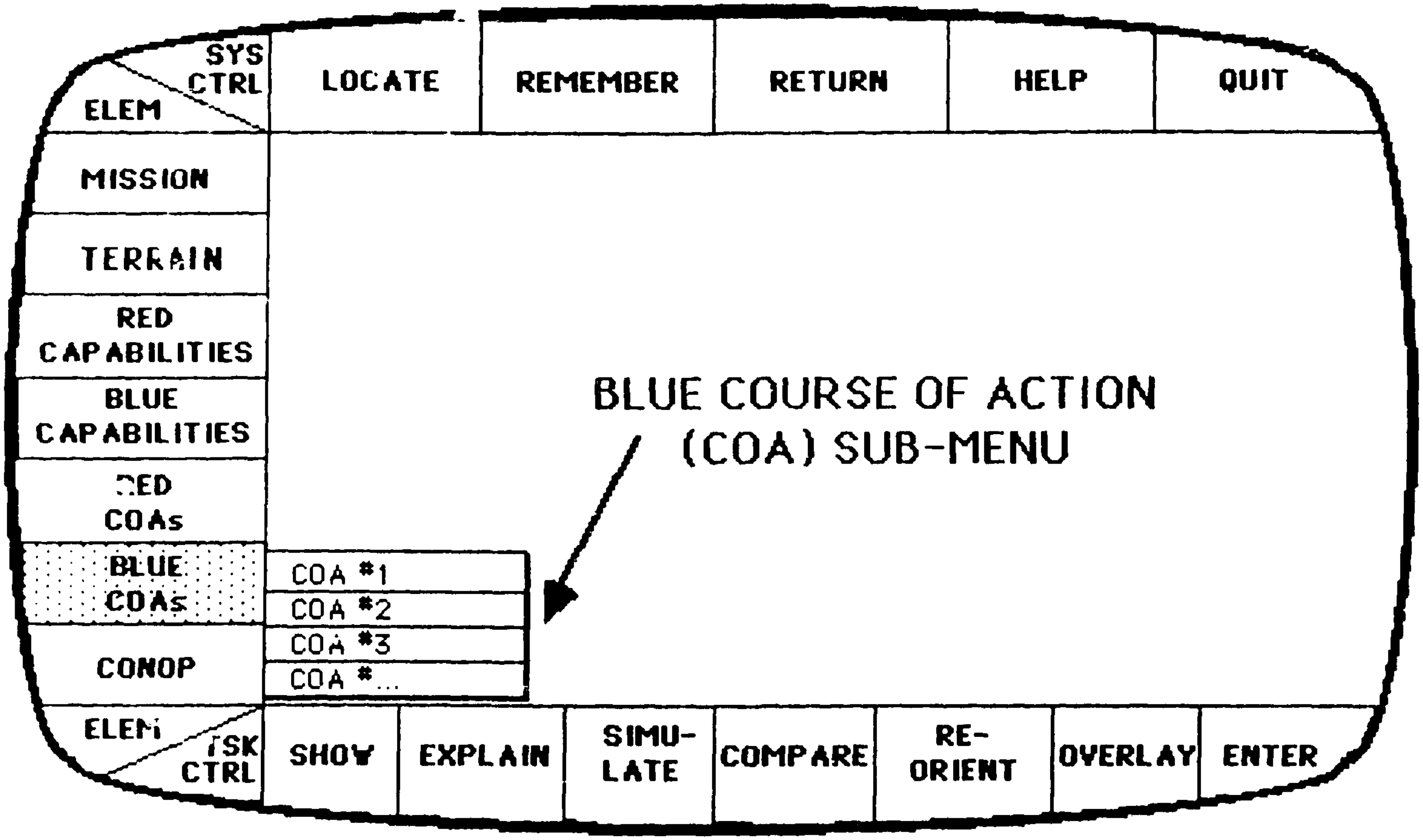

TERRAIN SUB-MENU
SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
KEY TERRAIN
OBSTACLES
AIR AVENUES
LAND AVENUES
BARRIERS
LOCs/MSRs
RADAR MASKING
CPs
FORDING SITES
CCM/W-T
WEATHER
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

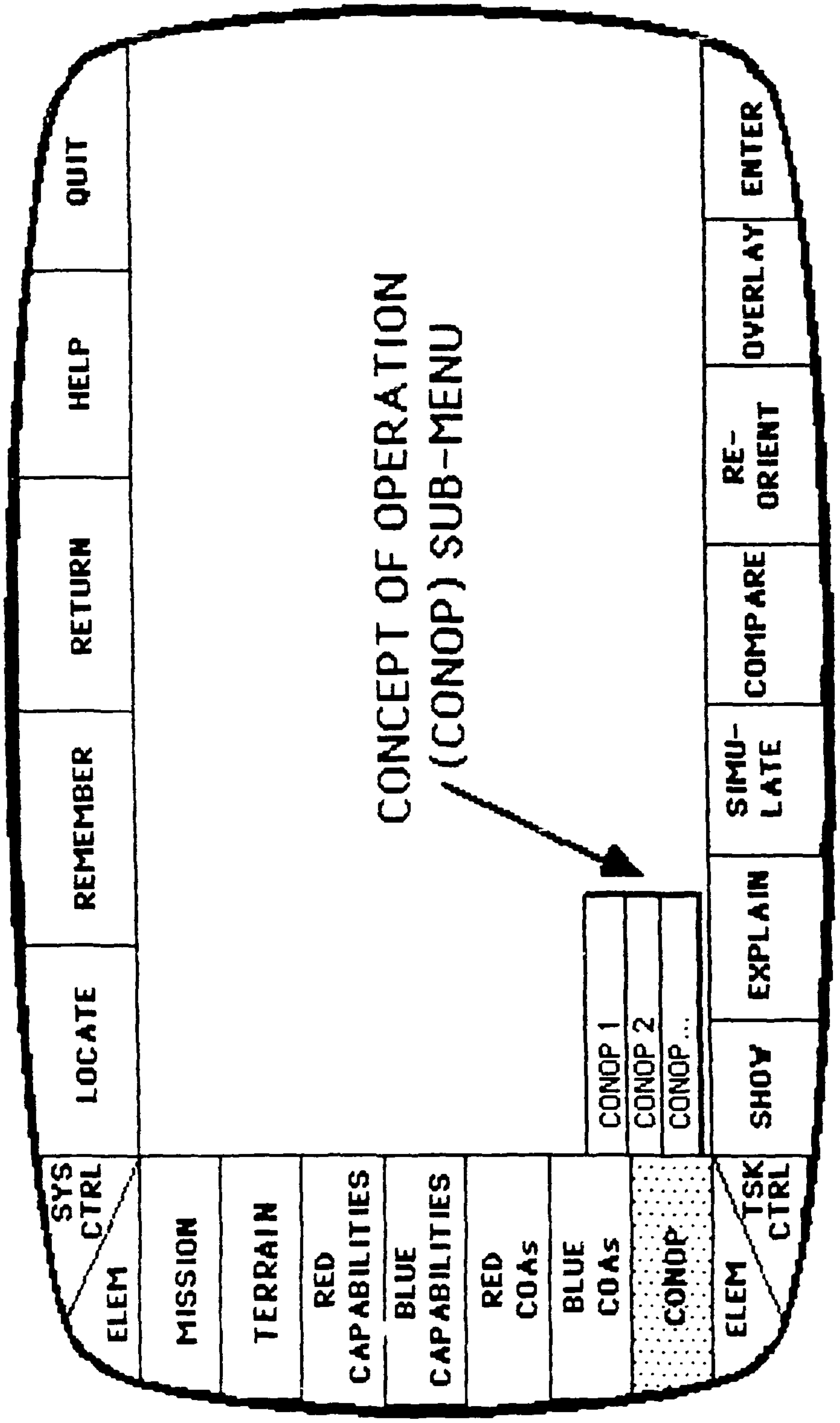

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
DISPOSITION
STRENGTH
CONDITION
RESERVES
LOGISTICS
C2 Cs/Vs
RED CAPABILITIES SUB-MENU
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER



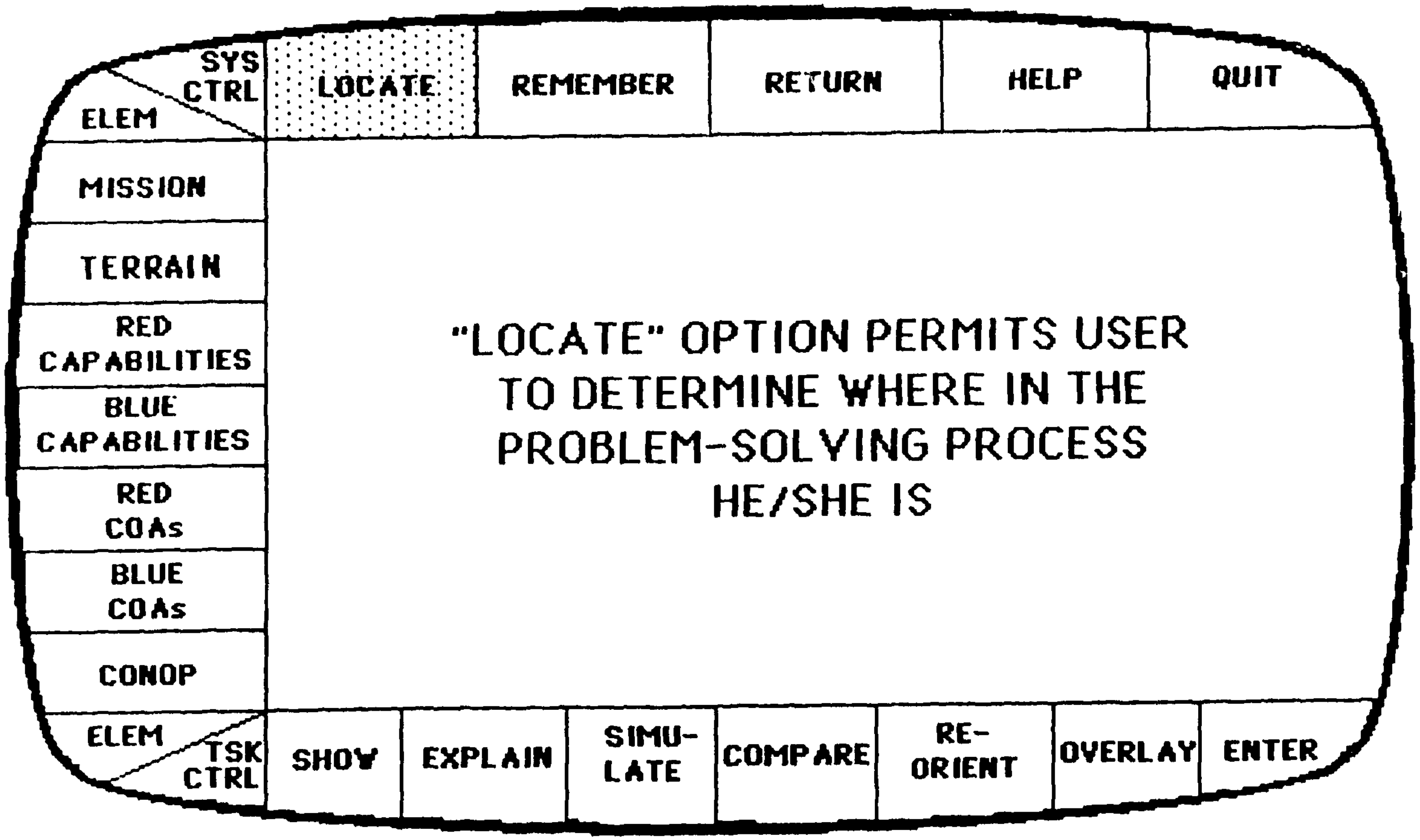
SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
DISPOSITION
STRENGTH
CONDITION
RESERVES
LOGISTICS
C2 Cs/Vs
BLUE CAPABILITIES SUB-MENU
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

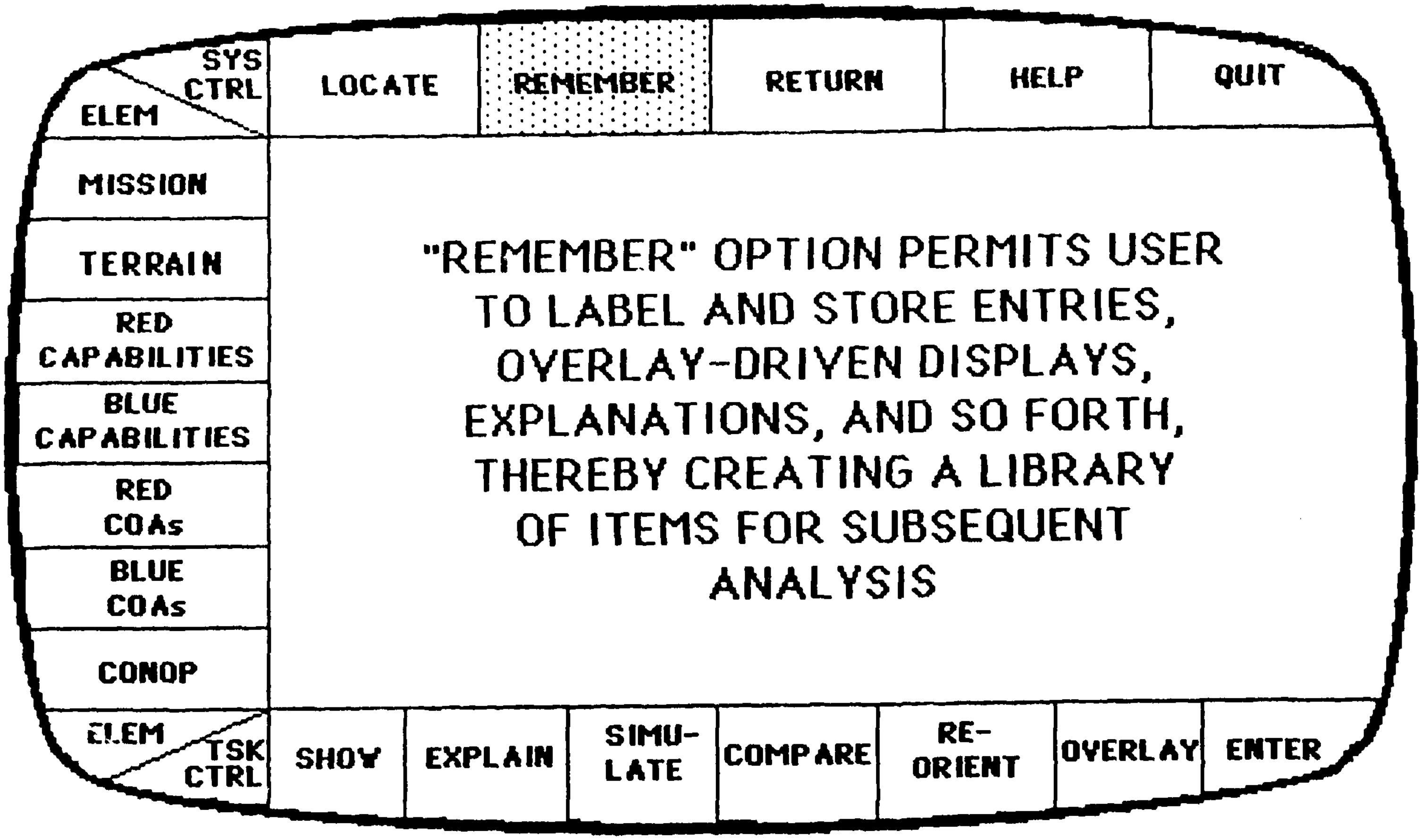

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COA
BLUE COAs
CONOP
ELEM
TSK CTRL
COA #1
COA #2
COA #3
COA #...
RED COURSE OF ACTION
(COA) SUB-MENU
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

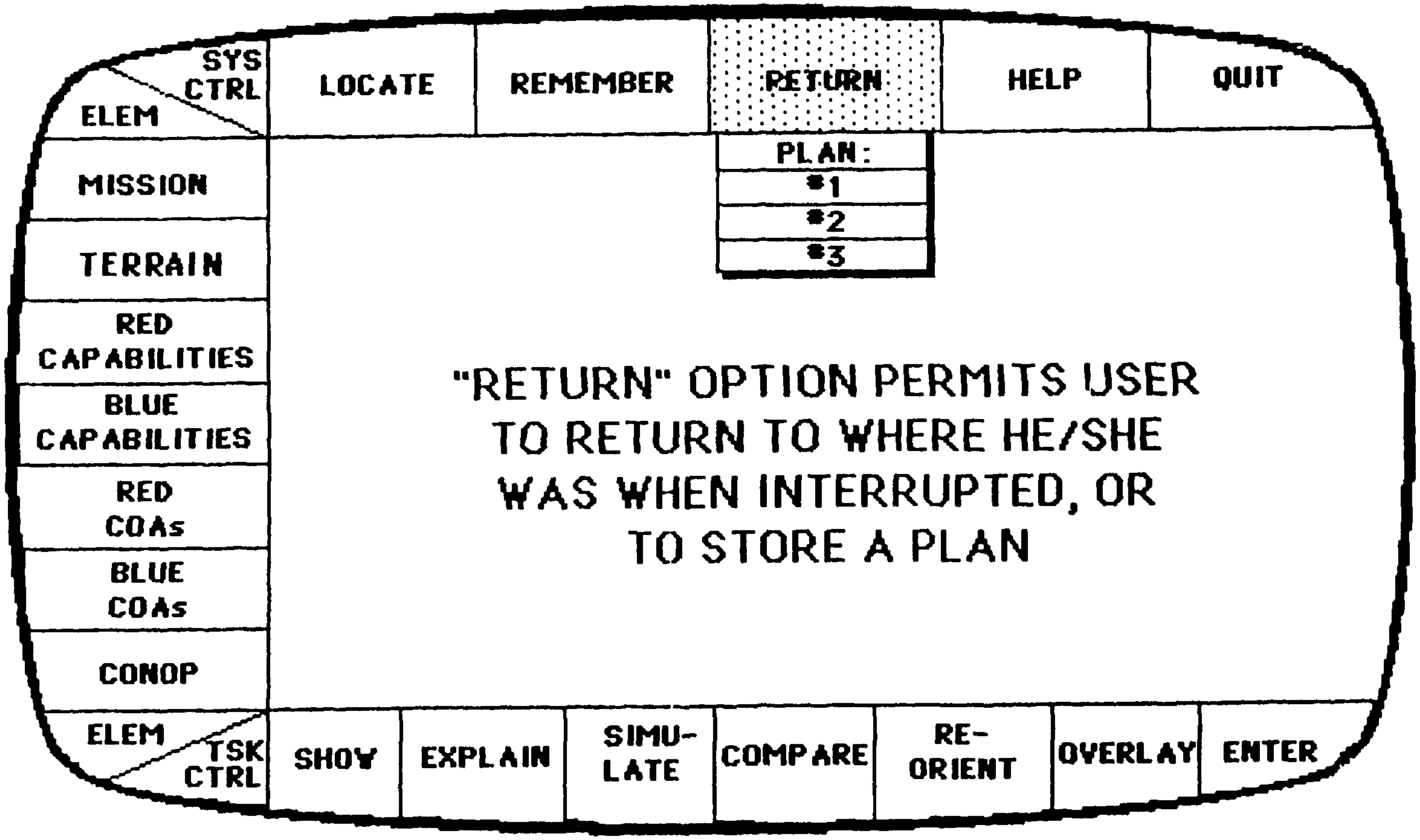

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
COA #1
COA #2
COA #3
COA #...
BLUE COURSE OF ACTION
(COA) SUB-MENU
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

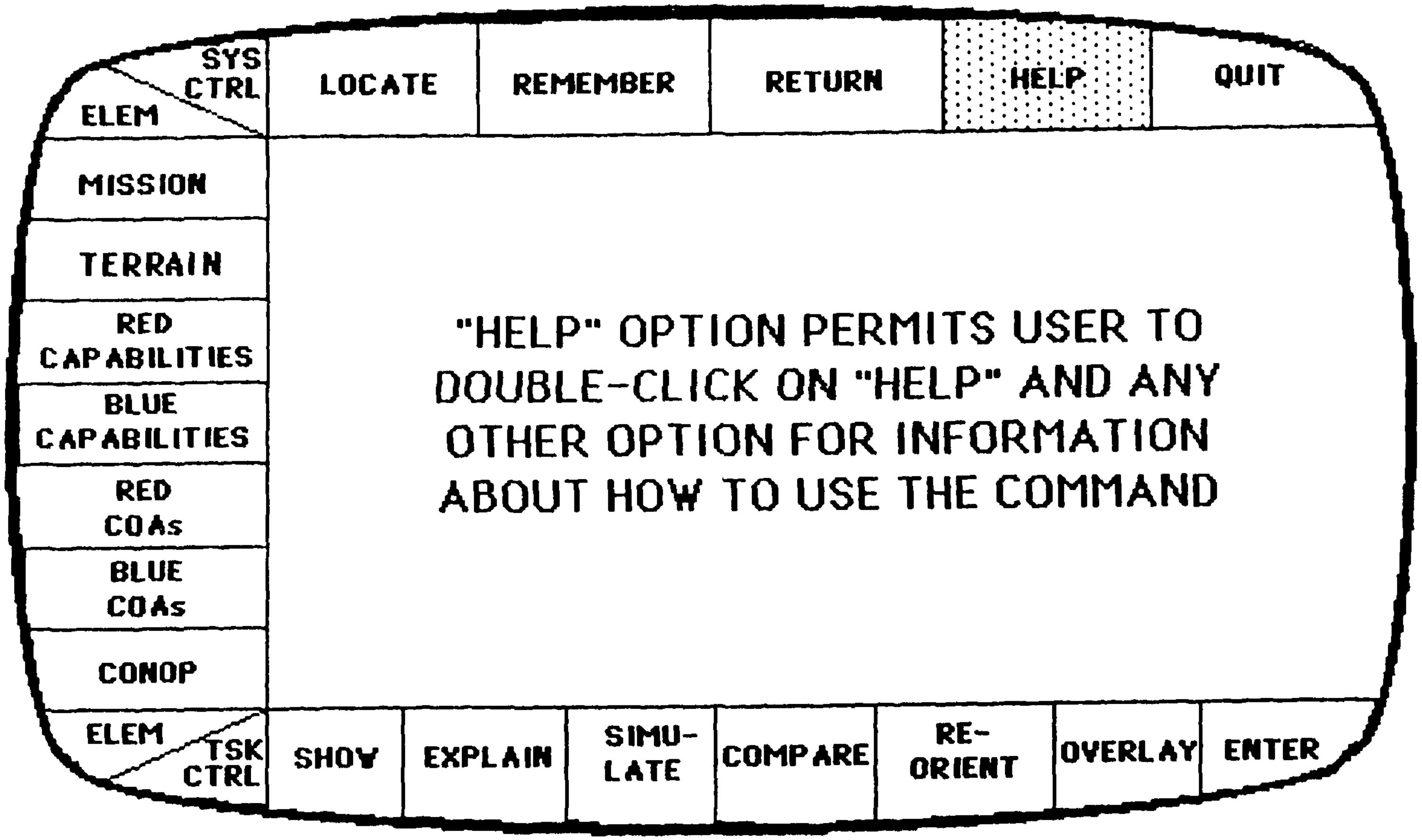


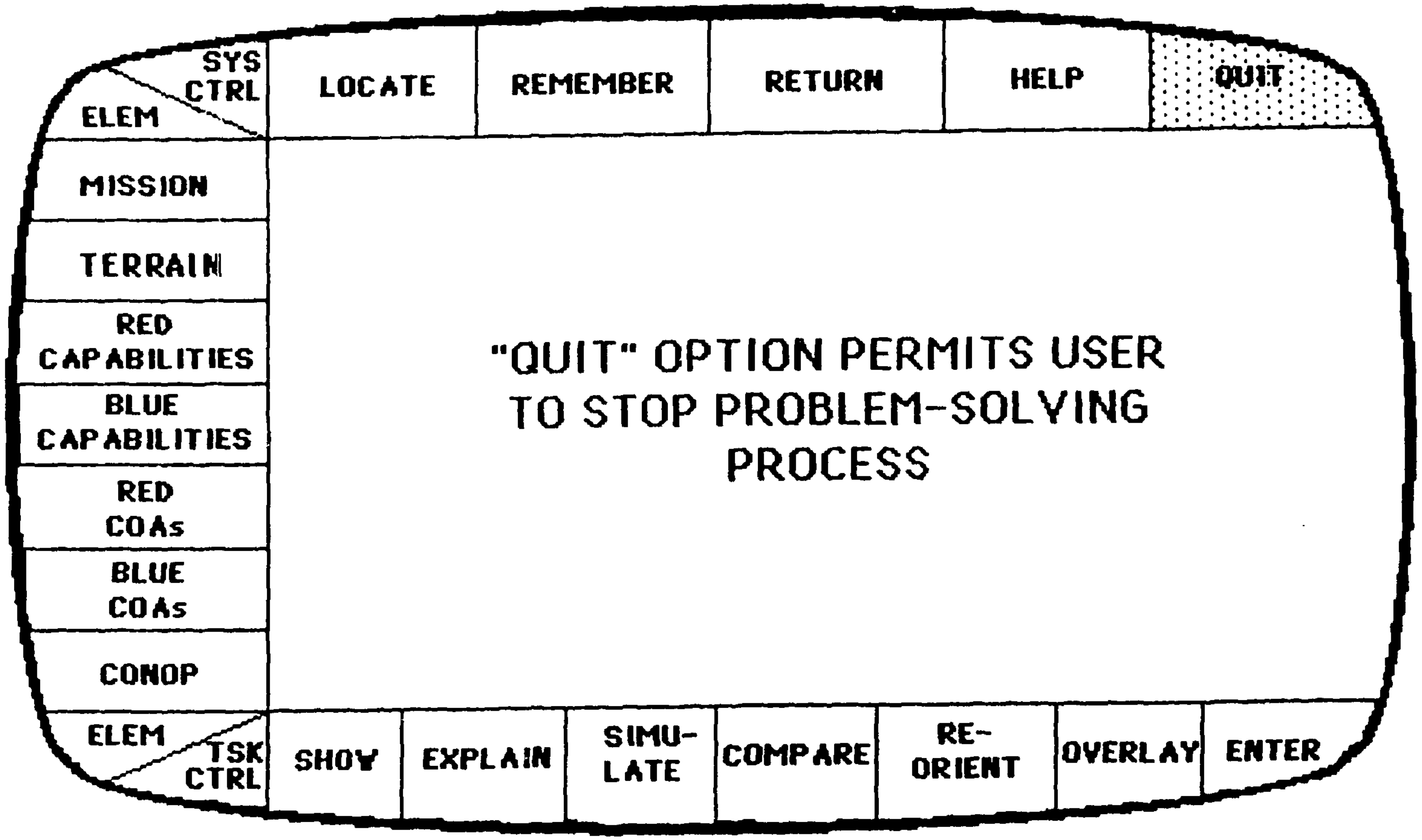
SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER
CONOP 1
CONOP 2
CONOP...
CONCEPT OF OPERATION (CONOP) SUB-MENU

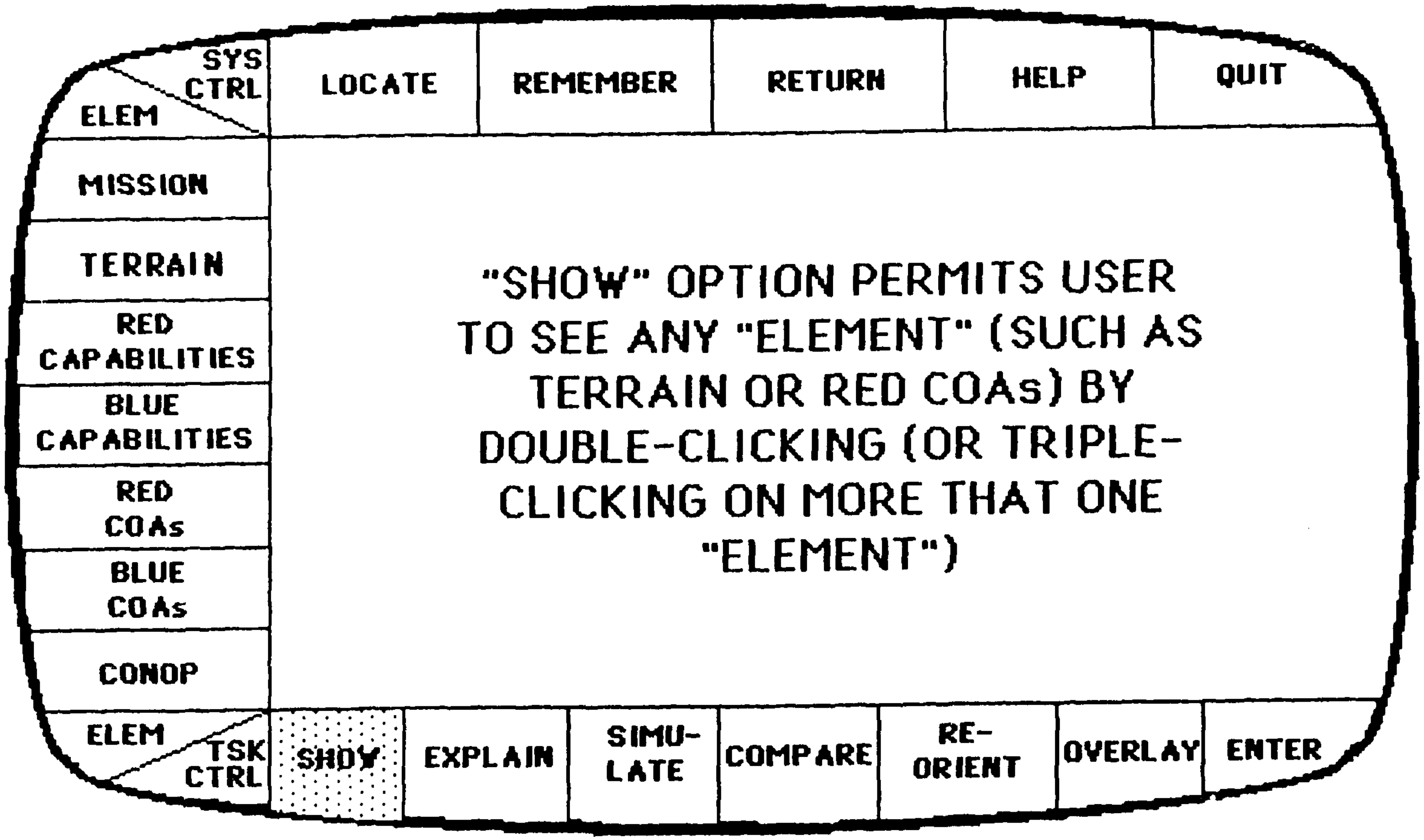

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER
"LOCATE" OPTION PERMITS USER TO DETERMINE WHERE IN THE PROBLEM-SOLVING PROCESS HE/SHE IS

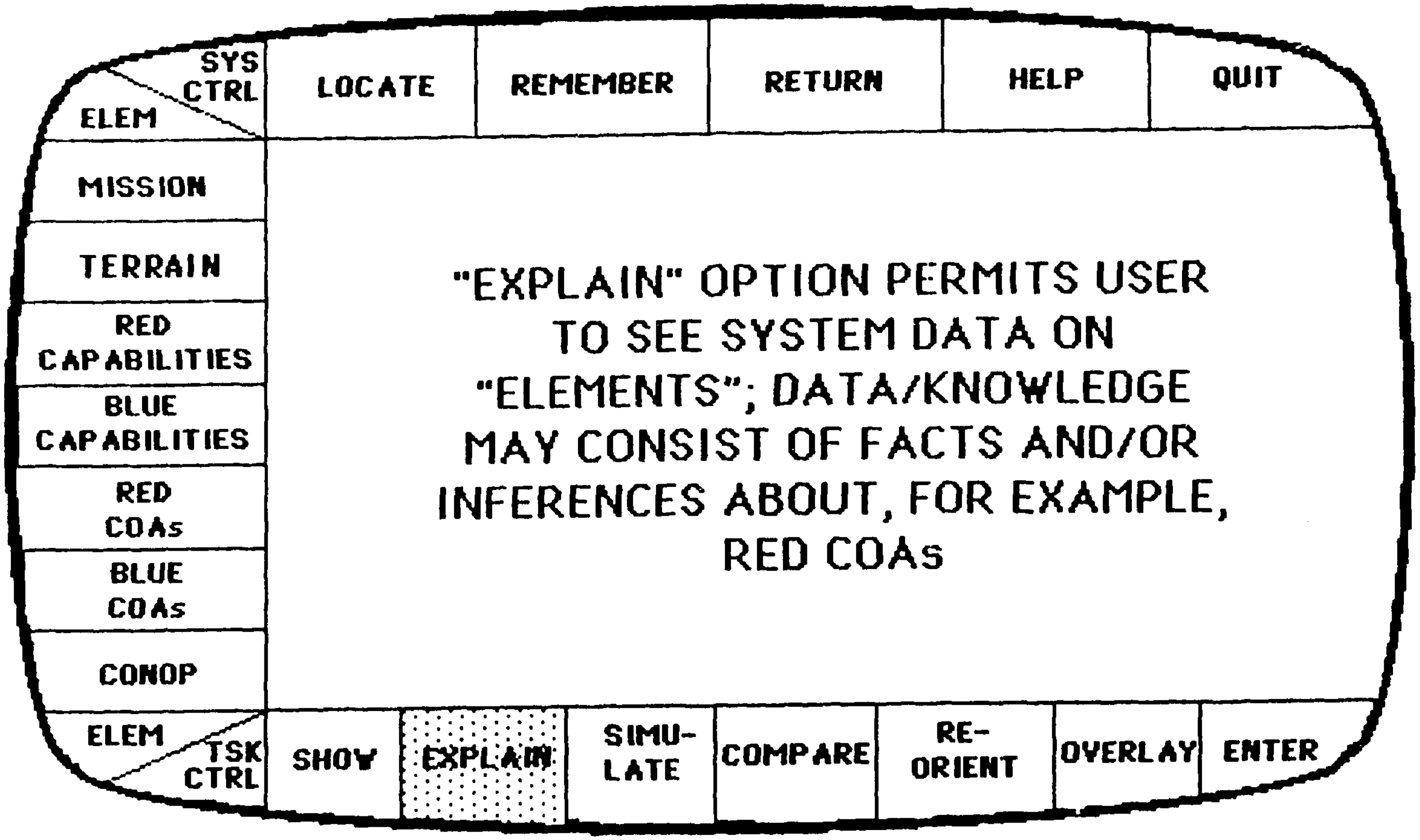

ELEM
SYS CTRL
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
"REMEMBER" OPTION PERMITS USER TO LABEL AND STORE ENTRIES, OVERLAY-DRIVEN DISPLAYS, EXPLANATIONS, AND SO FORTH, THEREBY CREATING A LIBRARY OF ITEMS FOR SUBSEQUENT ANALYSIS
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

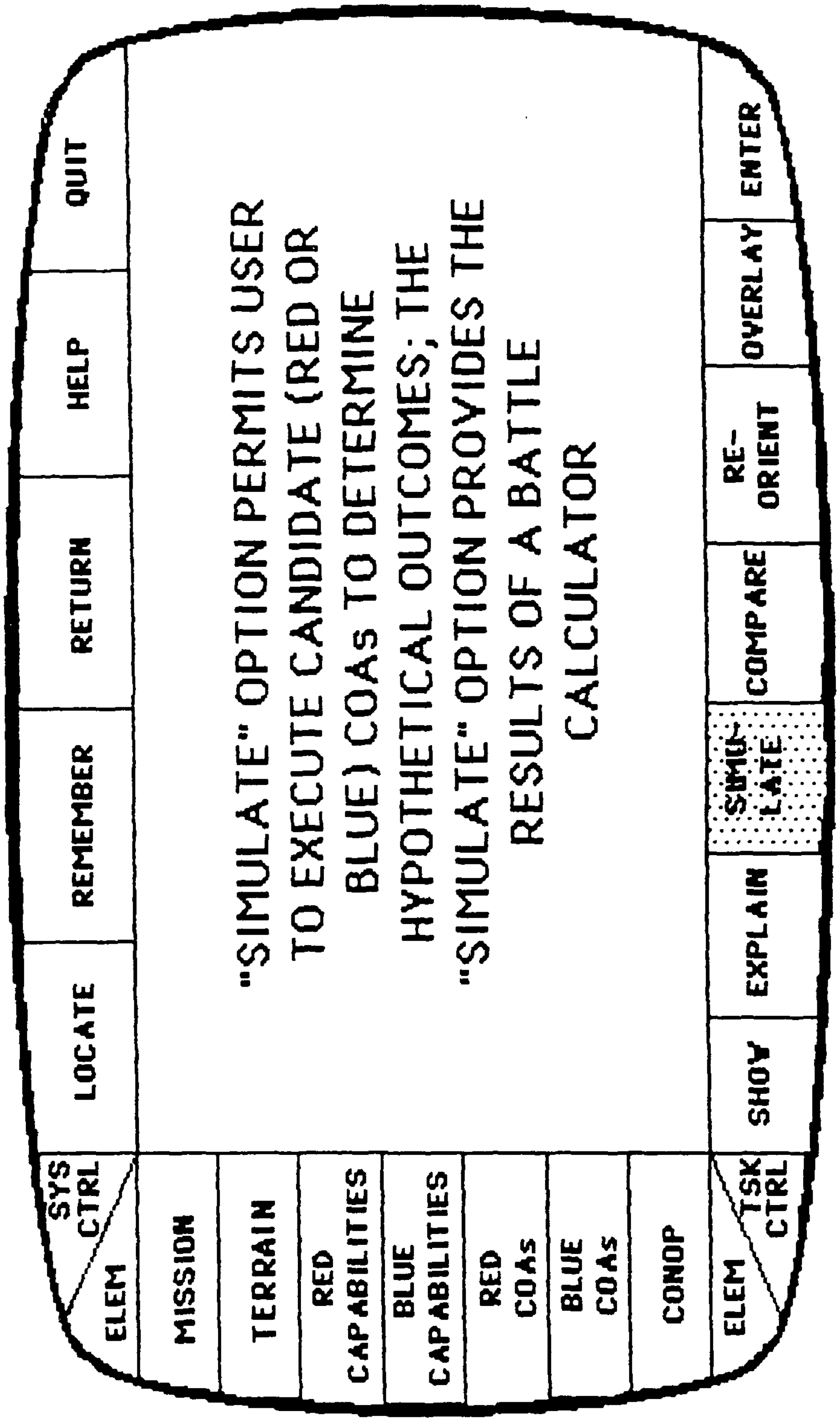

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
PLAN:
#1
#2
#3
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
"RETURN" OPTION PERMITS USER TO RETURN TO WHERE HE/SHE WAS WHEN INTERRUPTED, OR TO STORE A PLAN
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

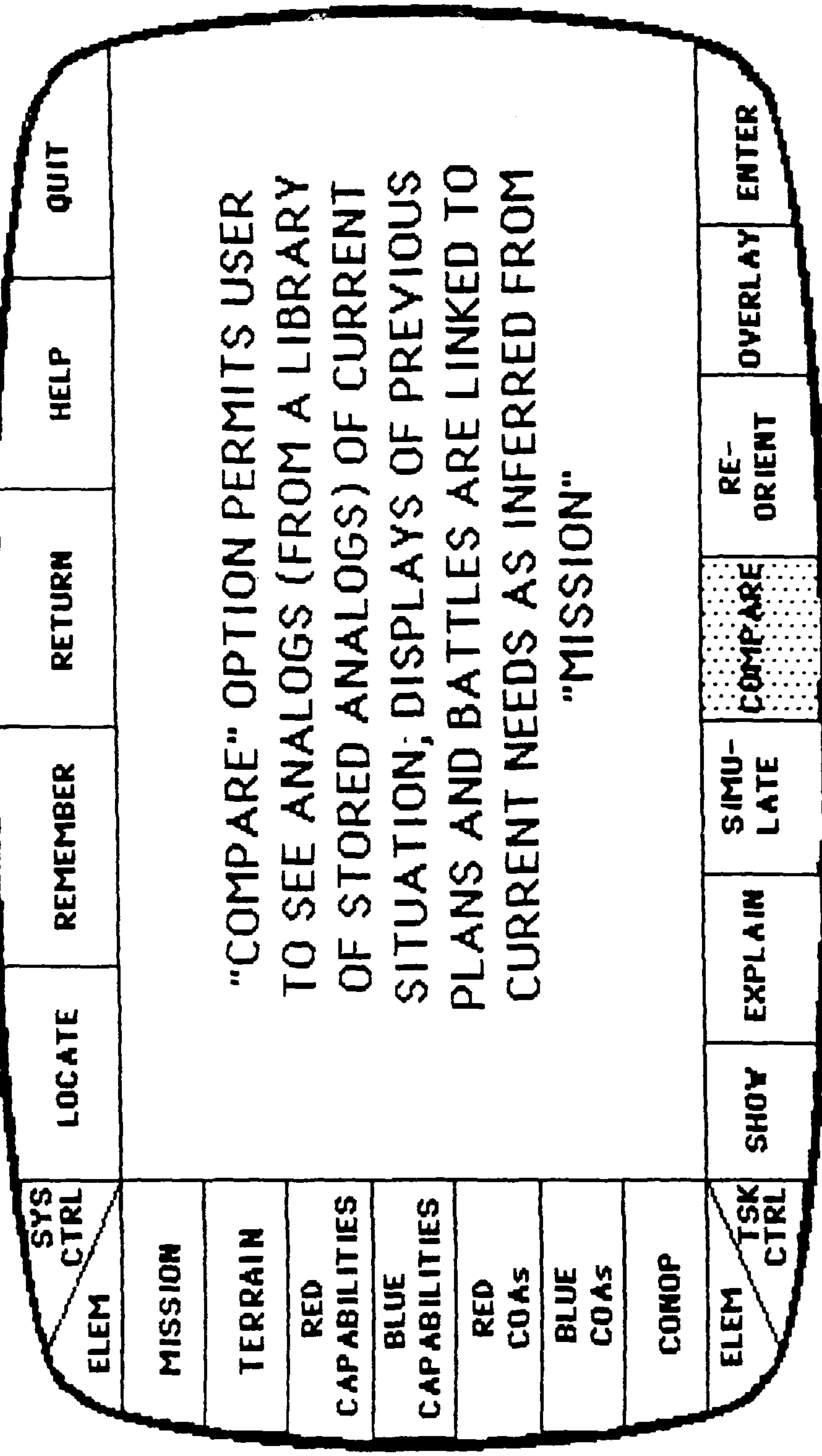

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
"HELP" OPTION PERMITS USER TO DOUBLE-CLICK ON "HELP" AND ANY OTHER OPTION FOR INFORMATION ABOUT HOW TO USE THE COMMAND
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

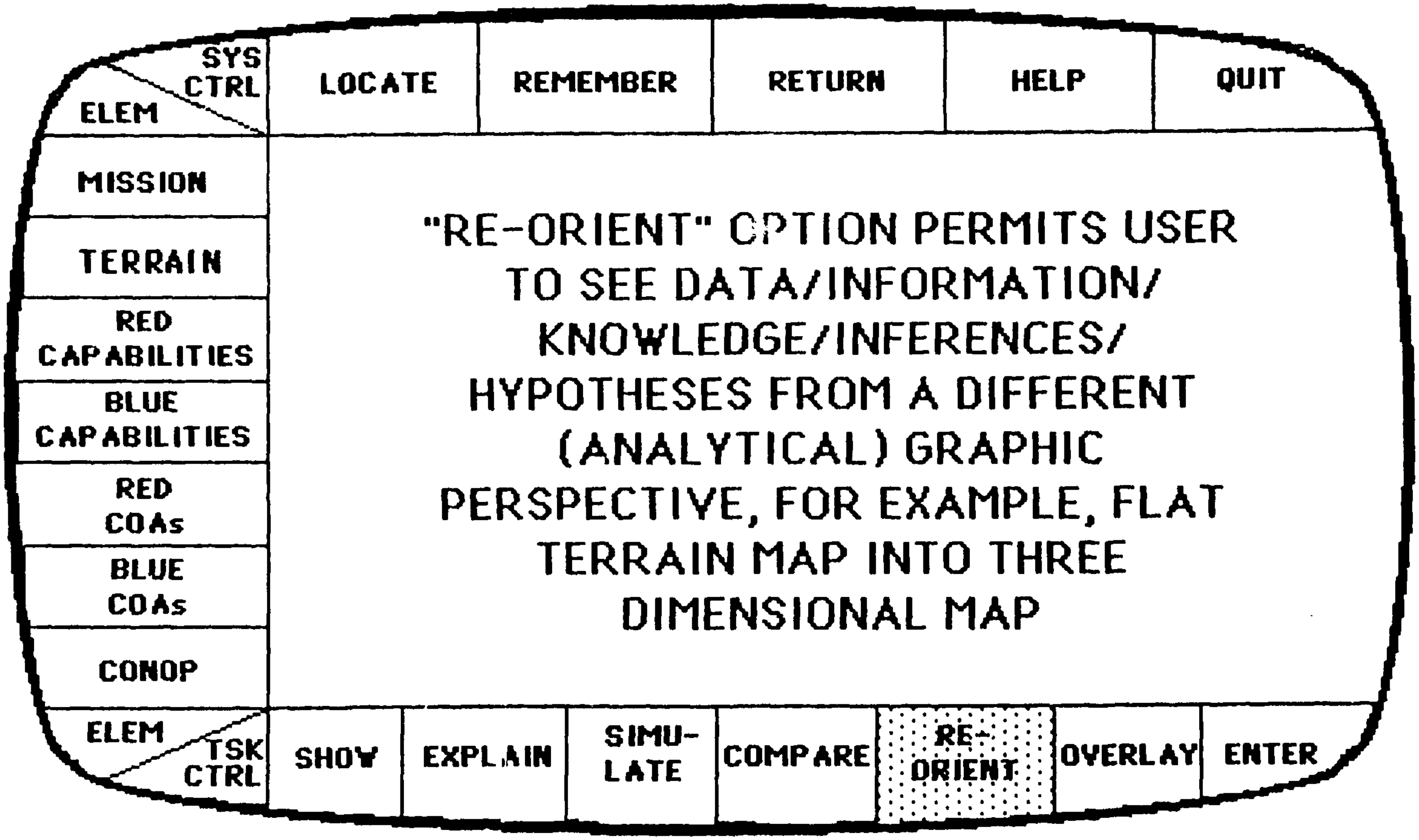

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
"QUIT" OPTION PERMITS USER TO STOP PROBLEM-SOLVING PROCESS
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

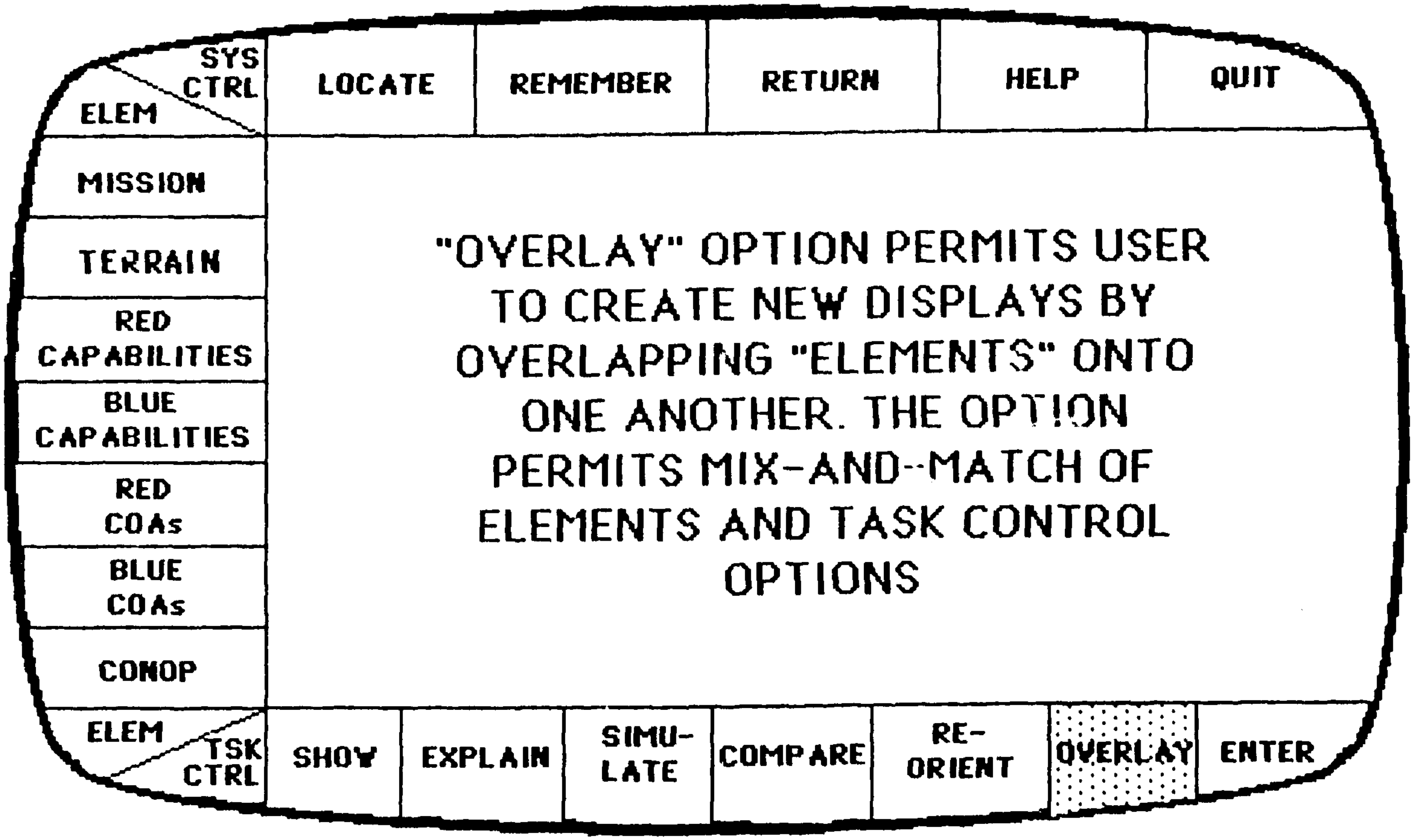

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER
"SHOW" OPTION PERMITS USER TO SEE ANY "ELEMENT" (SUCH AS TERRAIN OR RED COAs) BY DOUBLE-CLICKING (OR TRIPLE-CLICKING ON MORE THAT ONE "ELEMENT")

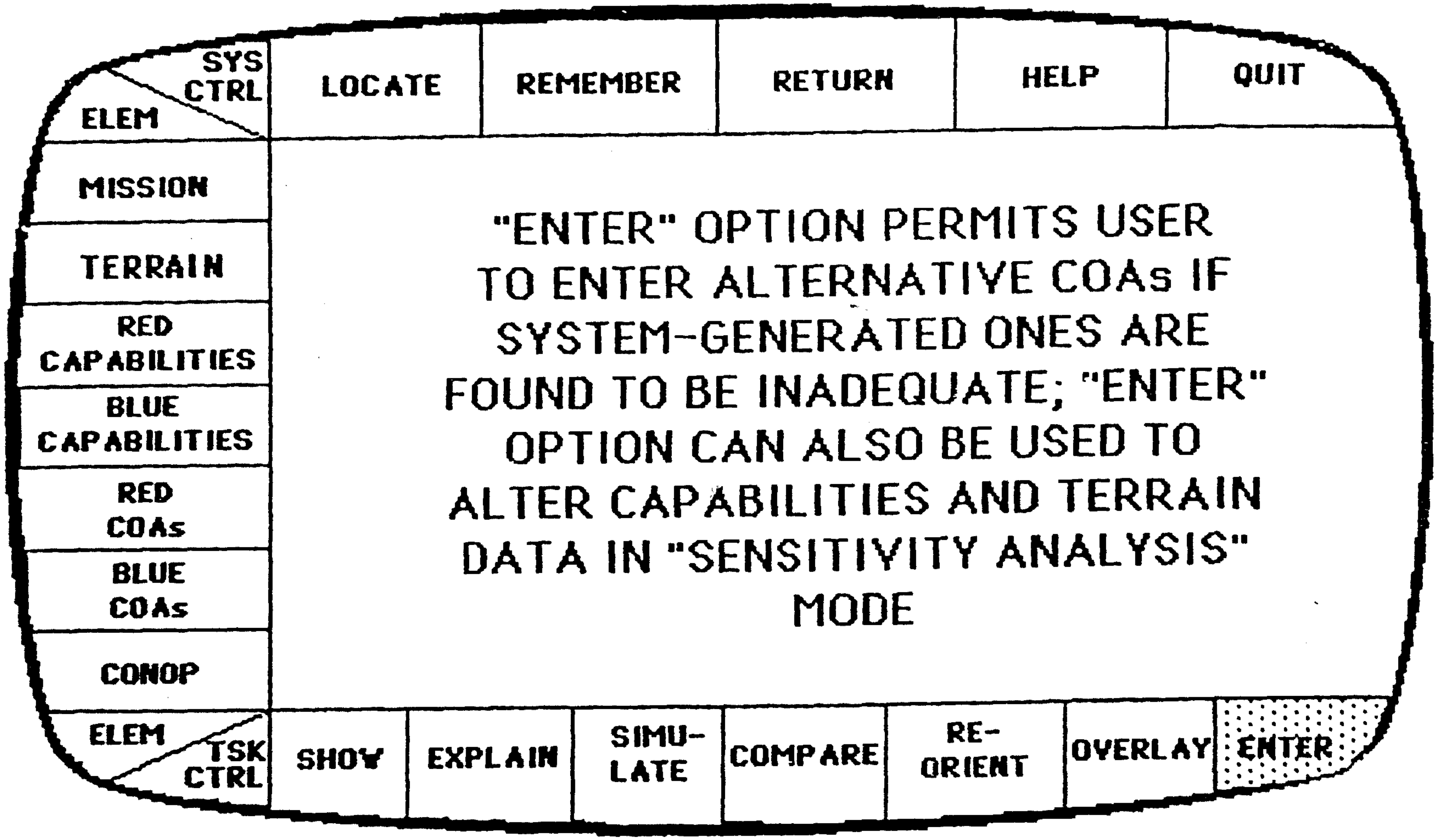

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER
"EXPLAIN" OPTION PERMITS USER TO SEE SYSTEM DATA ON "ELEMENTS"; DATA/KNOWLEDGE MAY CONSIST OF FACTS AND/OR INFERENCES ABOUT, FOR EXAMPLE, RED COAs


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
"SIMULATE" OPTION PERMITS USER TO EXECUTE CANDIDATE (RED OR BLUE) COAs TO DETERMINE HYPOTHETICAL OUTCOMES; THE "SIMULATE" OPTION PROVIDES THE RESULTS OF A BATTLE CALCULATOR
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
"COMPARE" OPTION PERMITS USER TO SEE ANALOGS (FROM A LIBRARY OF STORED ANALOGS) OF CURRENT SITUATION; DISPLAYS OF PREVIOUS PLANS AND BATTLES ARE LINKED TO CURRENT NEEDS AS INFERRED FROM "MISSION"
TSK CTRL
ELEM
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
"RE-ORIENT" OPTION PERMITS USER TO SEE DATA/INFORMATION/ KNOWLEDGE/INFERENCES/ HYPOTHESES FROM A DIFFERENT (ANALYTICAL) GRAPHIC PERSPECTIVE, FOR EXAMPLE, FLAT TERRAIN MAP INTO THREE DIMENSIONAL MAP
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER


SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
"OVERLAY" OPTION PERMITS USER TO CREATE NEW DISPLAYS BY OVERLAPPING "ELEMENTS" ONTO ONE ANOTHER. THE OPTION PERMITS MIX-AND-MATCH OF ELEMENTS AND TASK CONTROL OPTIONS
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

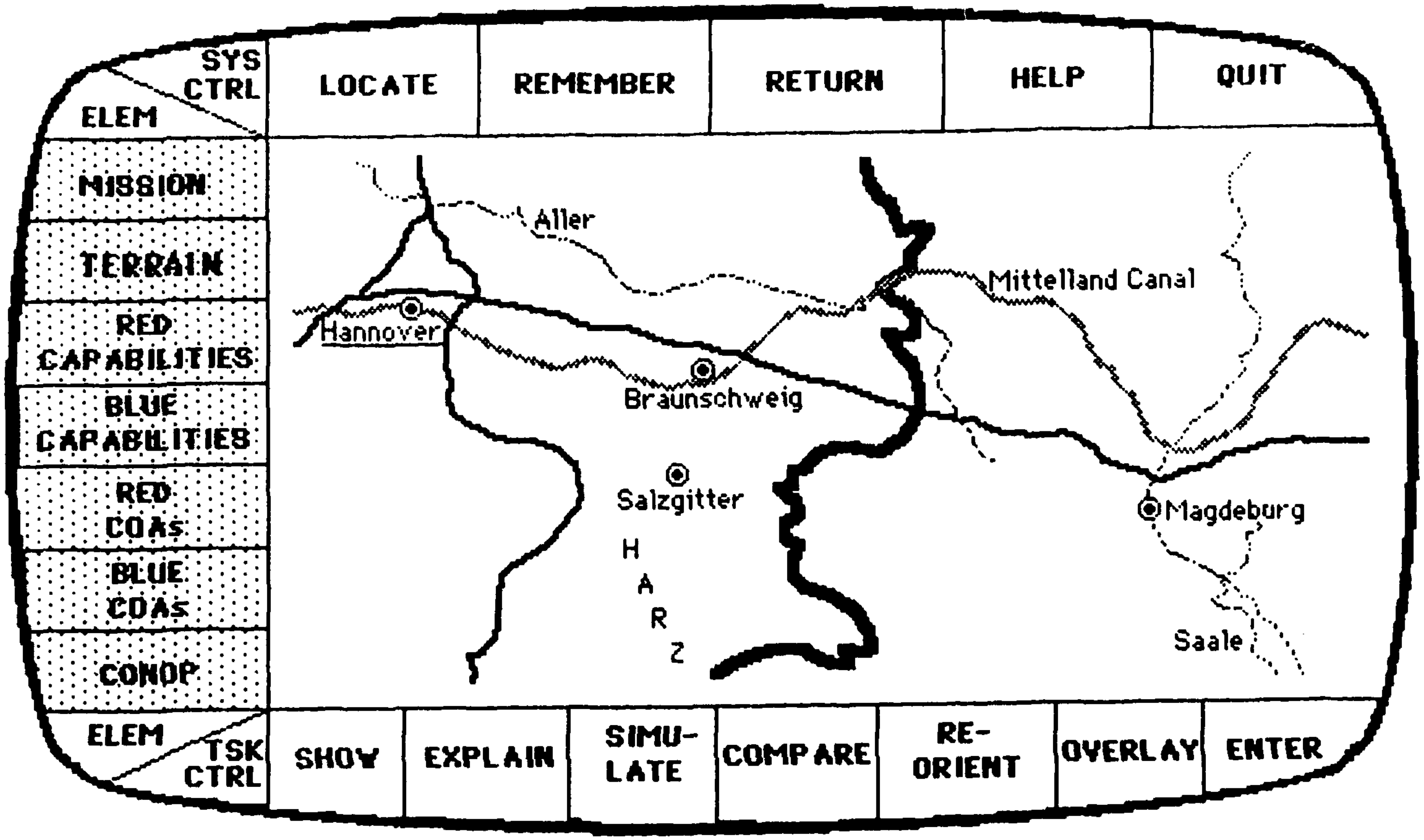

SYS CTRL
ELEM
LOCATE
REMEMBER
RETURN
HELP
QUIT
MISSION
TERRAIN
RED CAPABILITIES
BLUE CAPABILITIES
RED COAs
BLUE COAs
CONOP
"ENTER" OPTION PERMITS USER TO ENTER ALTERNATIVE COAs IF SYSTEM-GENERATED ONES ARE FOUND TO BE INADEQUATE; "ENTER" OPTION CAN ALSO BE USED TO ALTER CAPABILITIES AND TERRAIN DATA IN "SENSITIVITY ANALYSIS" MODE
ELEM
TSK CTRL
SHOW
EXPLAIN
SIMU-LATE
COMPARE
RE-ORIENT
OVERLAY
ENTER

of Action," "Blue Courses of Action," and the "Concept of Operations (CONOP)" -- constitute the substantive essence of the system concept. The system and task controls are placed at the top and bottom of the displays, respectively. When an option is highlighted, it is "active" and can be clicked upon.

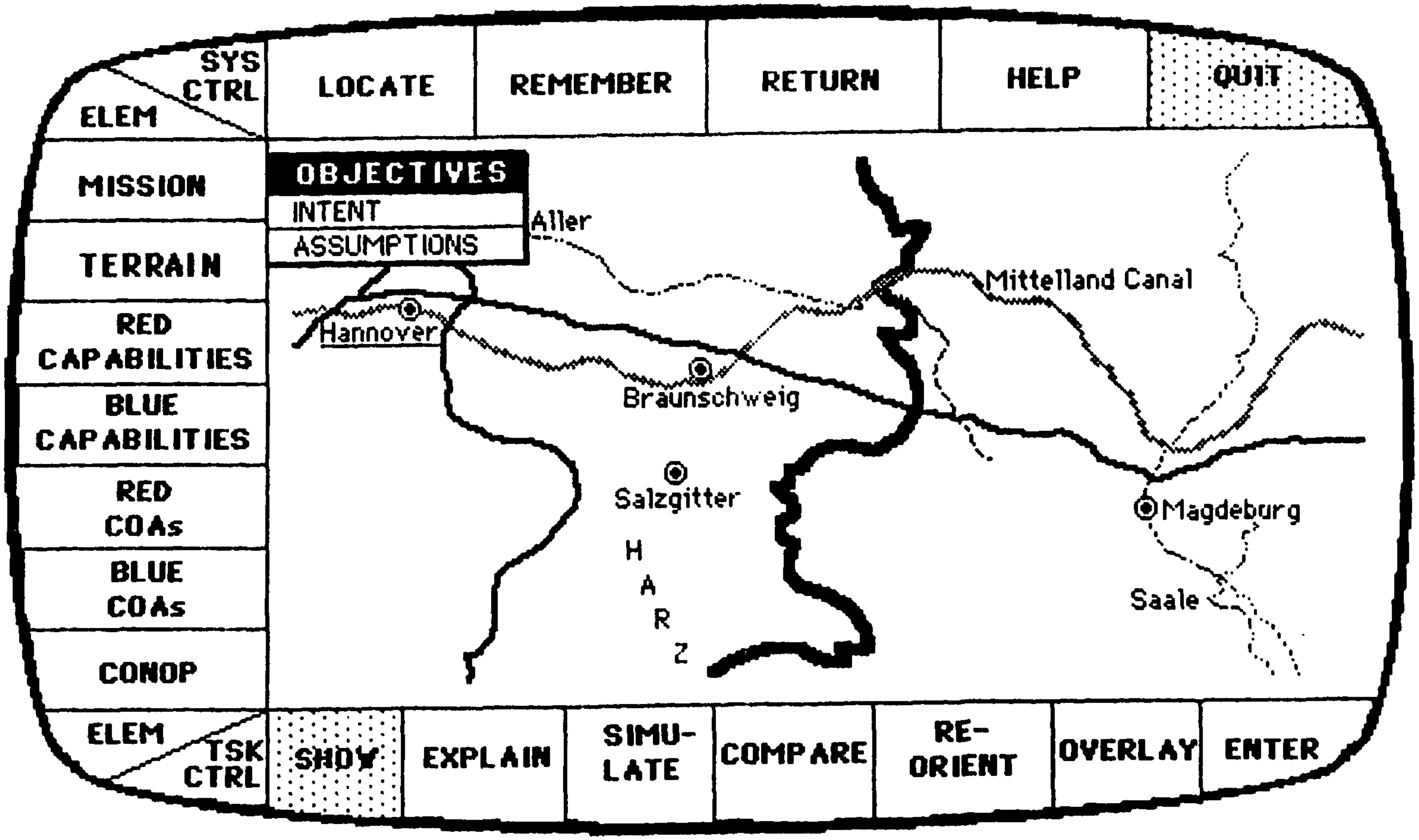
Each display in the storyboard is described by a brief note at the bottom. These notes are intended to guide the user through the storyboard and to explain what the board is doing at any point in time. The brevity of the notes is deliberate; our intention is to test the UCI system concept by providing storyboard users with as little information as possible: if users get lost then the system concept is perhaps not as good as we'd like; on the other hand, if they find the system immediately and intuitively easy to use -- much like a program for an Apple Macintosh -- then perhaps we have identified a workable concept.

The storyboard is replete with examples of "analytical graphics" and principles of "visual cognition." We tried to use graphics to substitute for alphanumerics and tools like animation to support visual cognition. We also tried to distill the interaction process down to its most diagnostic level, burdening the user only with necessary and important information and choices.

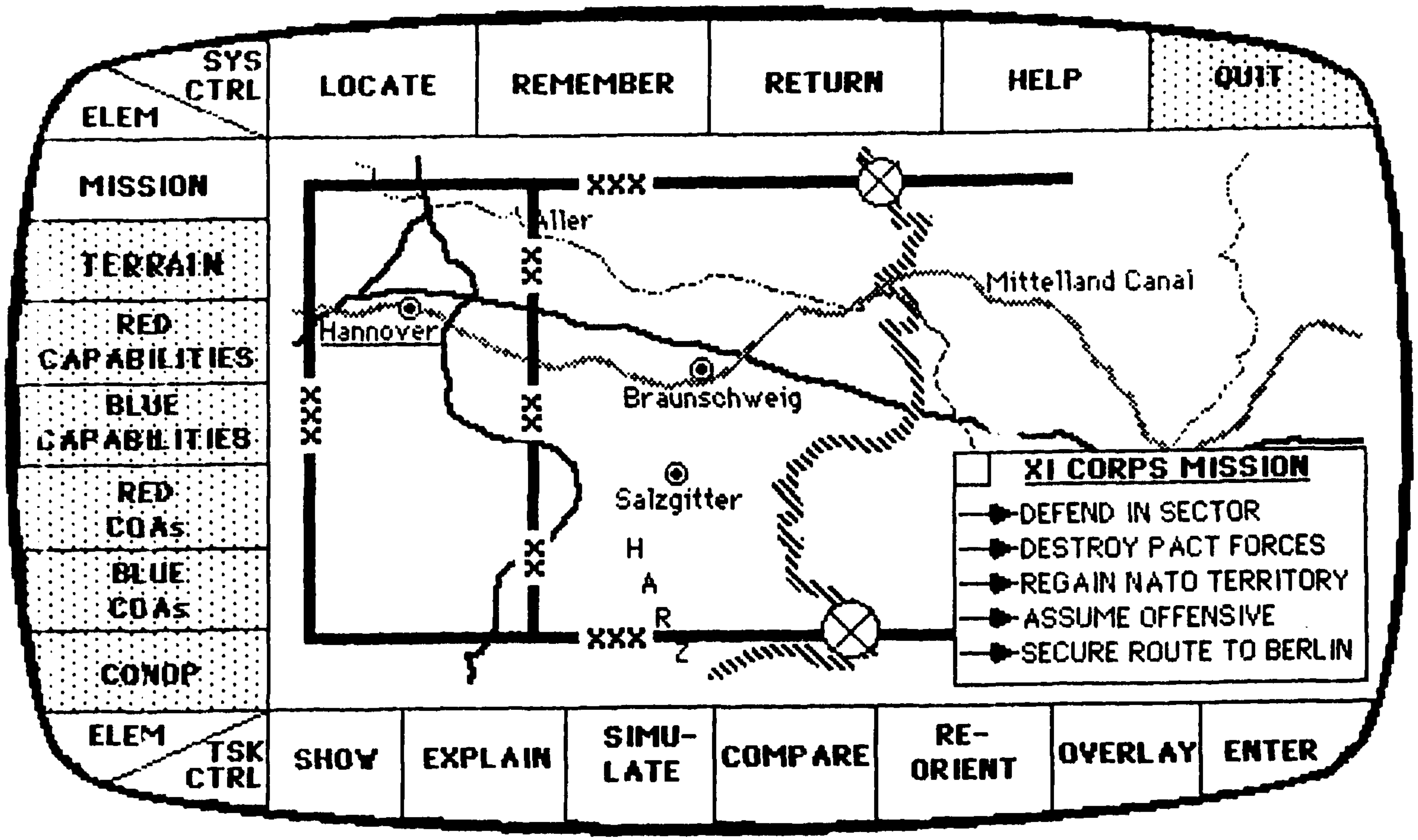
The storyboard that appears in this report is a "paper" storyboard that demonstrates the capabilities of the prototype; a software-based version that runs interactively also exists.



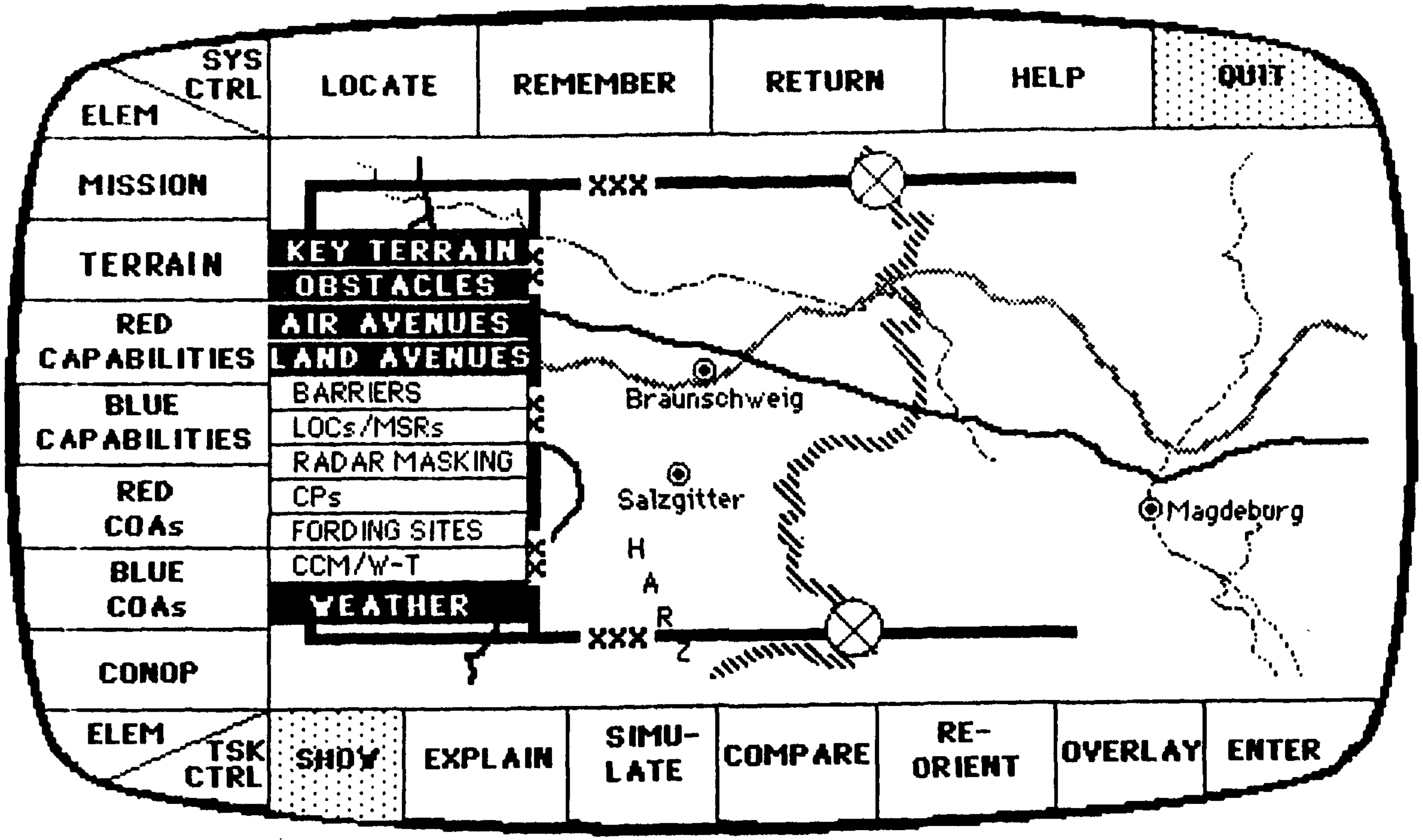
This display suggests how the work space within the menu structure can be filled with accessible data and information. In this case, the data is map-based, suggesting to the user that the primary "object" of analysis and manipulation will be the tactical map . . . the display also suggests that the planner can access any of the "elements" of tactical planning which reside along the left side of the display . . .



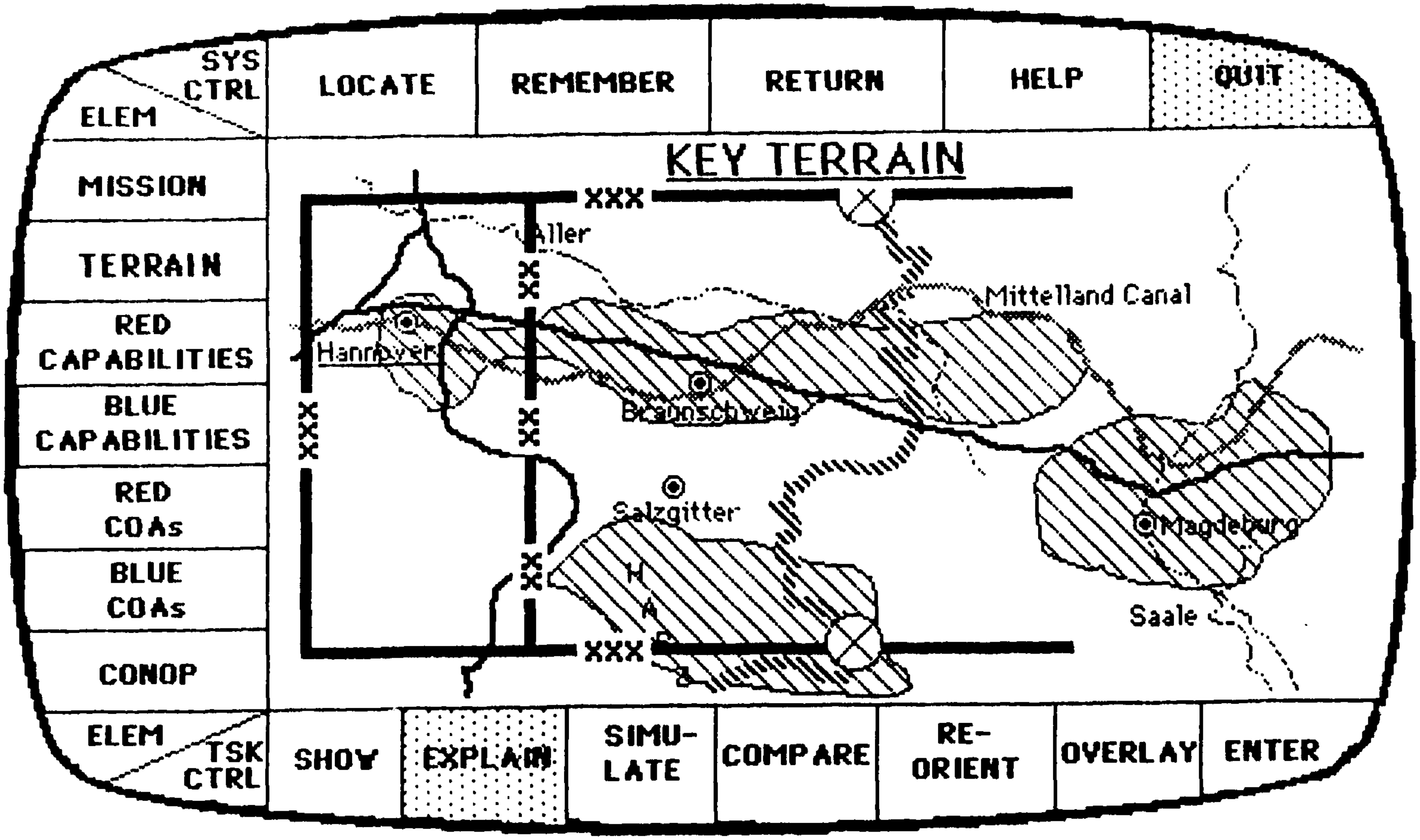
This display suggests how a user can execute planning tasks; in this case, the user is interested in the **Mission** and has selected information regarding the Mission's **Objectives** . . .



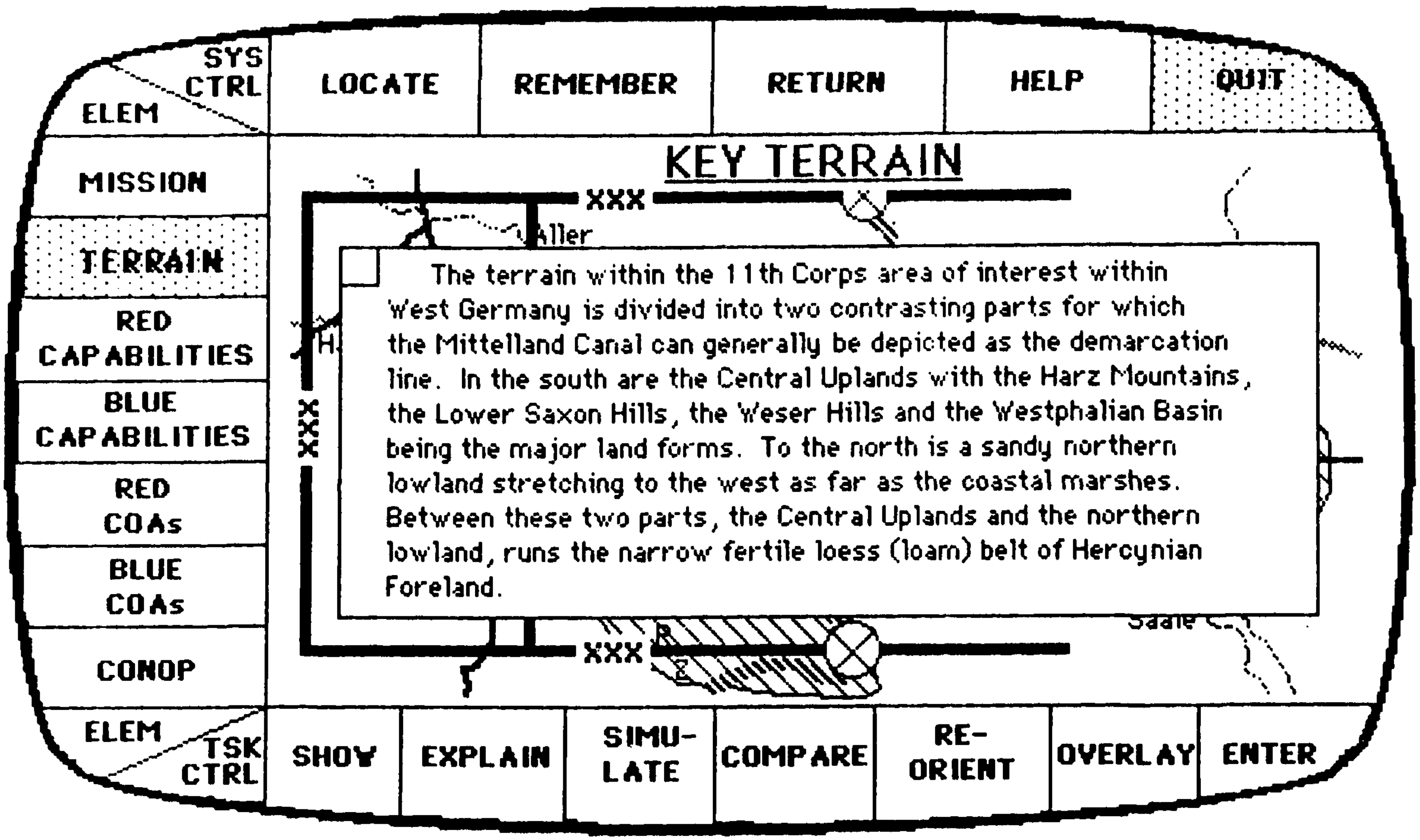
This display displays the **Mission's Objectives** to the user via integrated text and graphics . . . the Corps area of interest/operations/influence is displayed to the planner; the **Objectives** are also described in abbreviated text . . . the integration of text and graphics supports important cognitive functions . . .



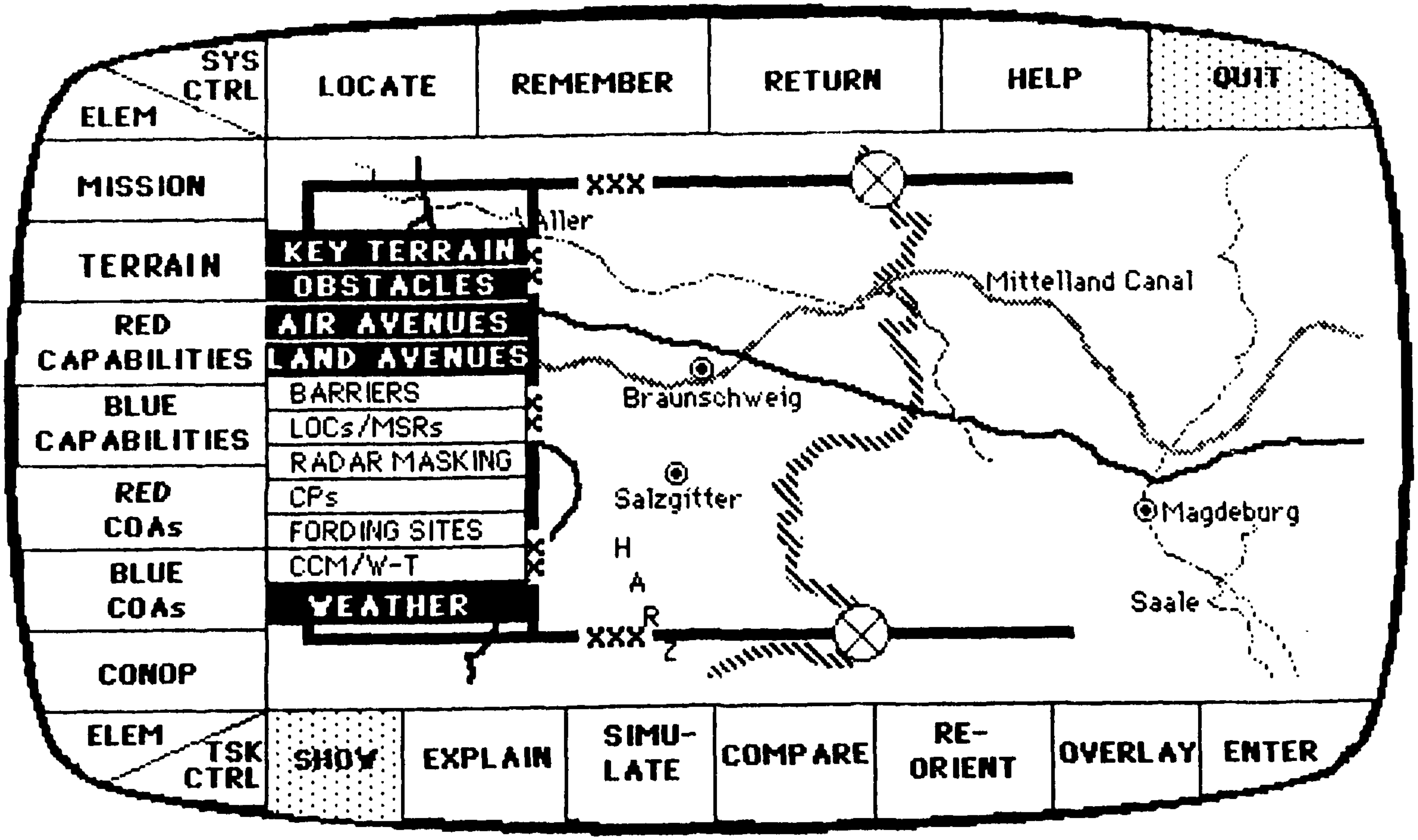
The user now requests to see some aspects of terrain by clicking on **Show** and several terrain options that appear in the **Terrain** sub-menu . . . the map displays remains as it was, displaying the Corps areas . . .



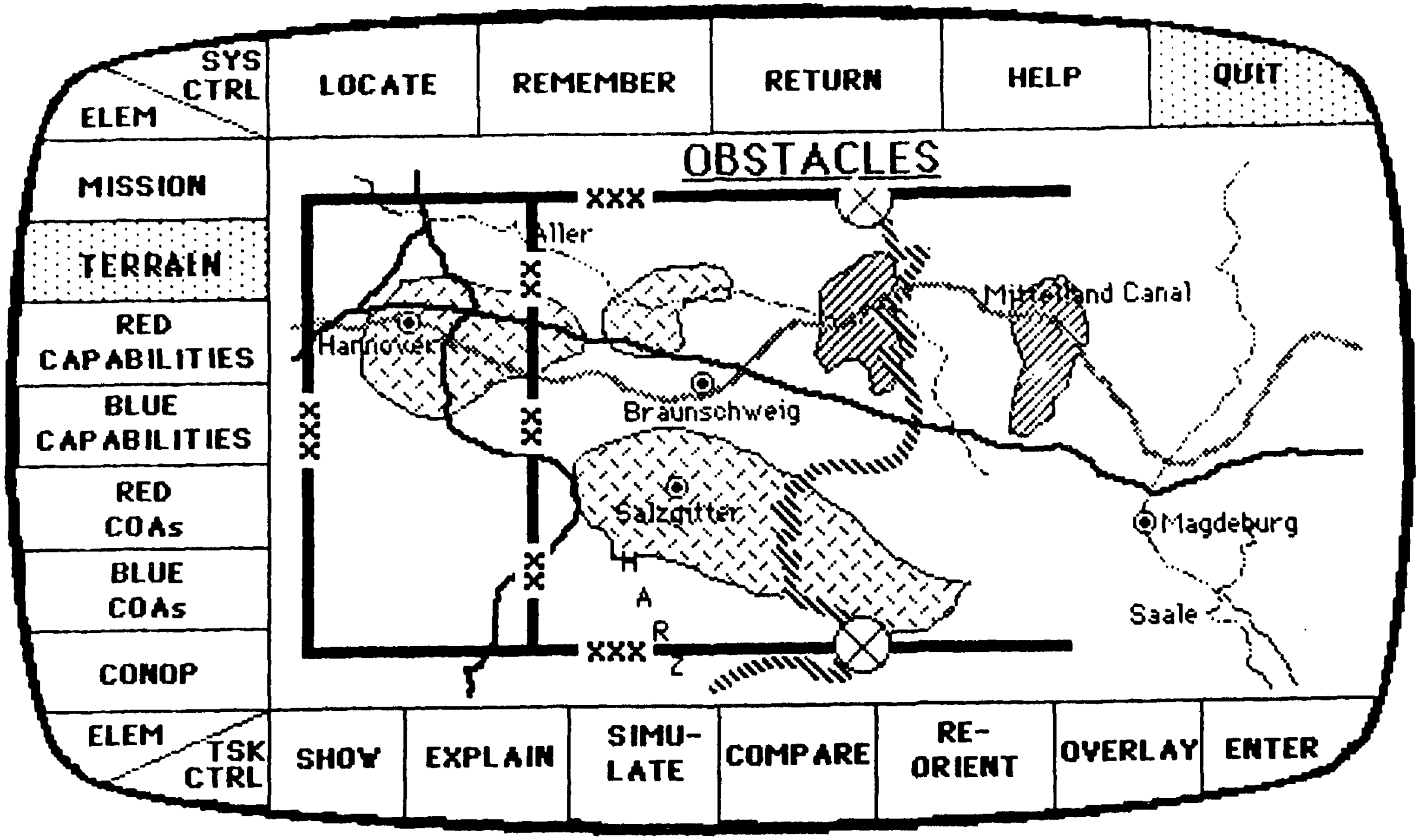
Key Terrain is displayed to the user; the option to request more information about the **Key Terrain** is ever-present and, in this case, accessed by the planner via the **Explain** command . . .



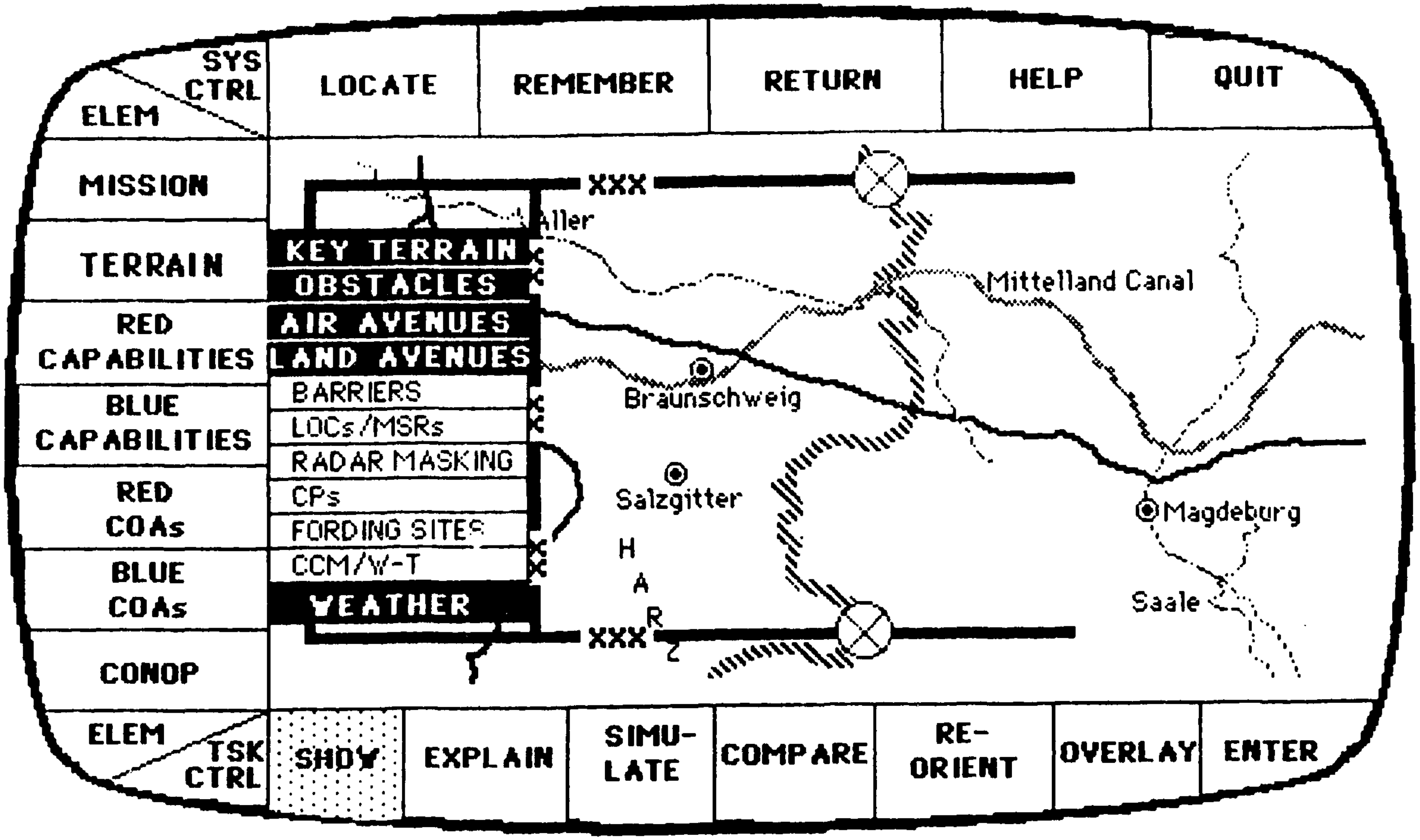
Text appears to explain what has already been presented to the user graphically . . . additional information can be stored hierarchically "under" the **Explain** command . . .



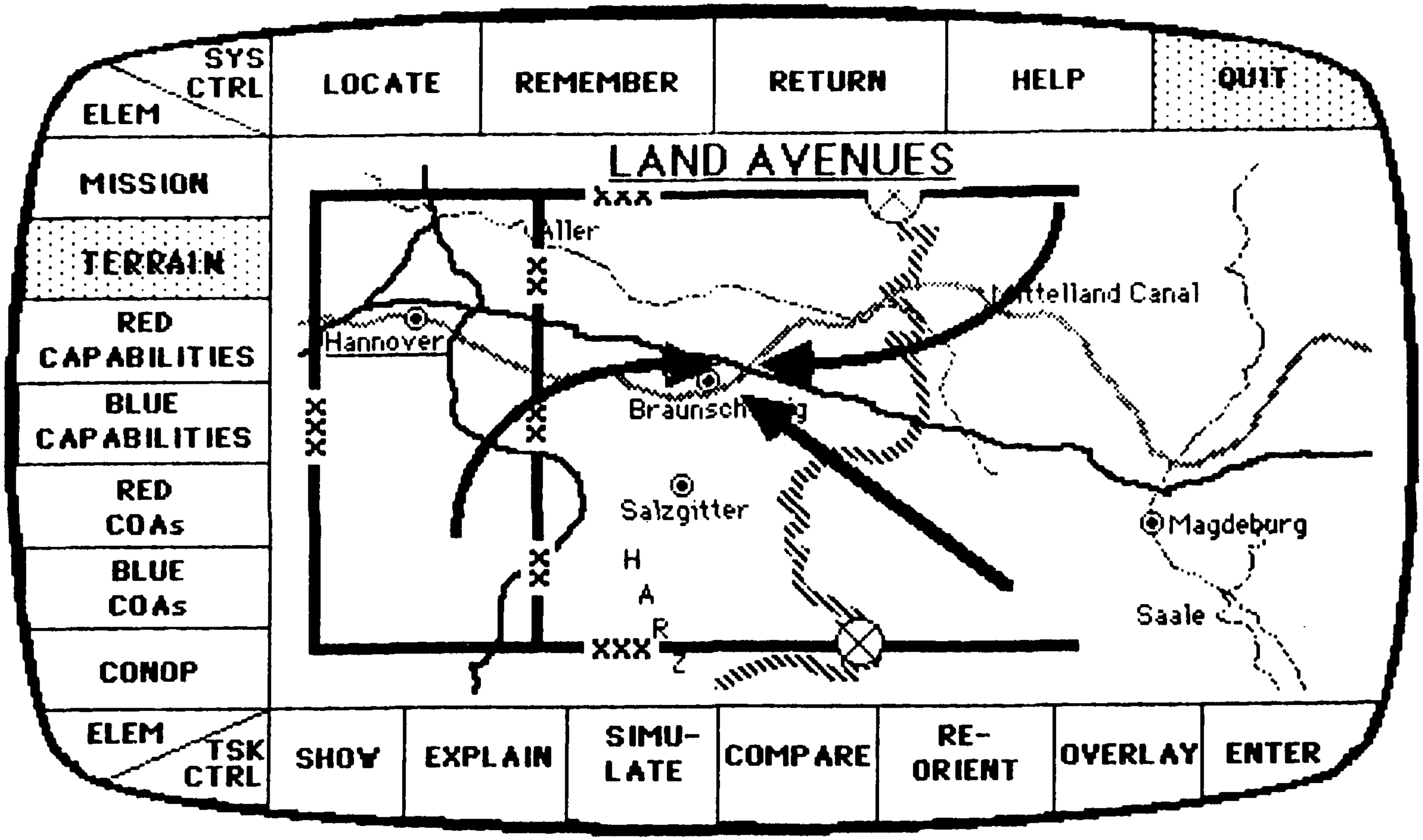
The user selects additional information about other aspects of terrain . . .



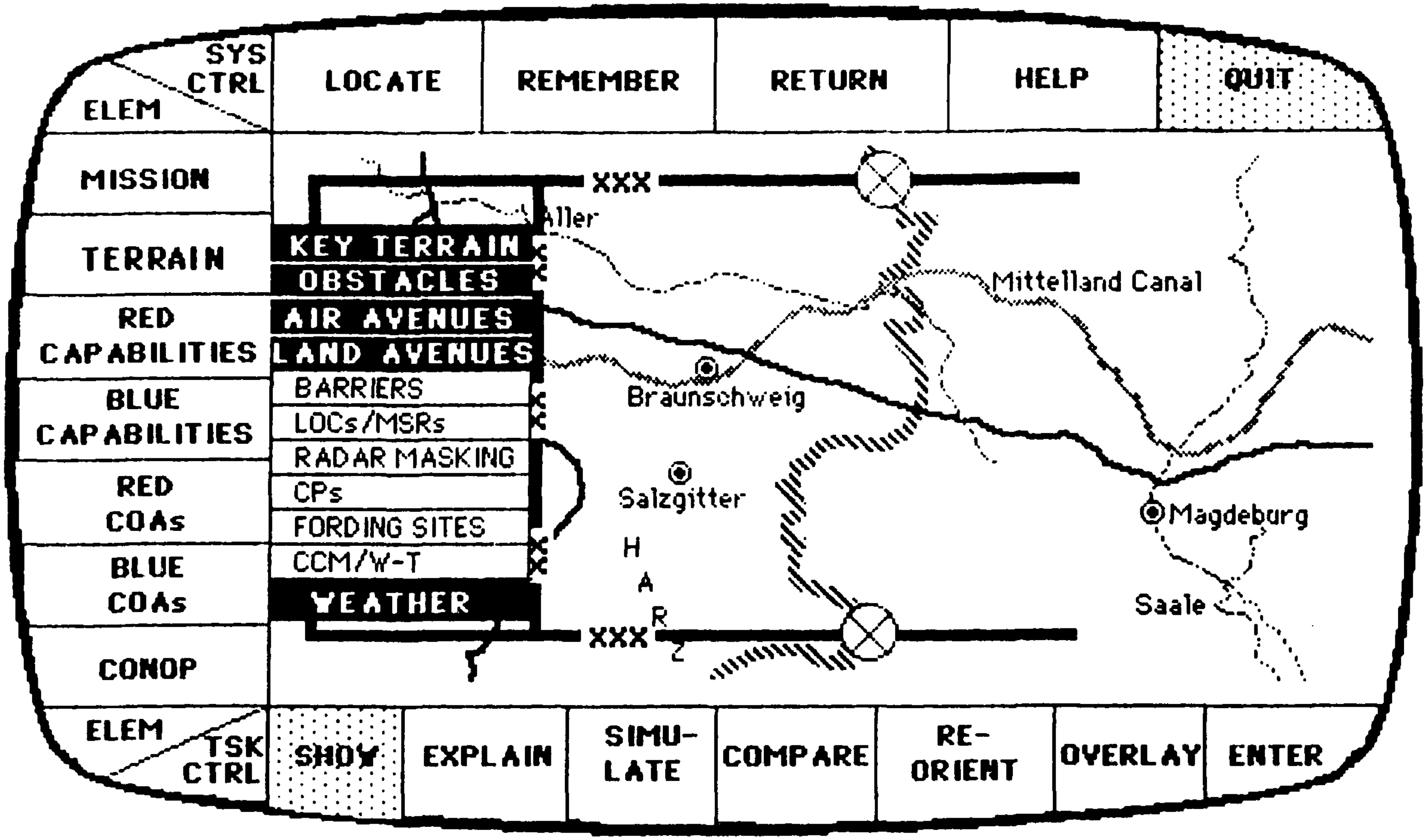
The system displays **Obstacles** to the planner . . . here too the user has the option of requesting more detail about the obstacles via the **Explain** command . . .



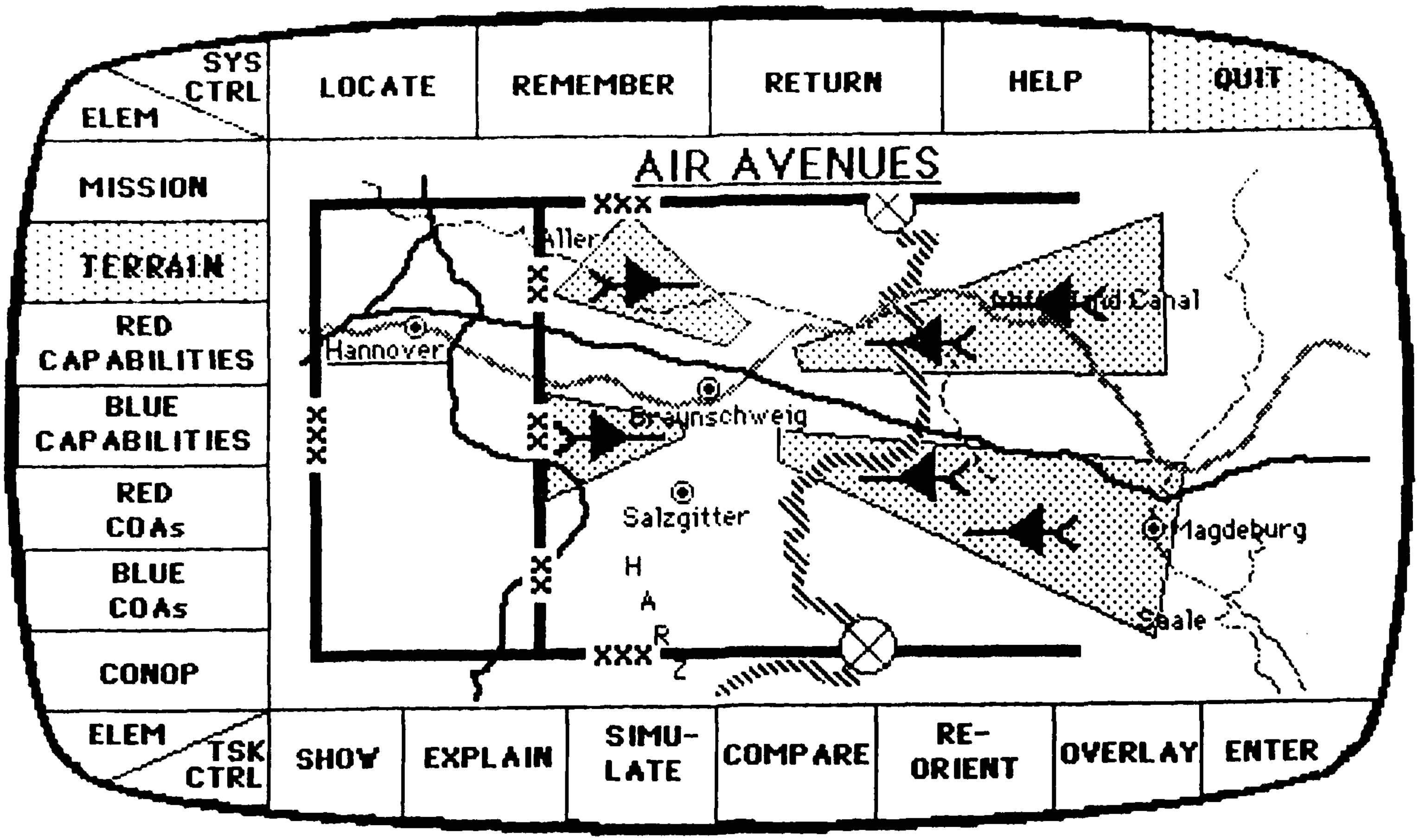
The planner seeks yet more information about **Terrain** . . .



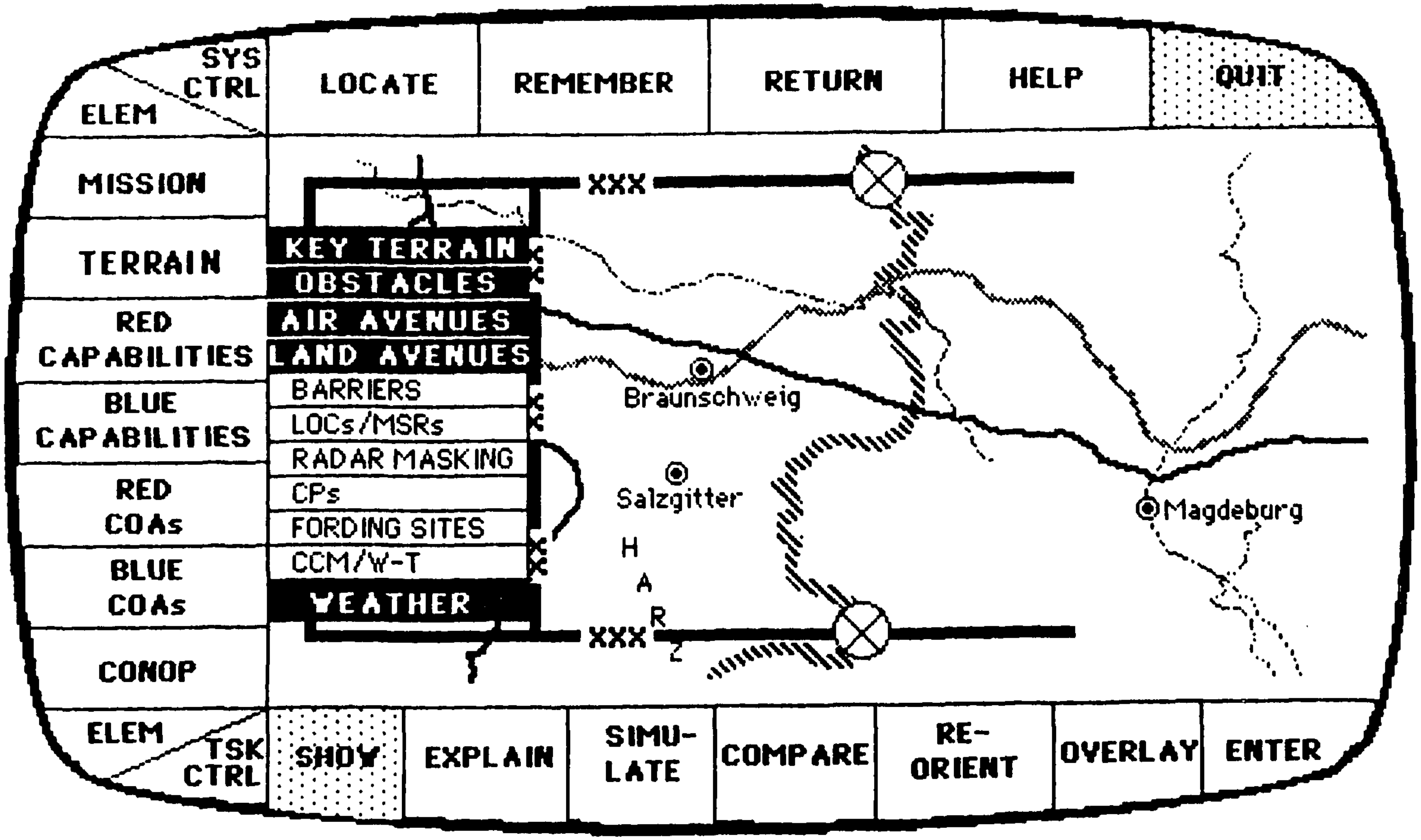
Land Avenues are displayed to the user . . .



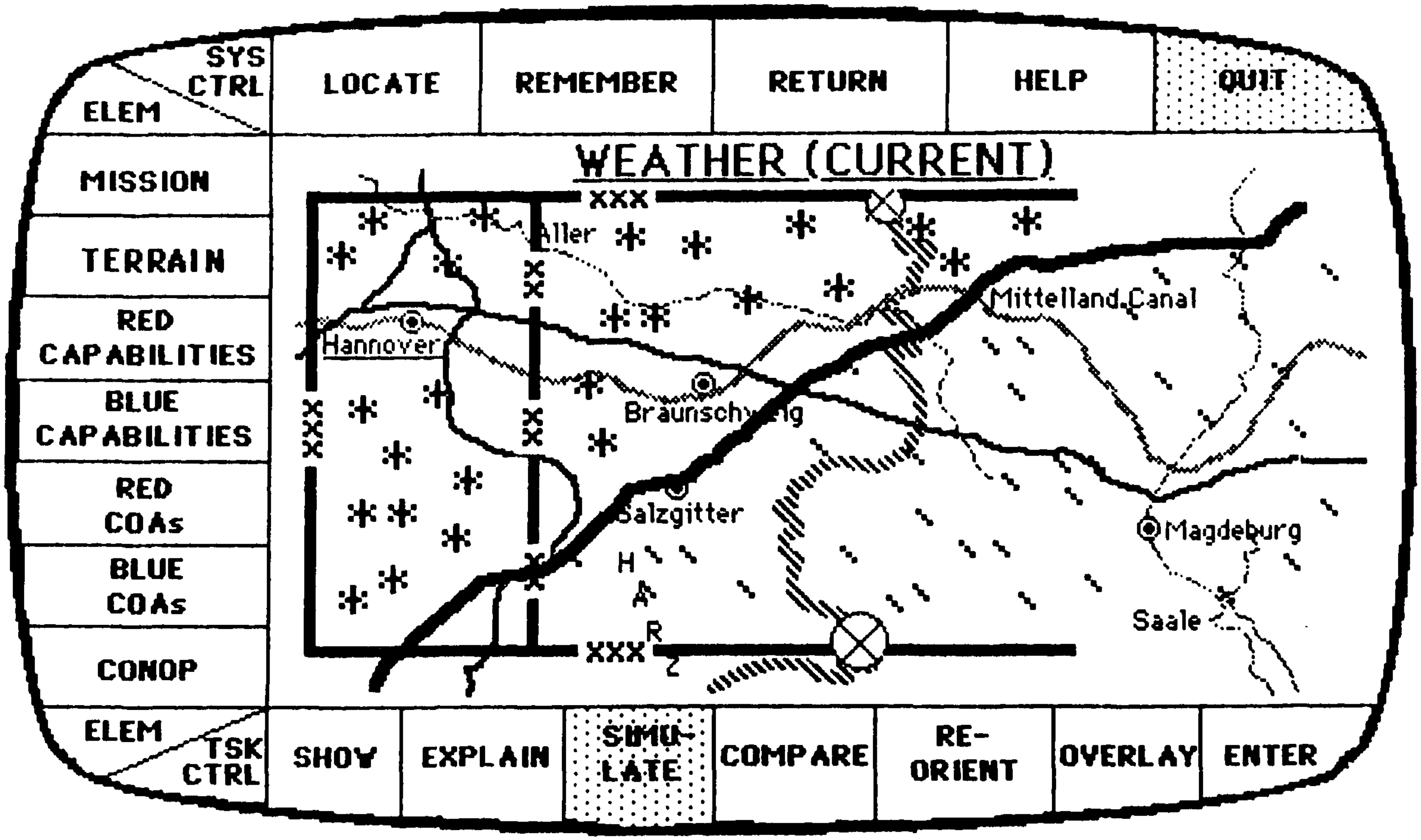
More information is requested . . .



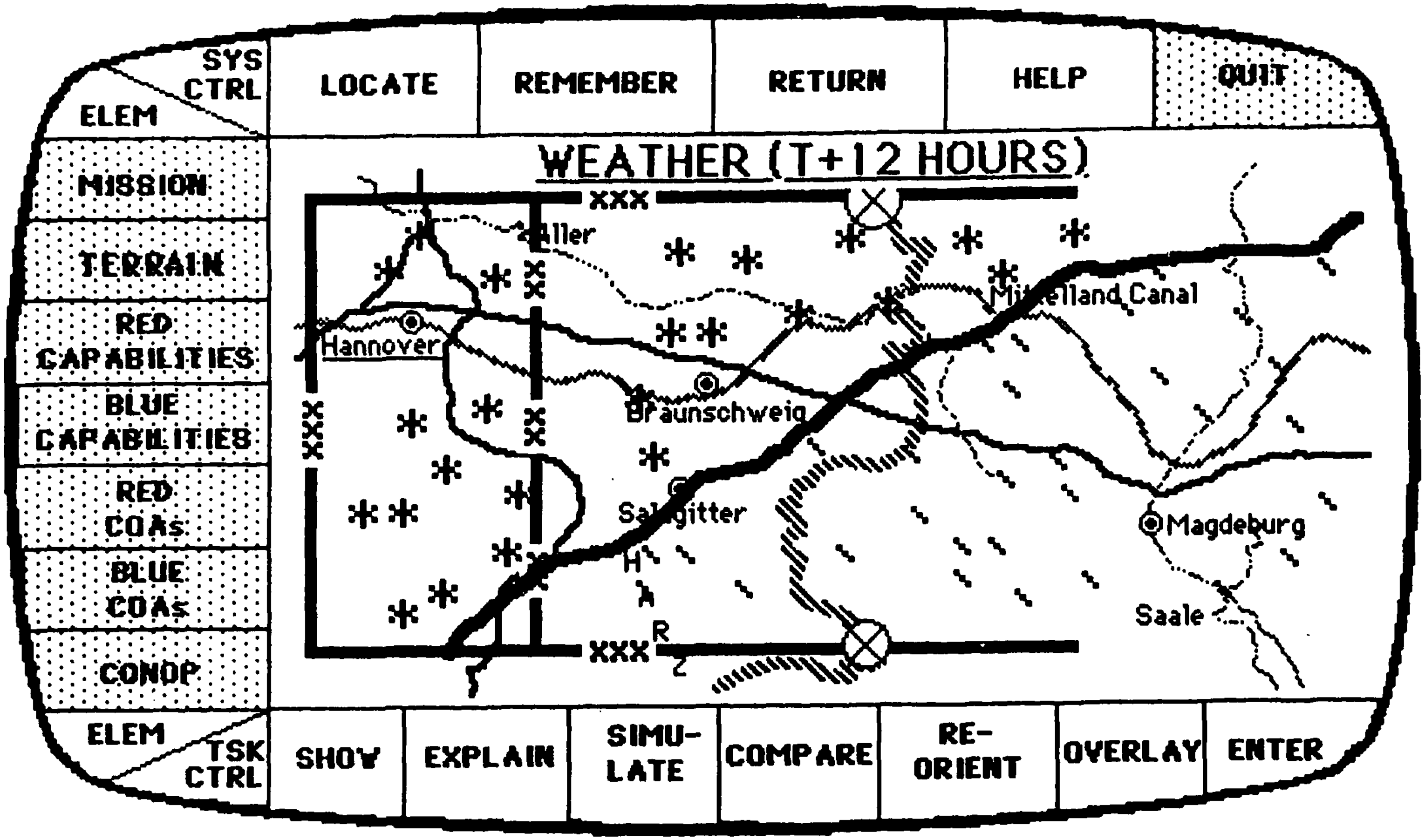
Air Avenues are displayed to the user . . .



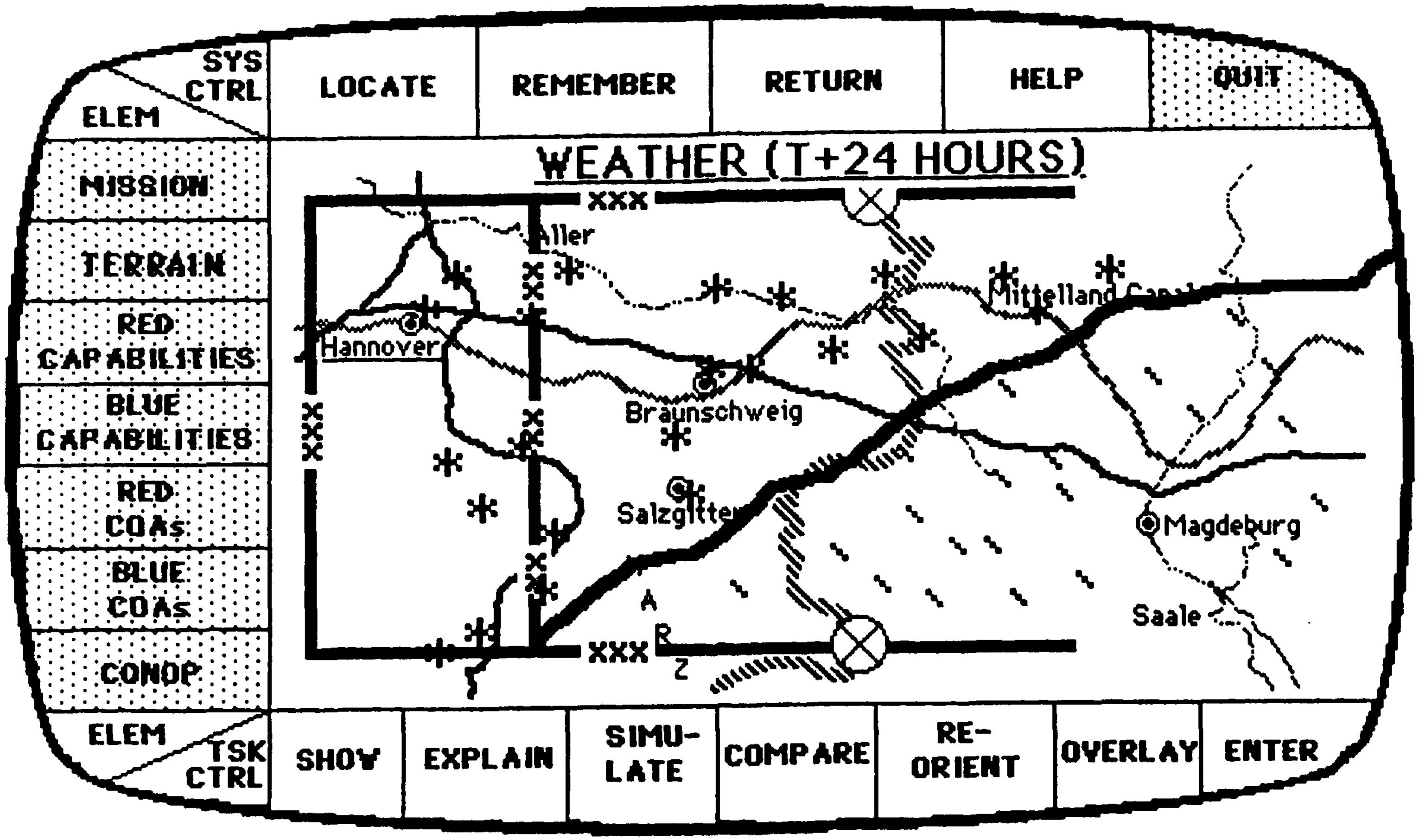
Information about **Weather** is requested . . .



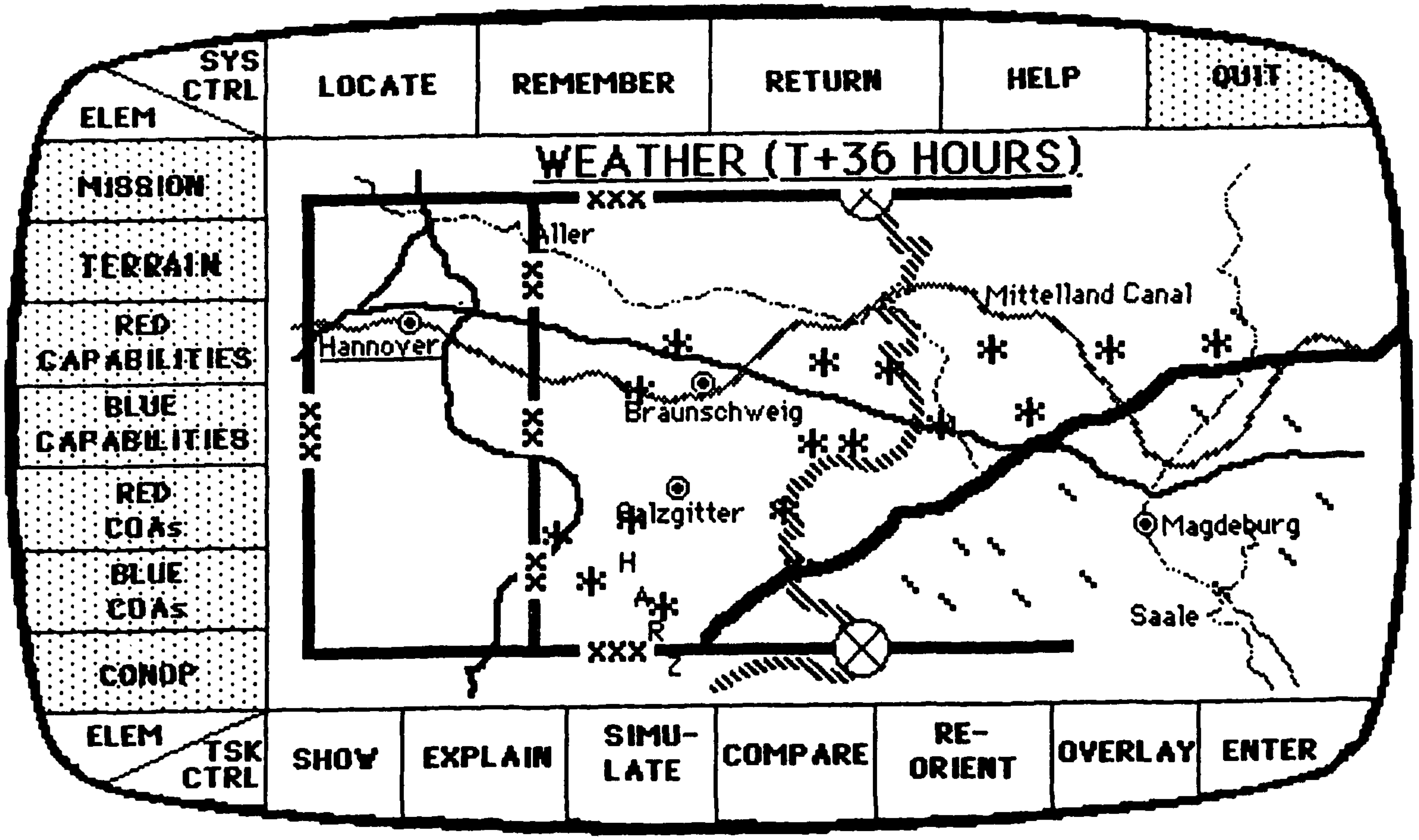
Weather is displayed to the planner -- who then decides to "see" the weather in a simulation (via the **Simulate** command) over time . . .



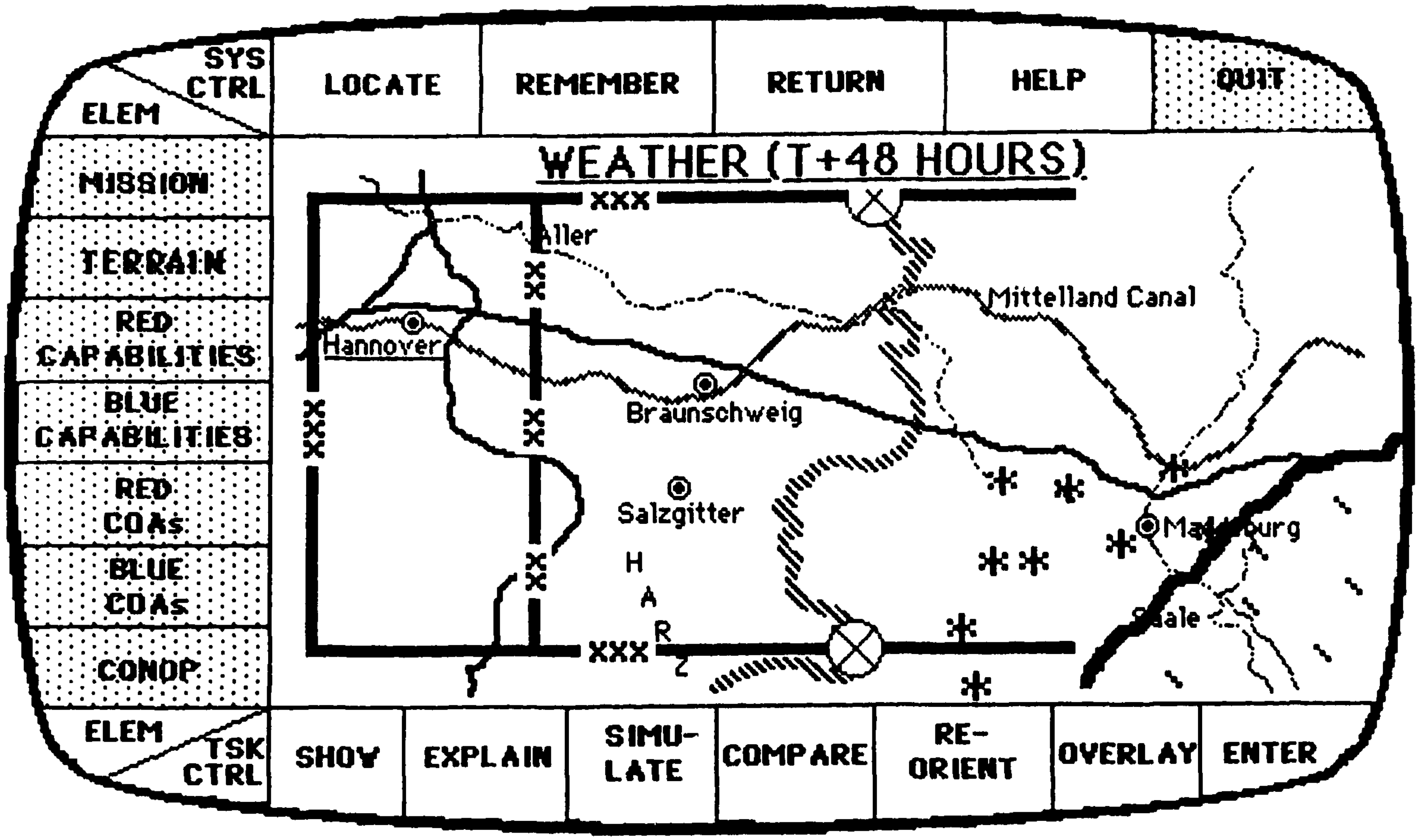
Weather is simulated over time . . .



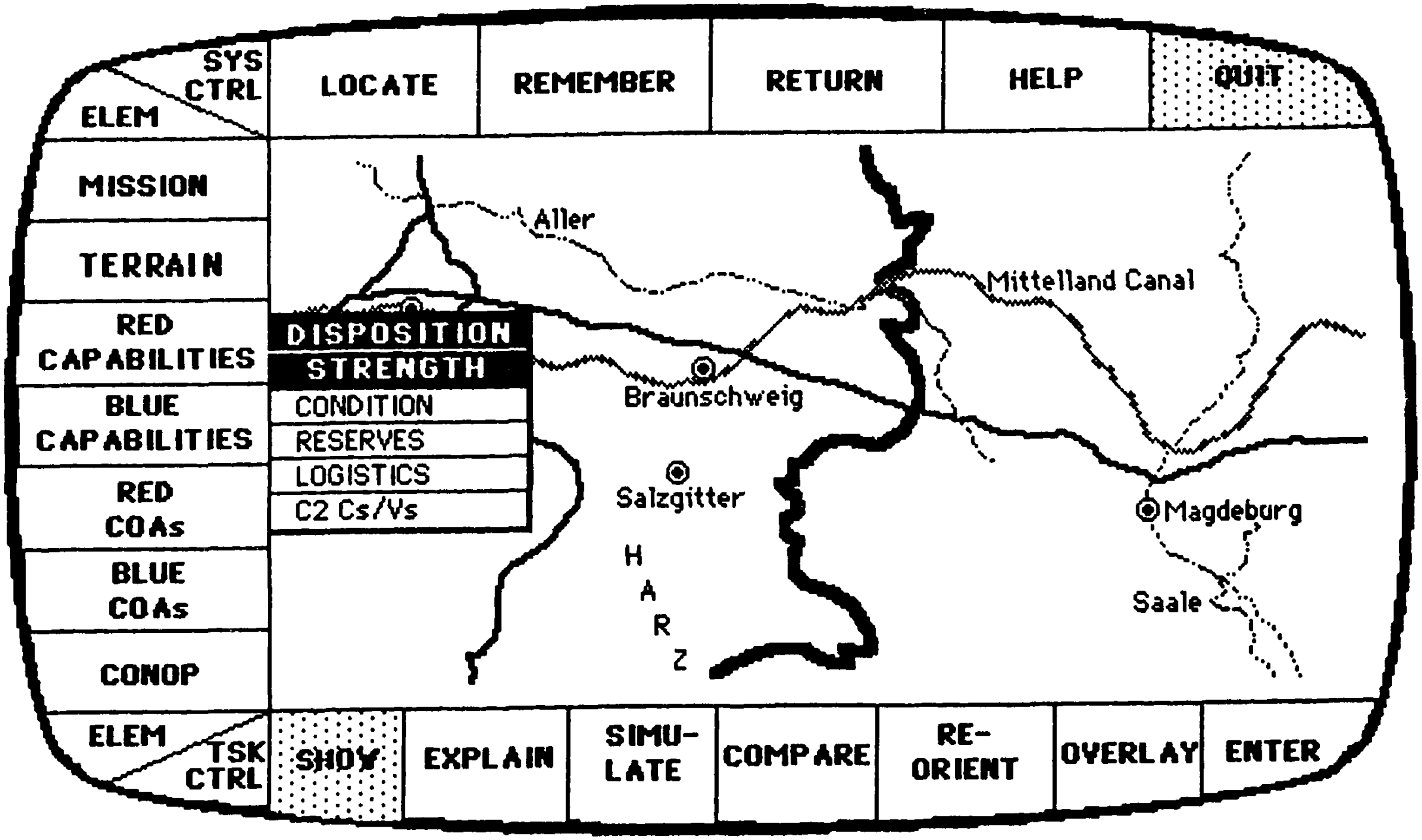
Weather is simulated over time . . .



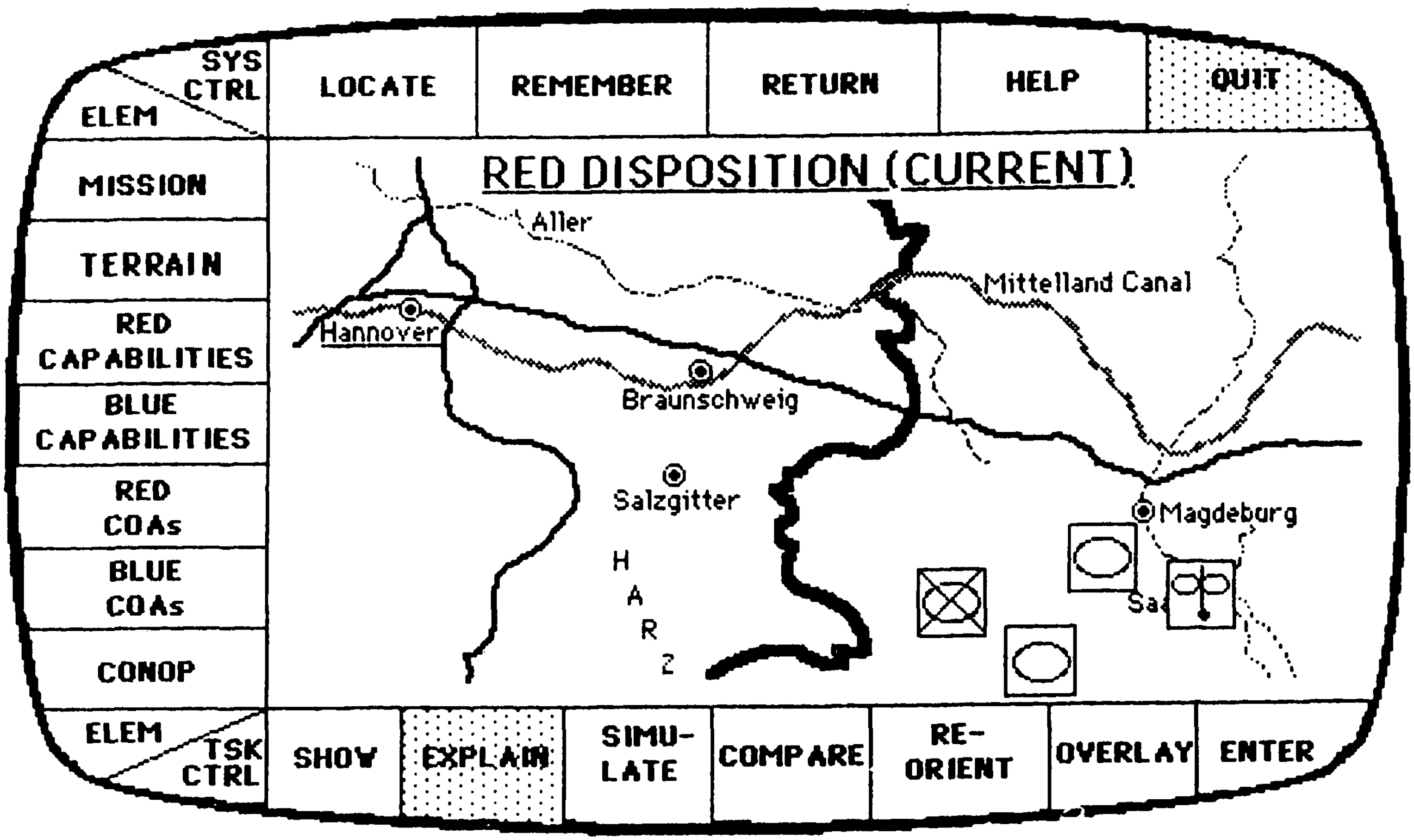
Weather is simulated over time . . .



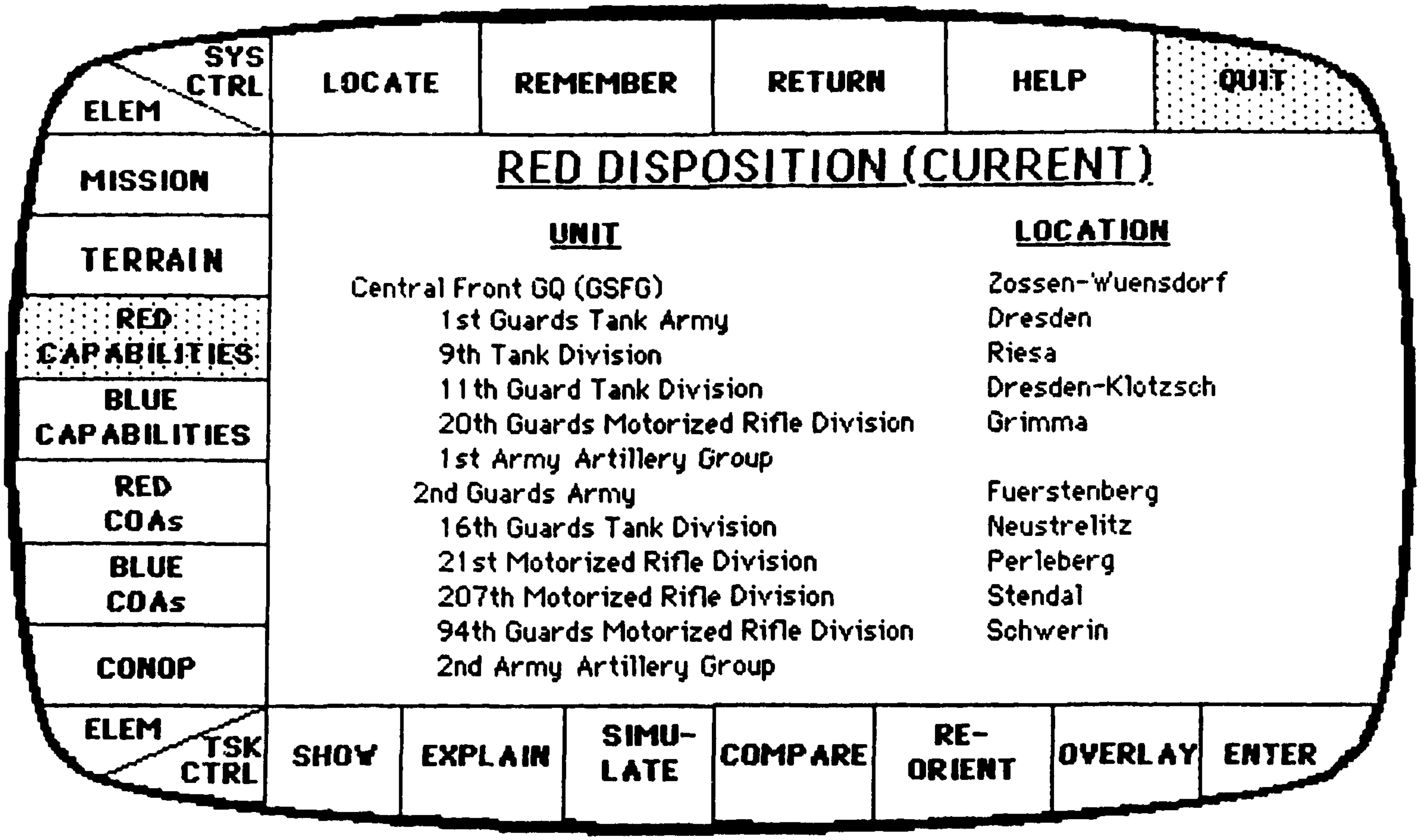
Weather is simulated over time . . .



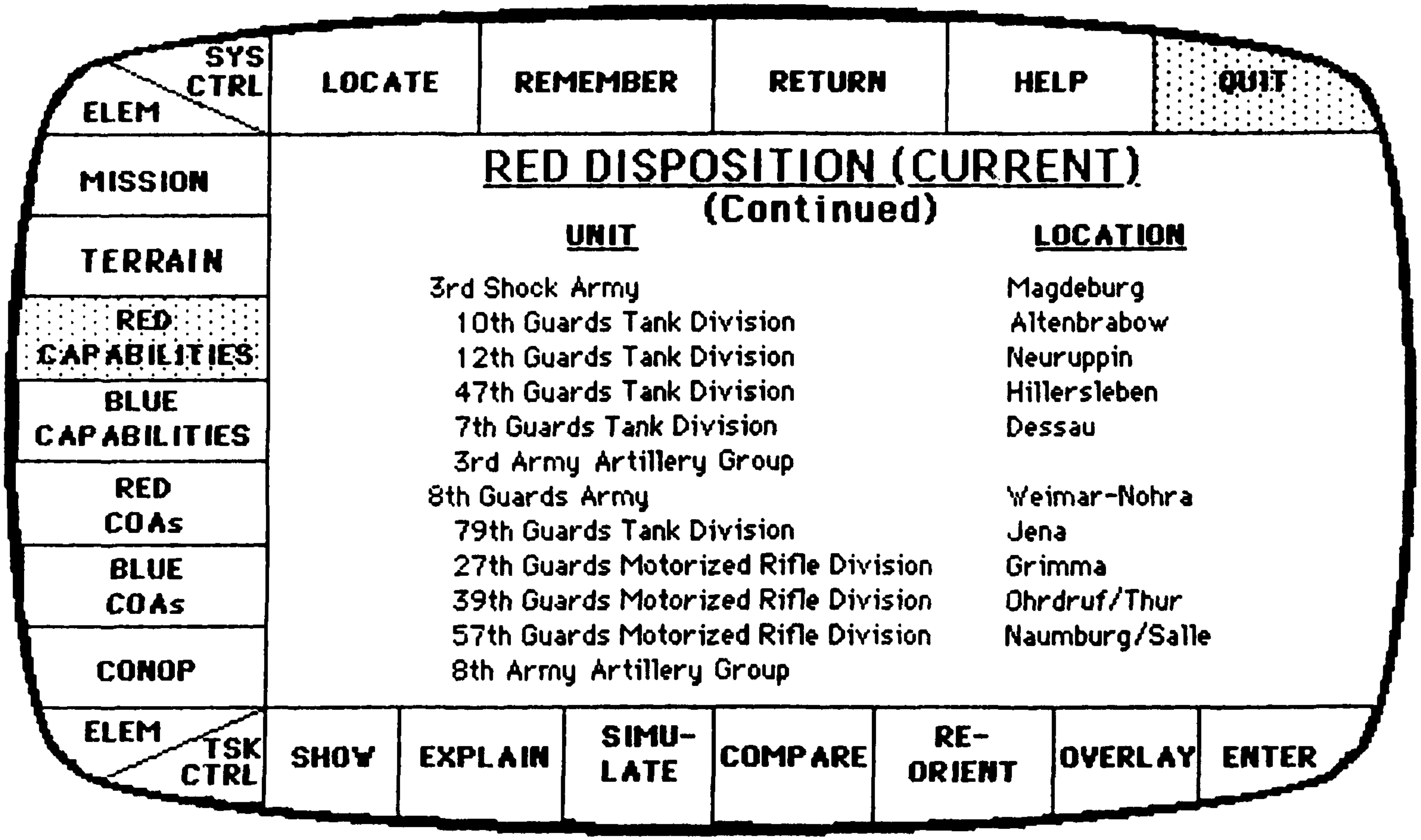
The planner shifts his interest to **Red Capabilities** . . .



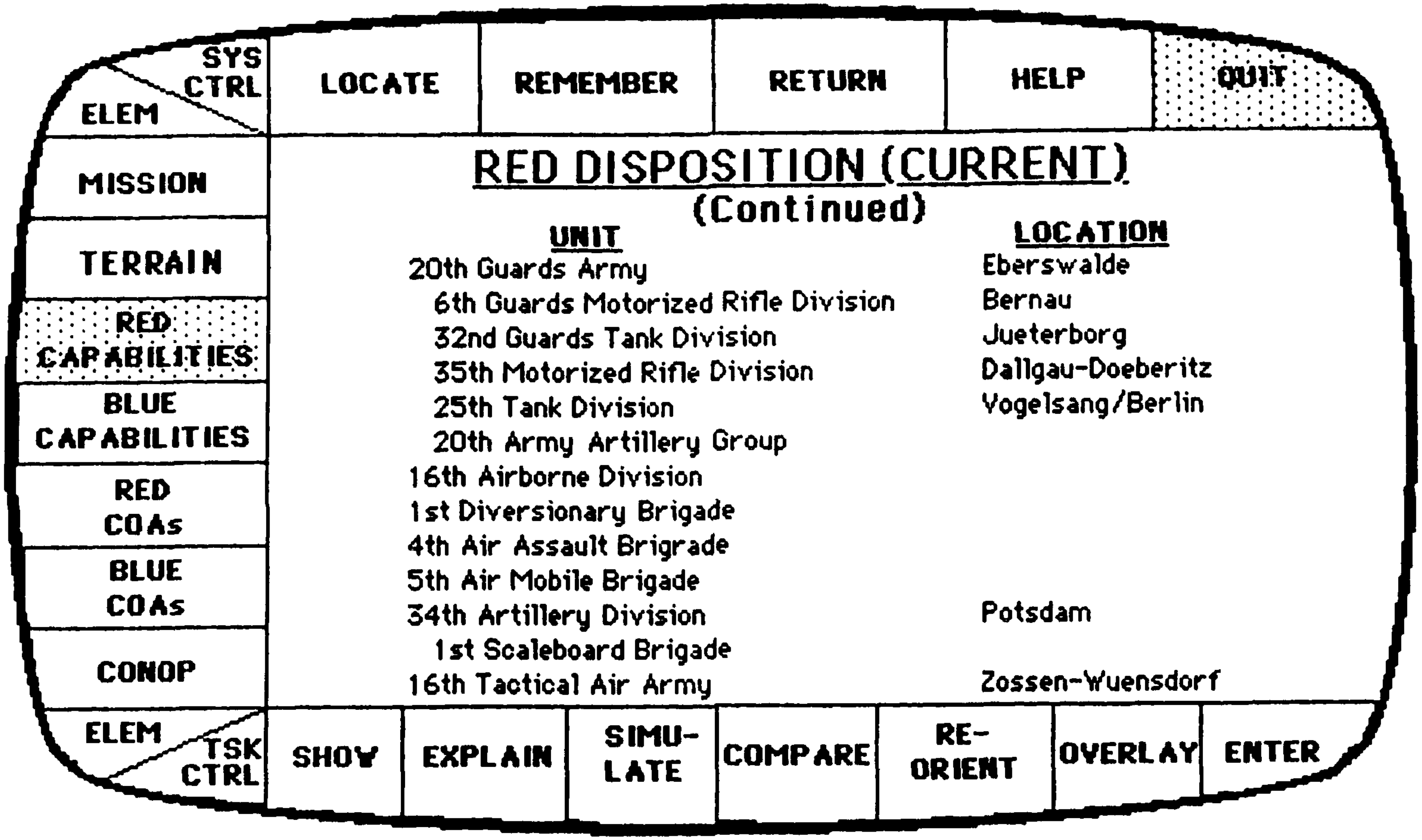
Red Disposition is displayed to the planner. This information is assumed to be received from G2 for use by G3 . . . the system receives the information and then displays it to the G3 in ways meaningful to his planning tasks . . . the planner clicks on **Explain** for additional data . . .



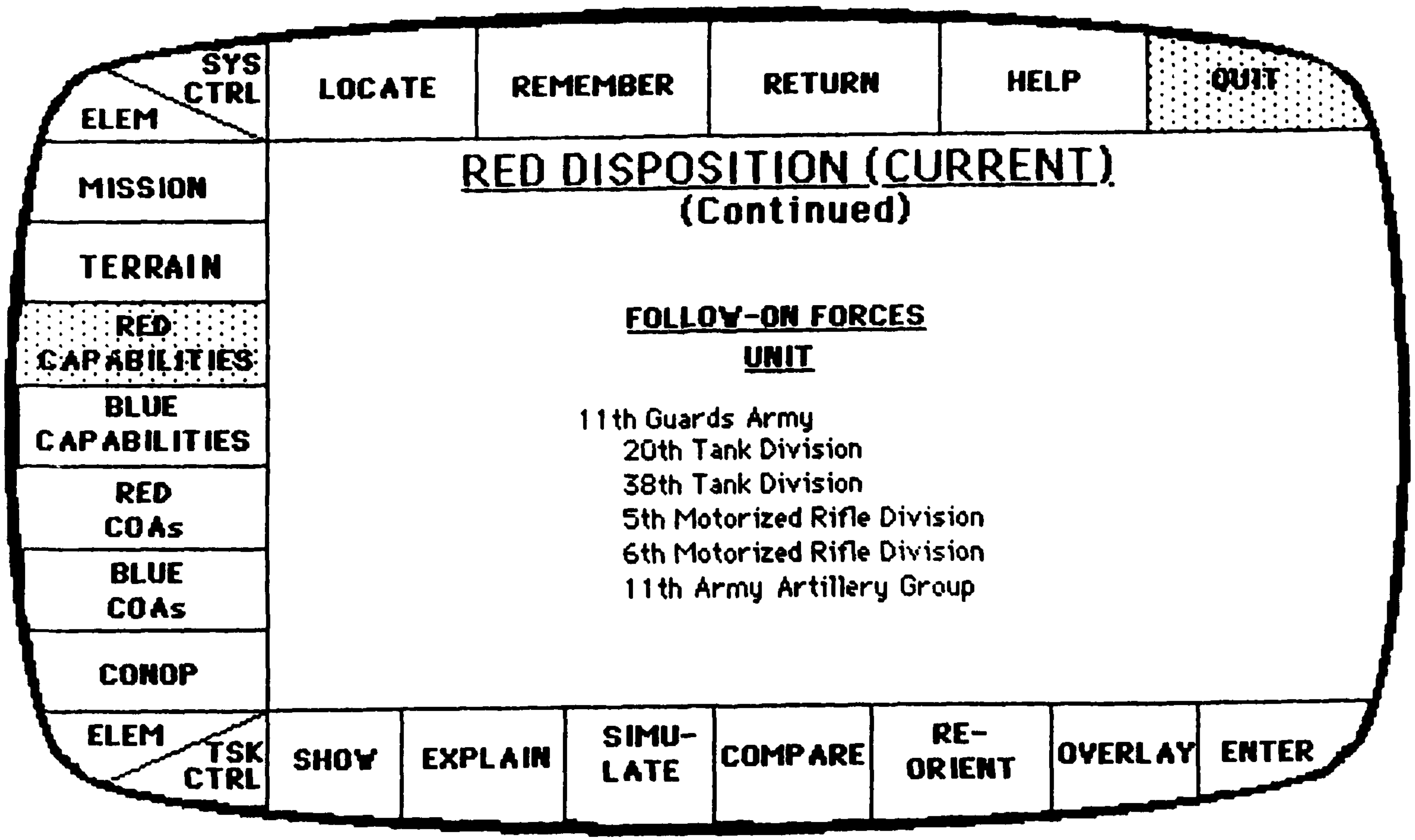
Red Disposition is displayed to the planner alphanumerically . . .



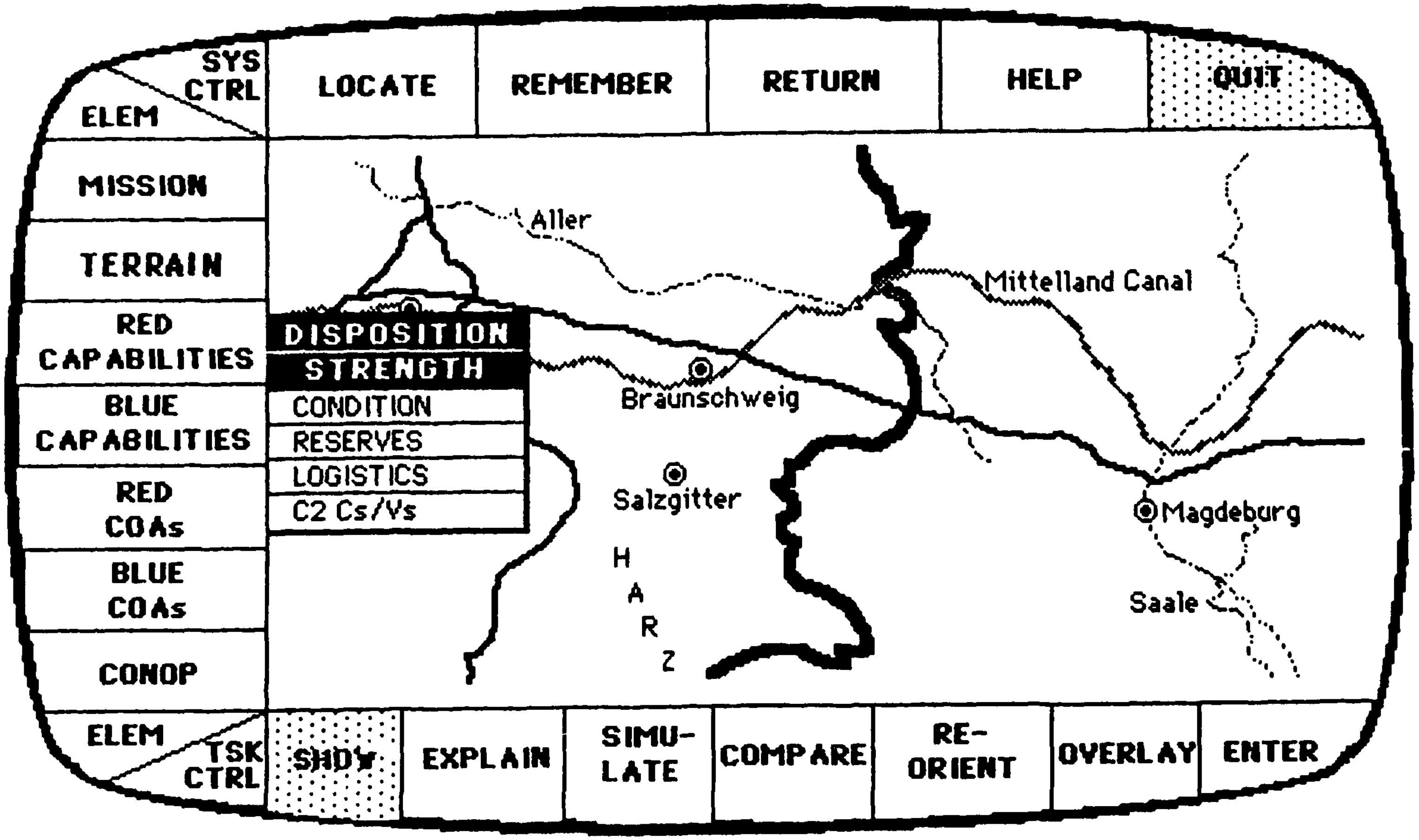
Red Disposition is displayed to the planner alphanumerically . . .



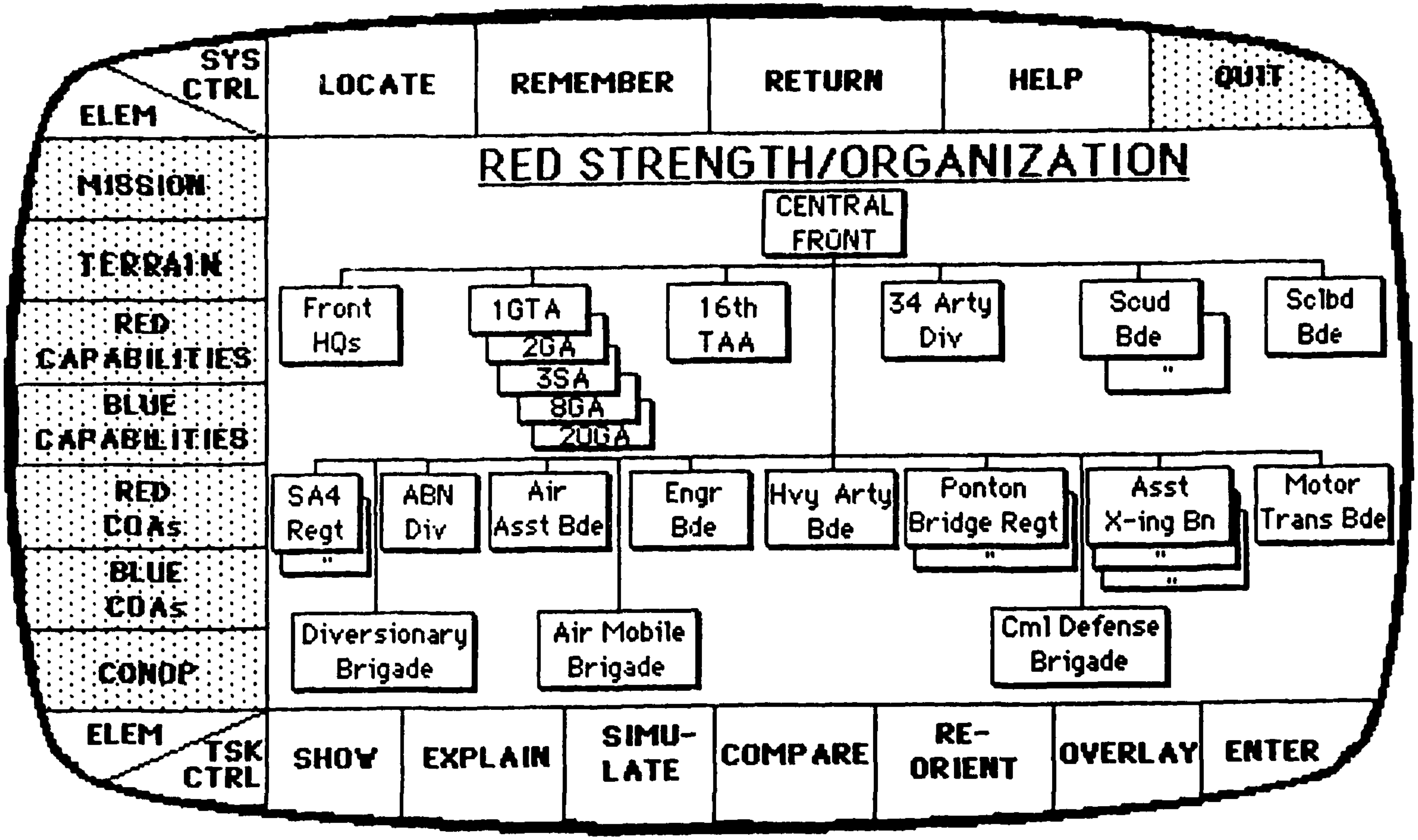
Red Disposition is displayed to the planner alphanumerically . . .



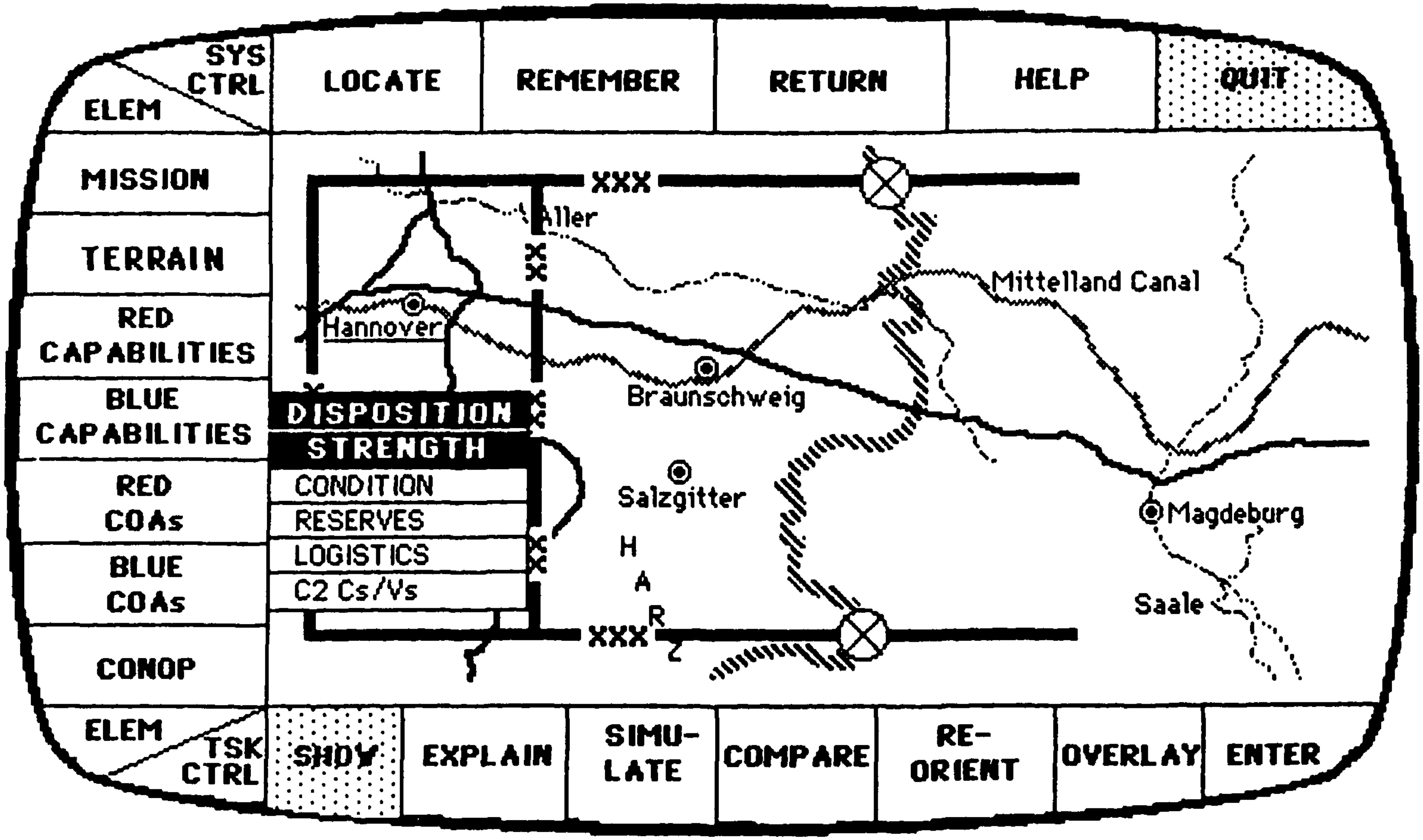
Red Disposition is displayed to the planner alphanumerically . . .



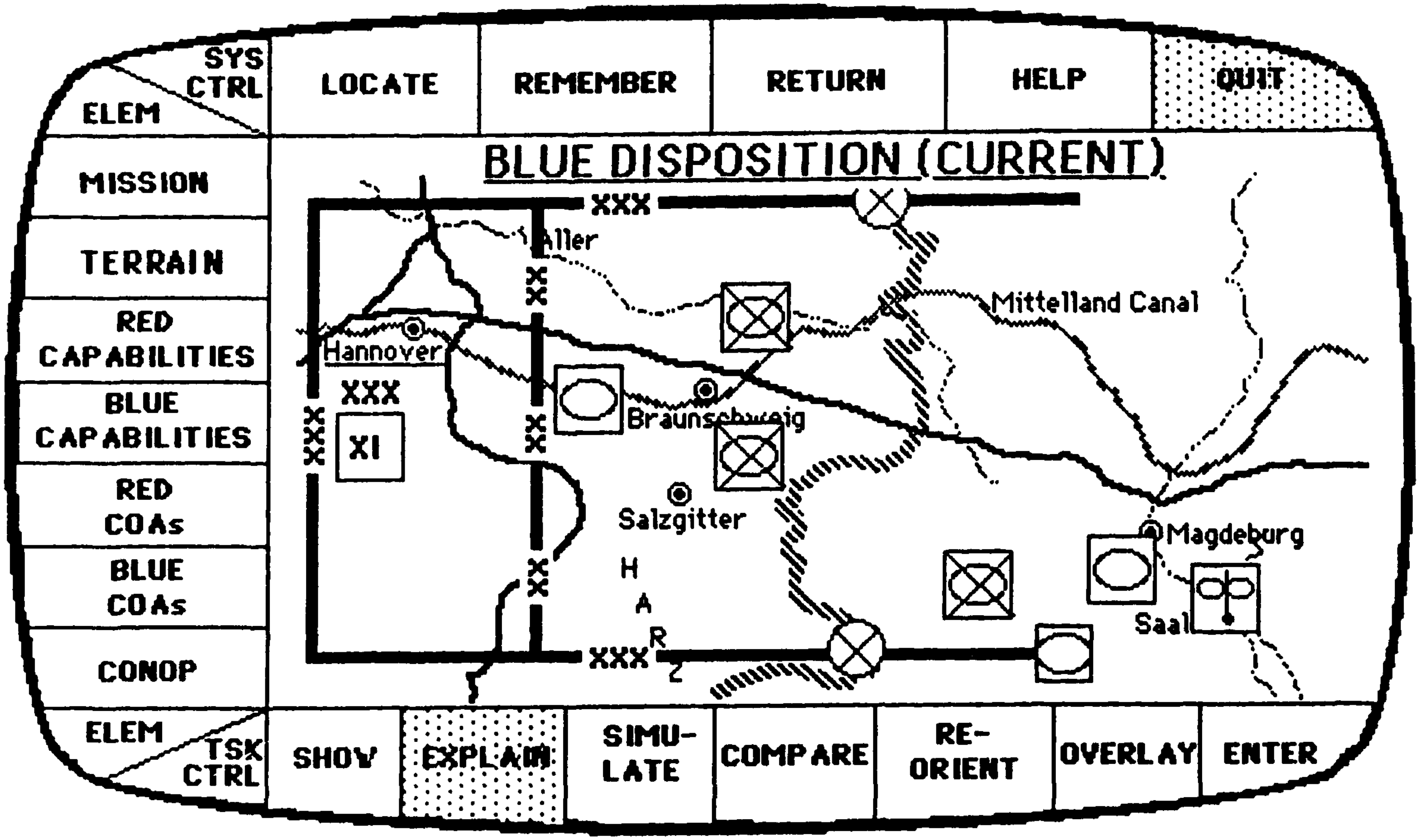
Additional information on **Red Capabilities** is requested . . .



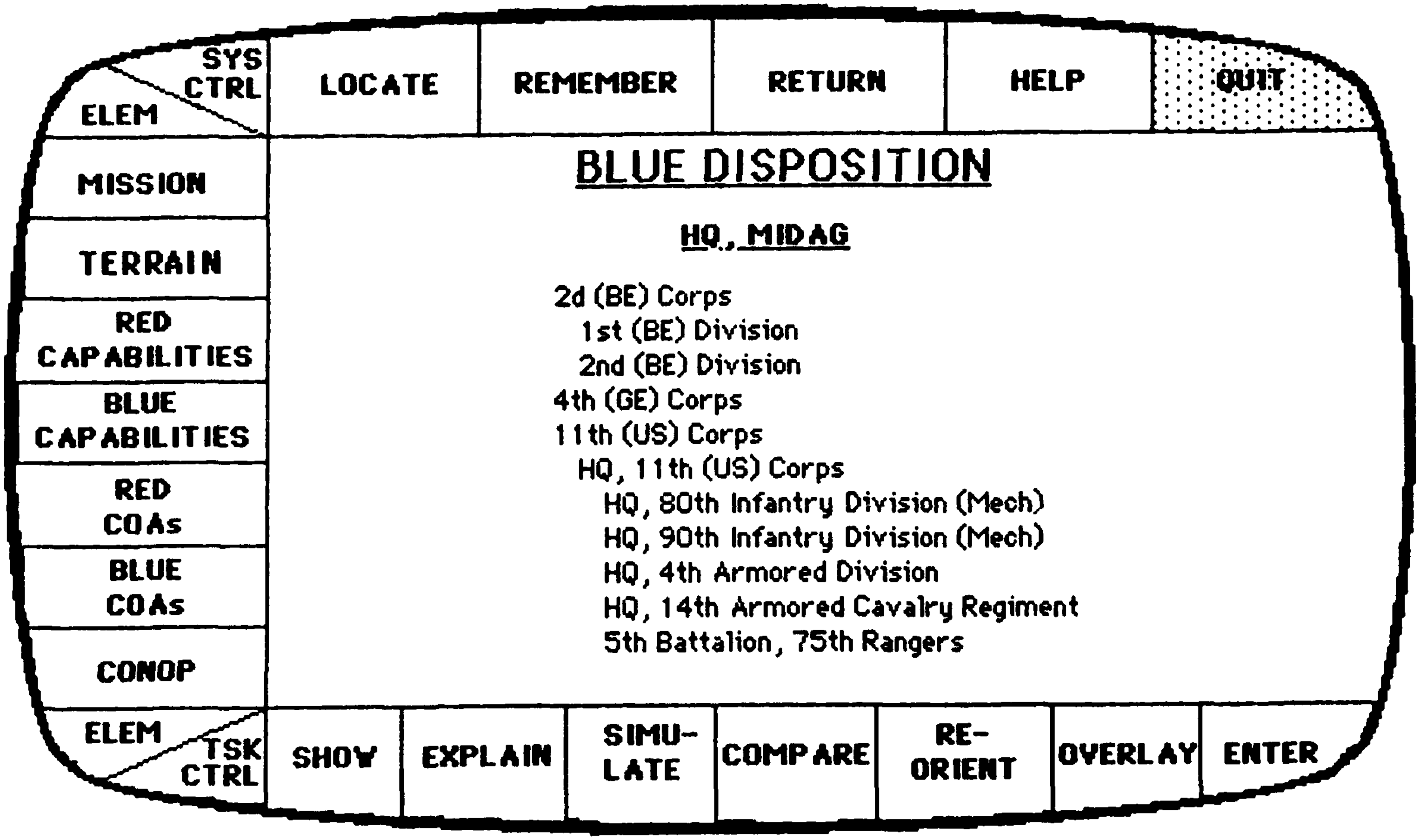
Red Strength and organization is displayed to the user . . .



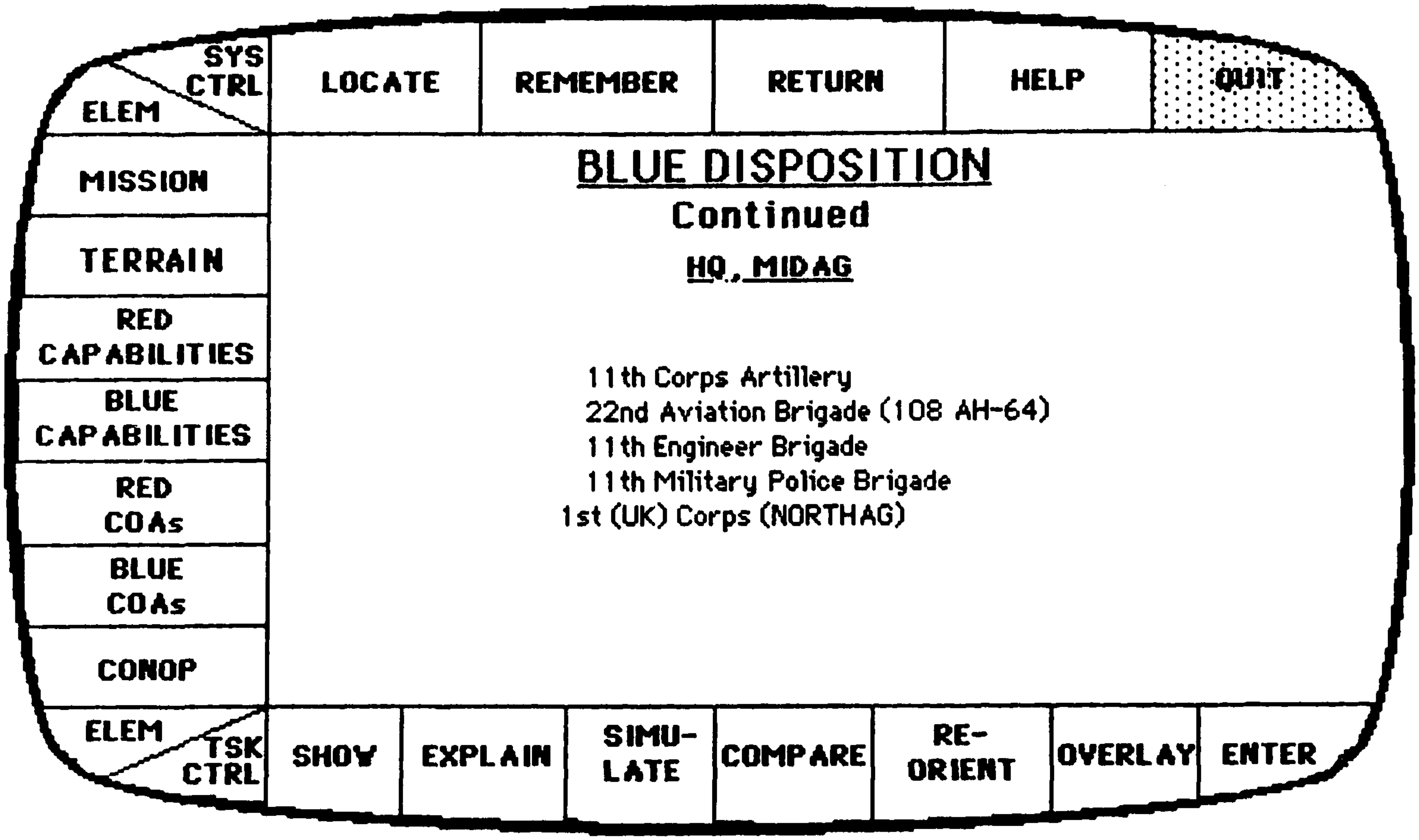
Blue Capabilities are selected . . .



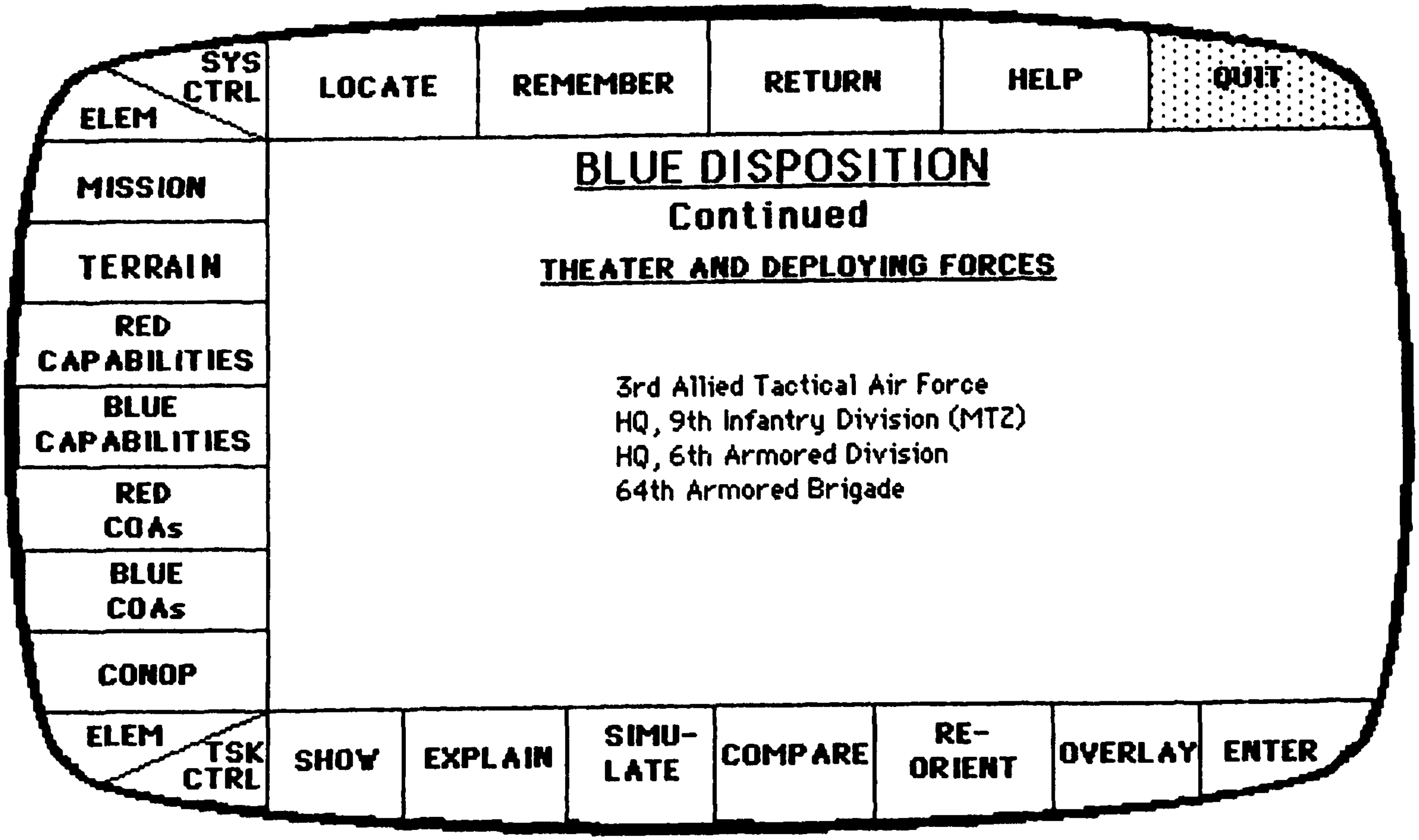
Blue Disposition is displayed . . .



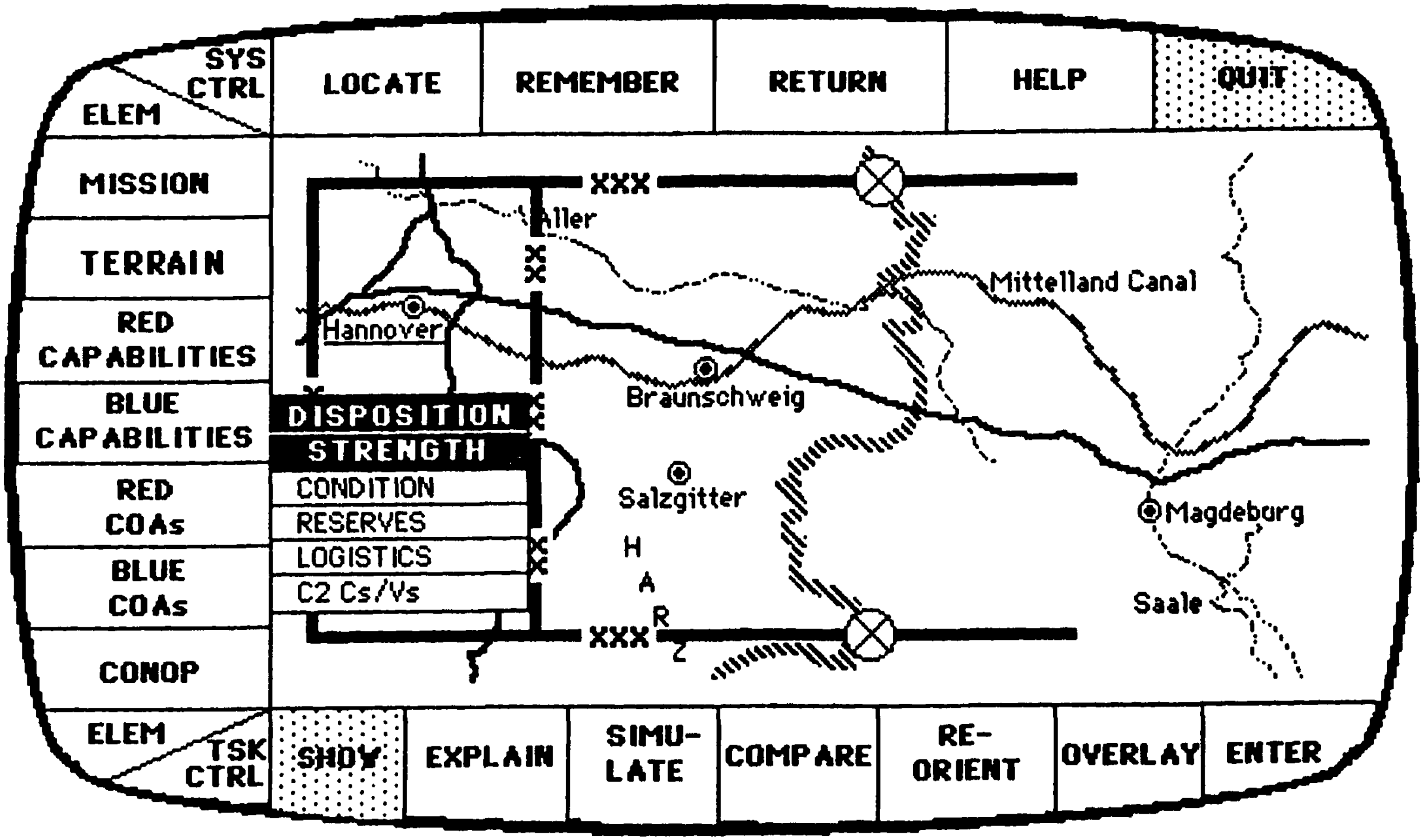
Additional data on **Blue Disposition** is displayed . . .



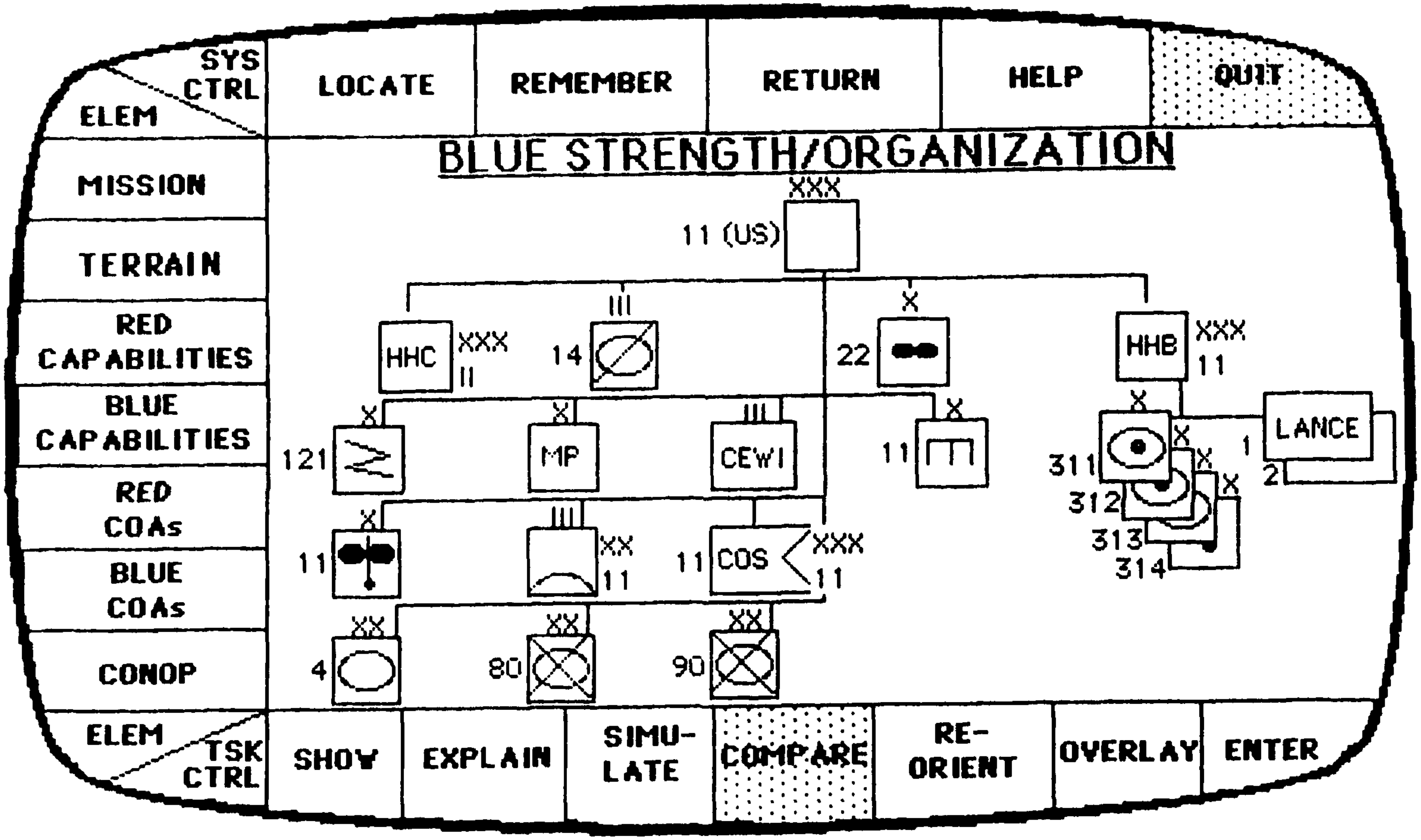
Additional data on **Blue Disposition** is displayed . . .



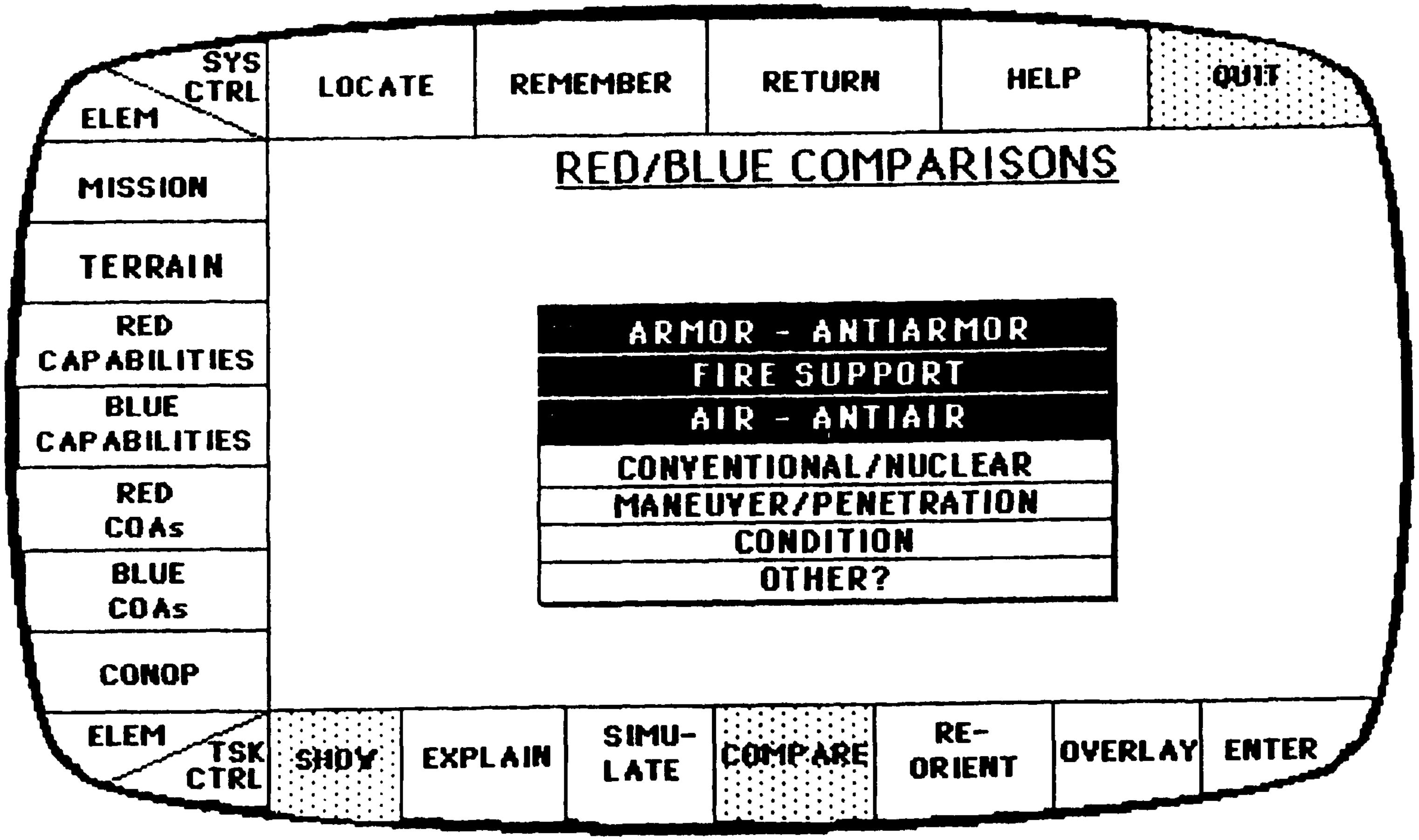
Additional data on **Blue Disposition** is displayed . . .



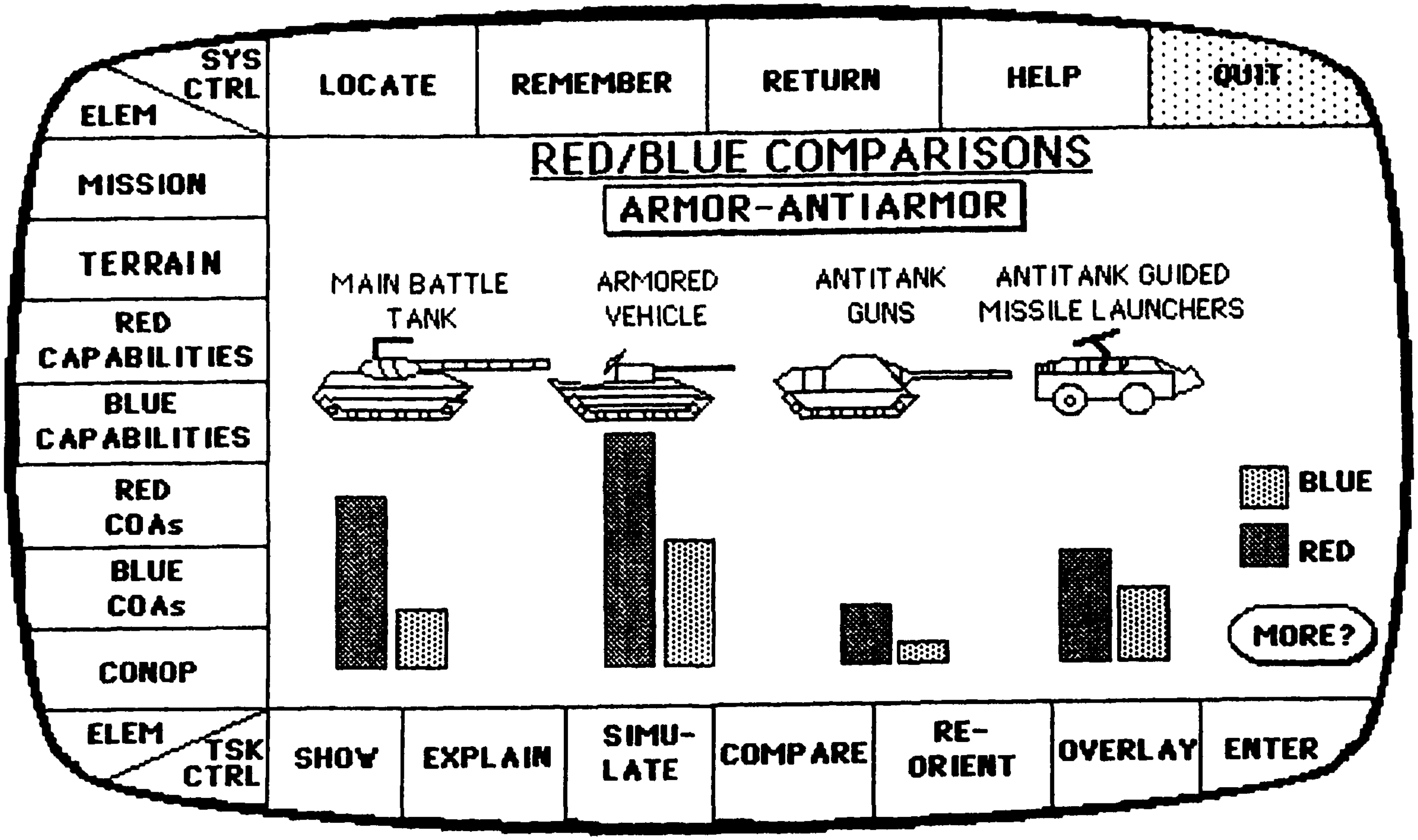
Additional information on **Blue Capabilities** is requested . . .



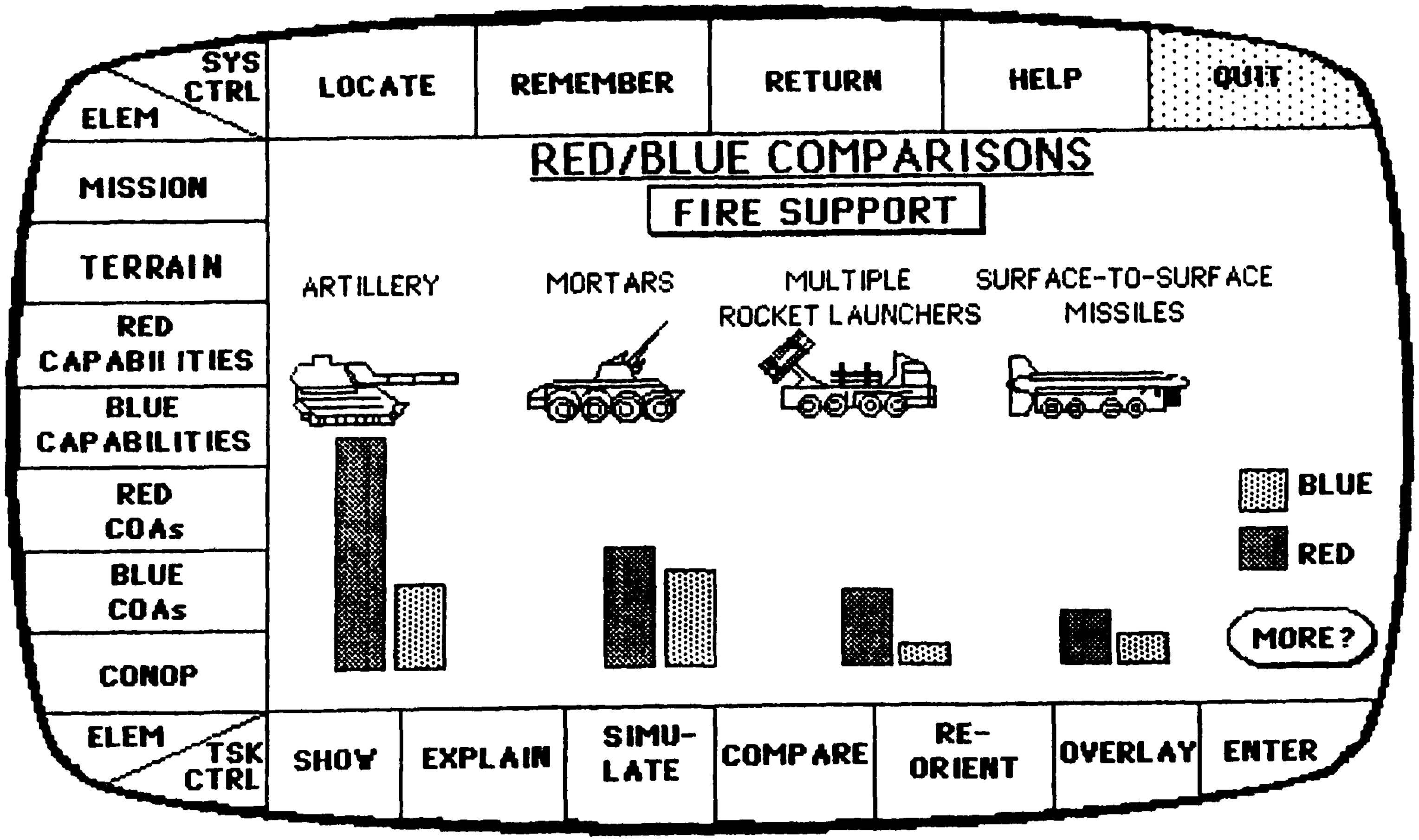
Blue Strength and organization is displayed . . . the planner clicks on the **Compare** option to see how Red and Blue forces compare to one another . .



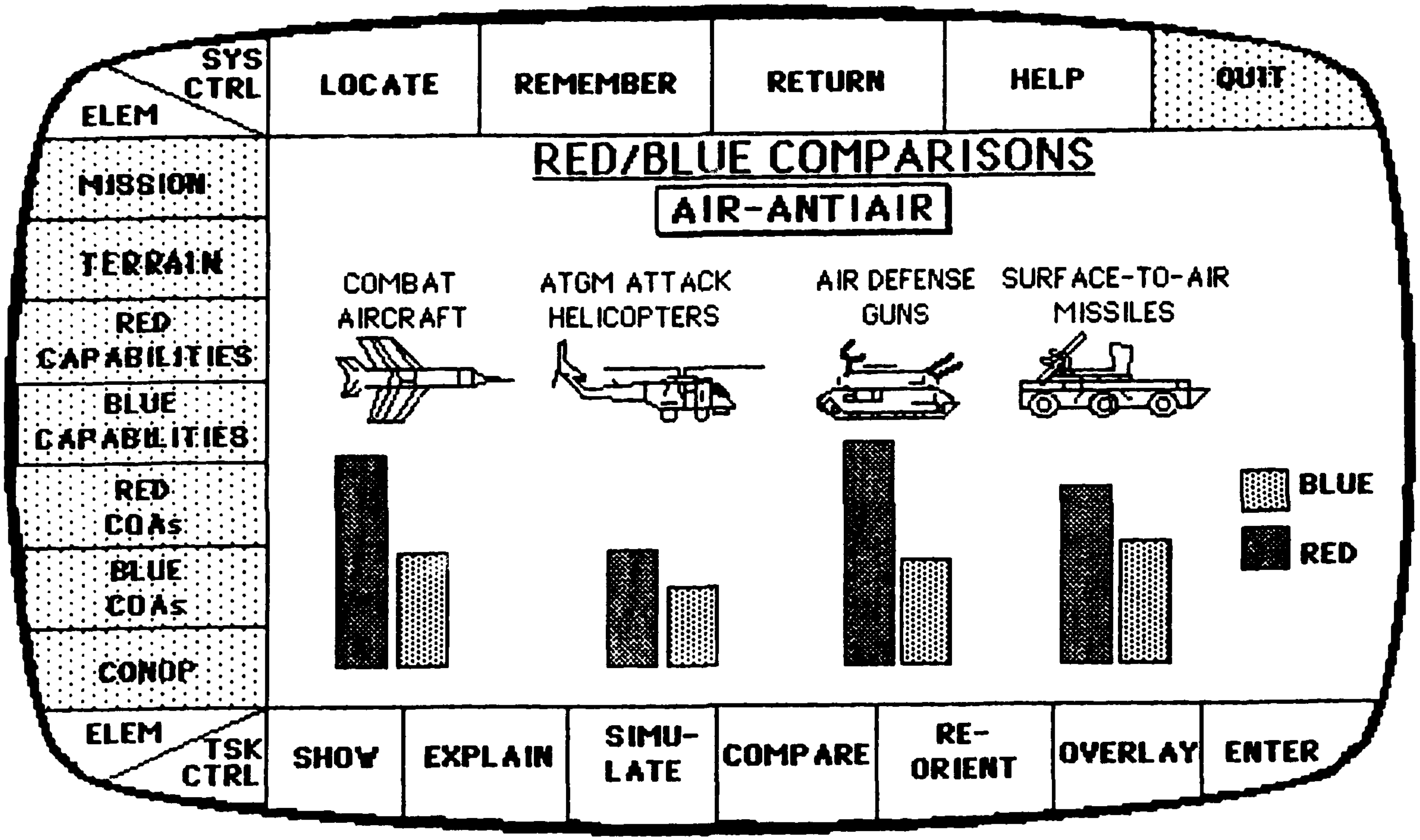
The system calculates strengths and weaknesses based on G2 data and information and provides the user with options for analyzing Red and Blue forces . . . the planner has three options active in this scenario and selects one by asking the system to **Show** the comparative data he has selected



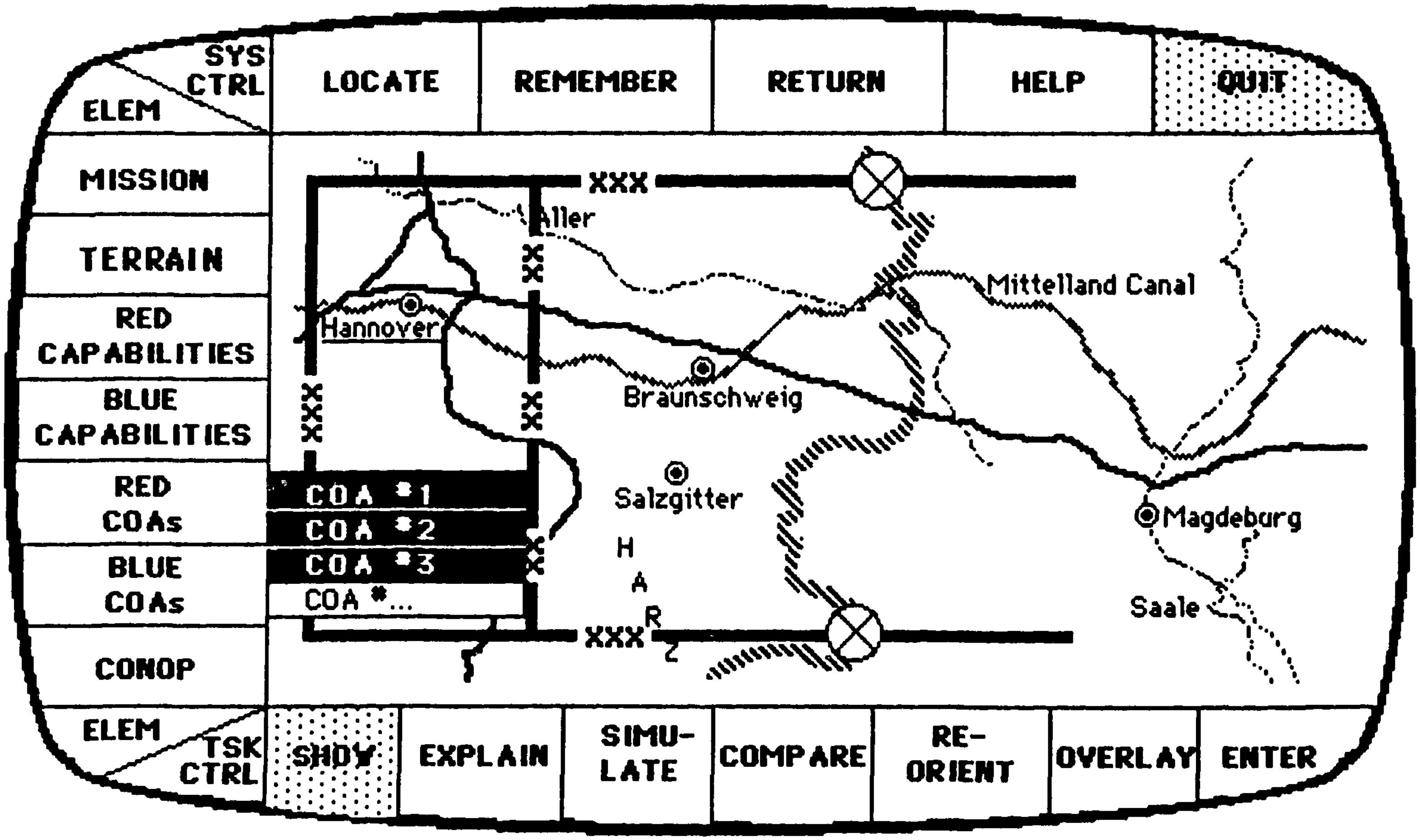
The system displays the data requested graphically. Note that all primary displays are graphic. The use of "analytical graphics" occurs throughout the system concept since our requirements data suggests that planners crave graphic and pictorial data for option generation and evaluation . . . visual cues are used throughout the storyboard to accelerate the planning process. Note also the **More?** menu box on the screen . . .



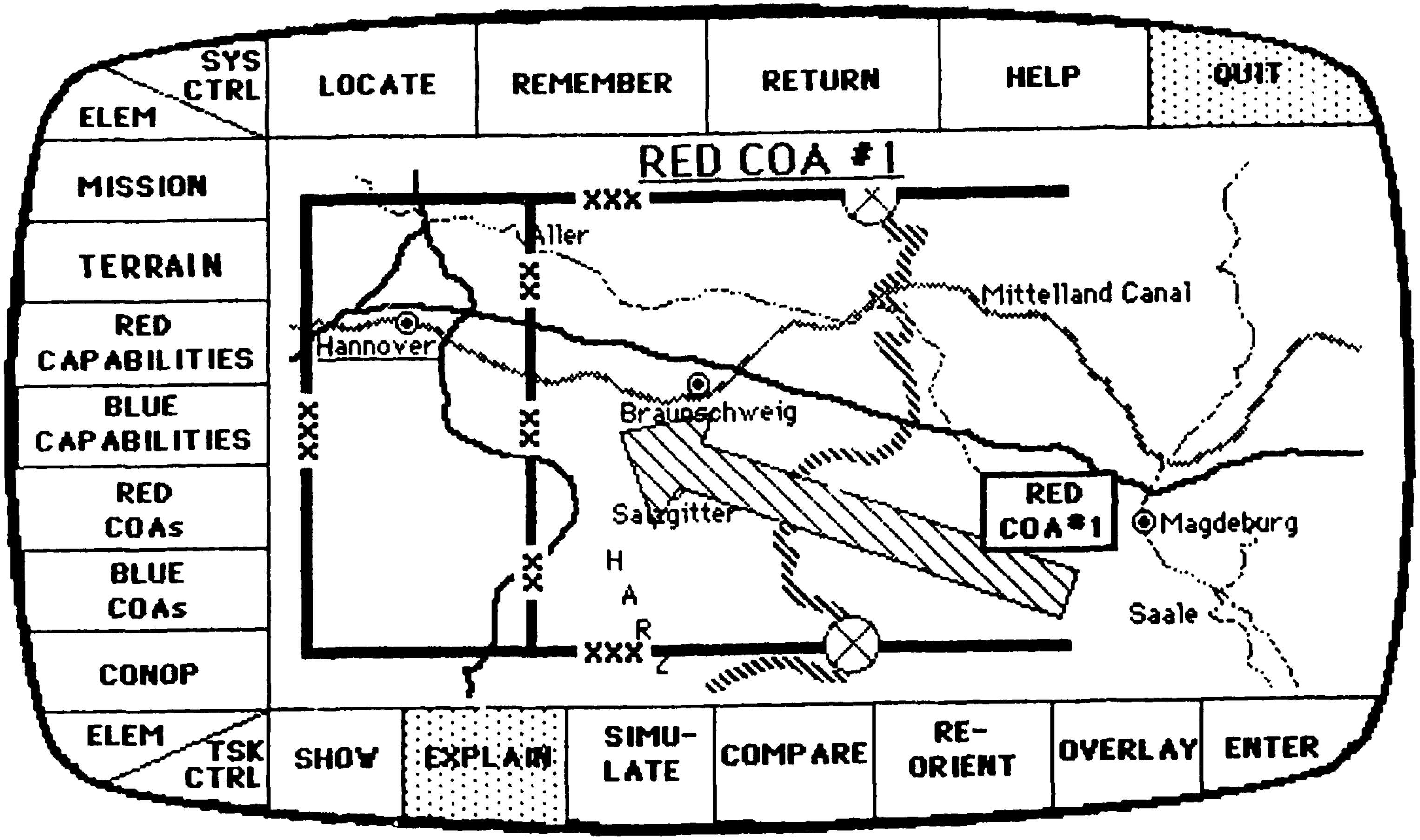
Additional comparative data is displayed . . .



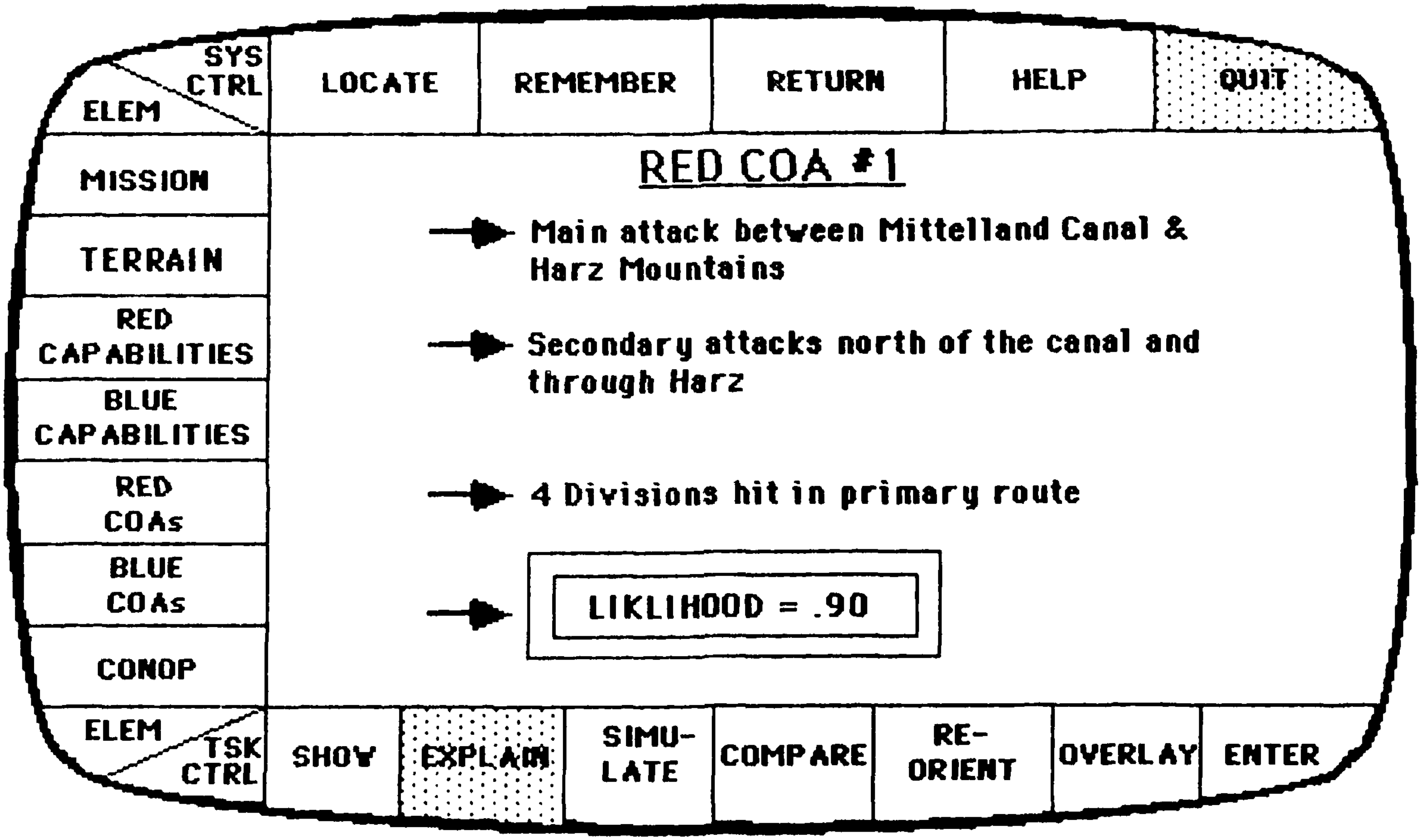
Additional comparative data is displayed . . .



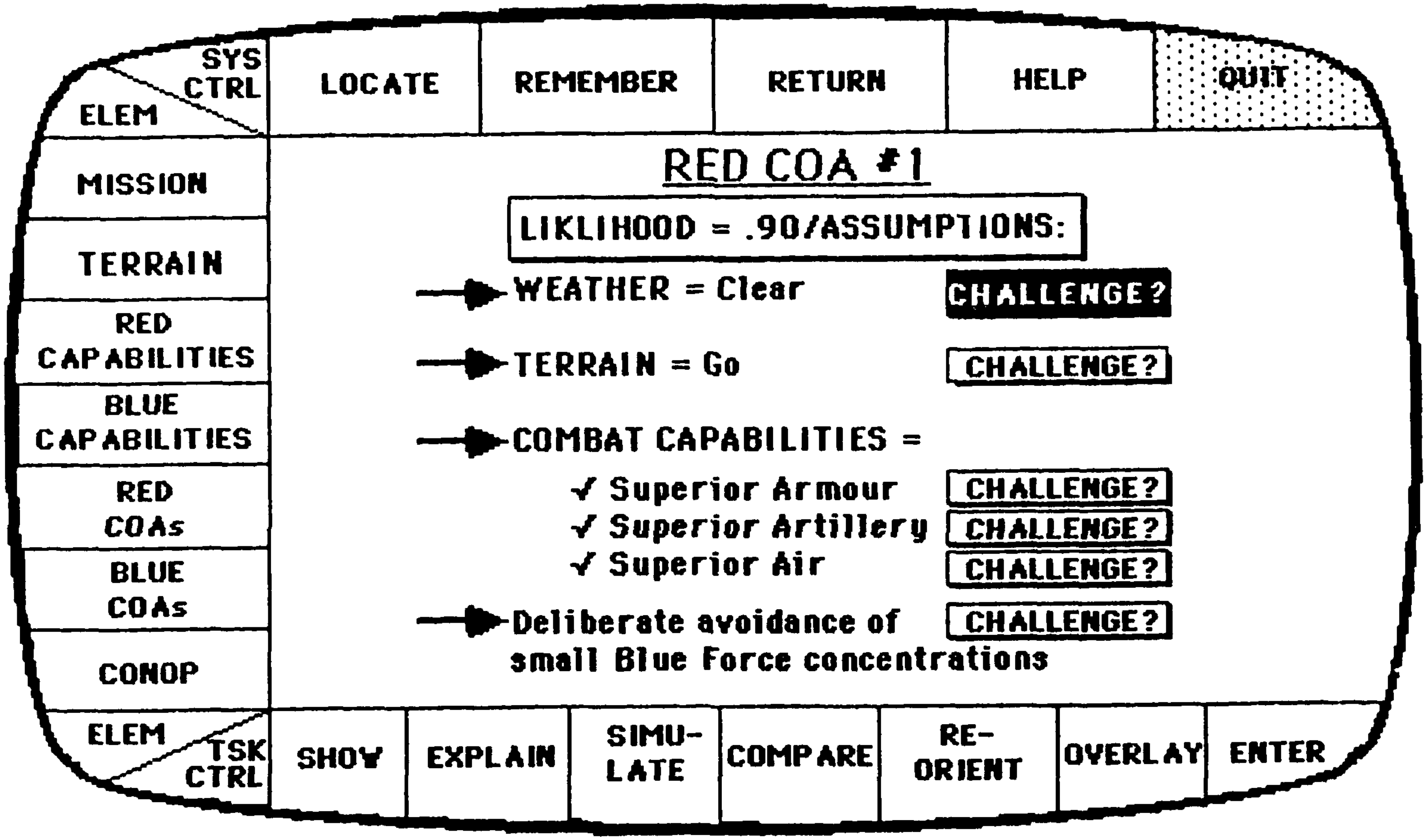
Red Courses of Action (COAs) are requested . . . here the planner is requesting information -- that is, hypotheses -- about where Red is coming from, with what forces, and the like . . .



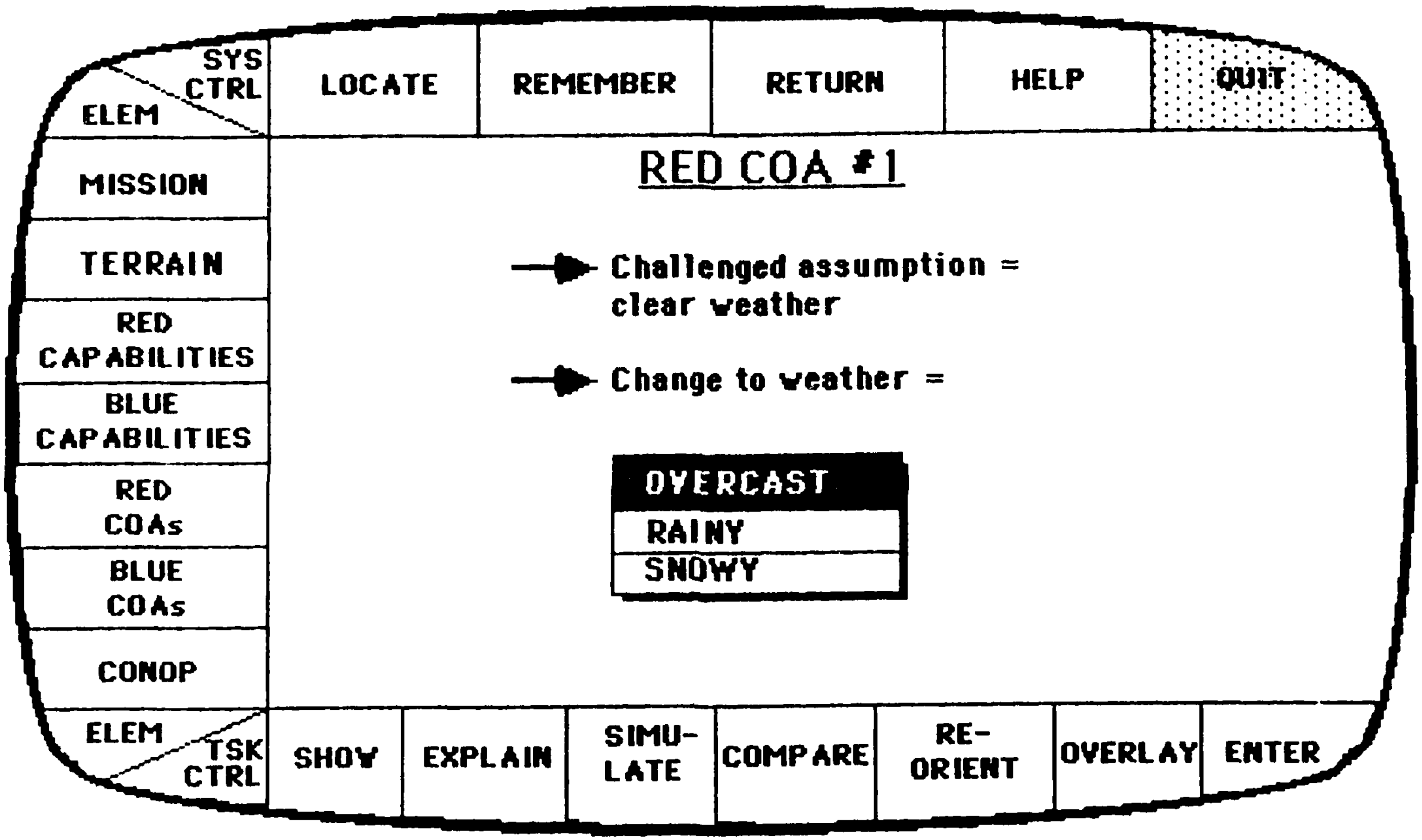
Red COA #1 is displayed to the user; the system generates several candidate "strawmen" for the planner to consider . . . **COA #1** is the first system-generated hypothesis about where Red might be coming from . . . this system concept assumes that planners are better at scrutinizing options than they are at generating them. A system-resident knowledge base generates the candidate plans for the user to accept, reject, or modify . . . the planner clicks on **Explain** for additional information . . .



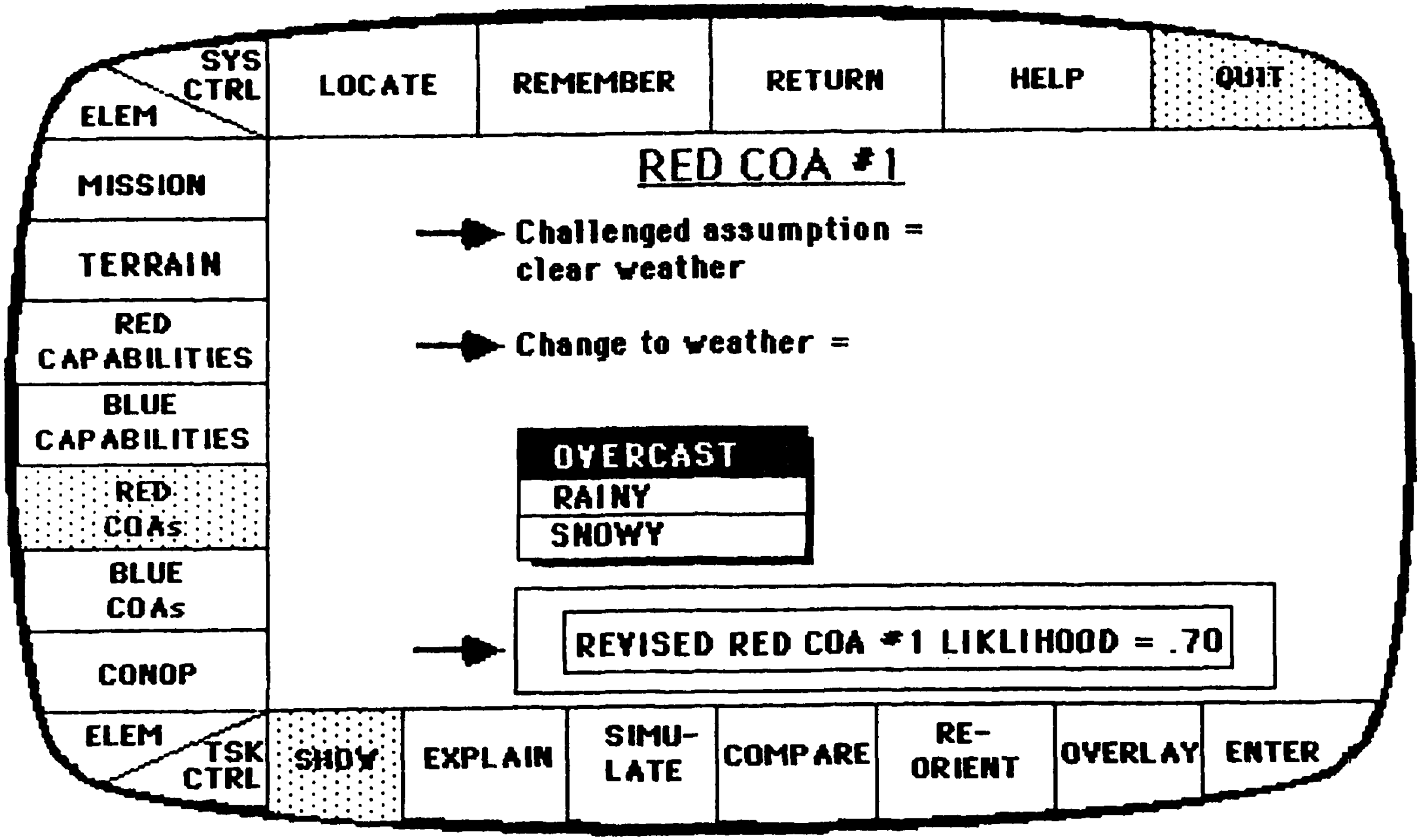
The system displays an explanation of the **Red COA #1** and generates a likelihood, a probability of the course of action for the planner to consider . . . in this system scenario the planner asks for yet more explanations . . . explanatory data can be stored hierarchically where descending levels in the hierarchy contain increasingly specific and detailed information . . .



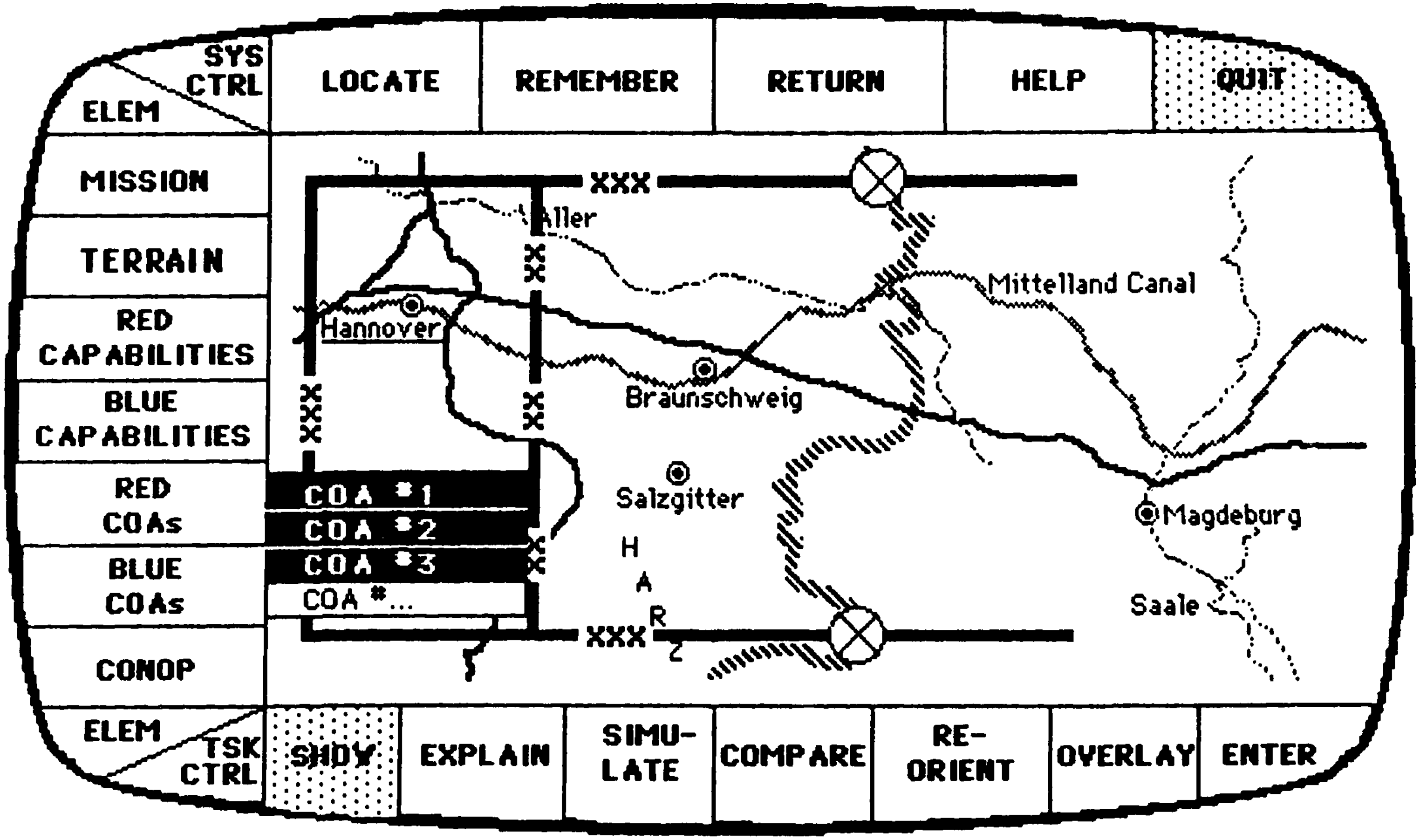
The system presents the planner with the bases upon which the probability was generated . . . each of the system's assumptions (derived, in turn, from its knowledge base) is explained to the planner. He is presented with the option of **Challenging** each assumption. In this case, the user decides that what he thinks he knows about one of the assumptions -- weather -- cannot be validated (perhaps because he has just received additional information from G2) . . .



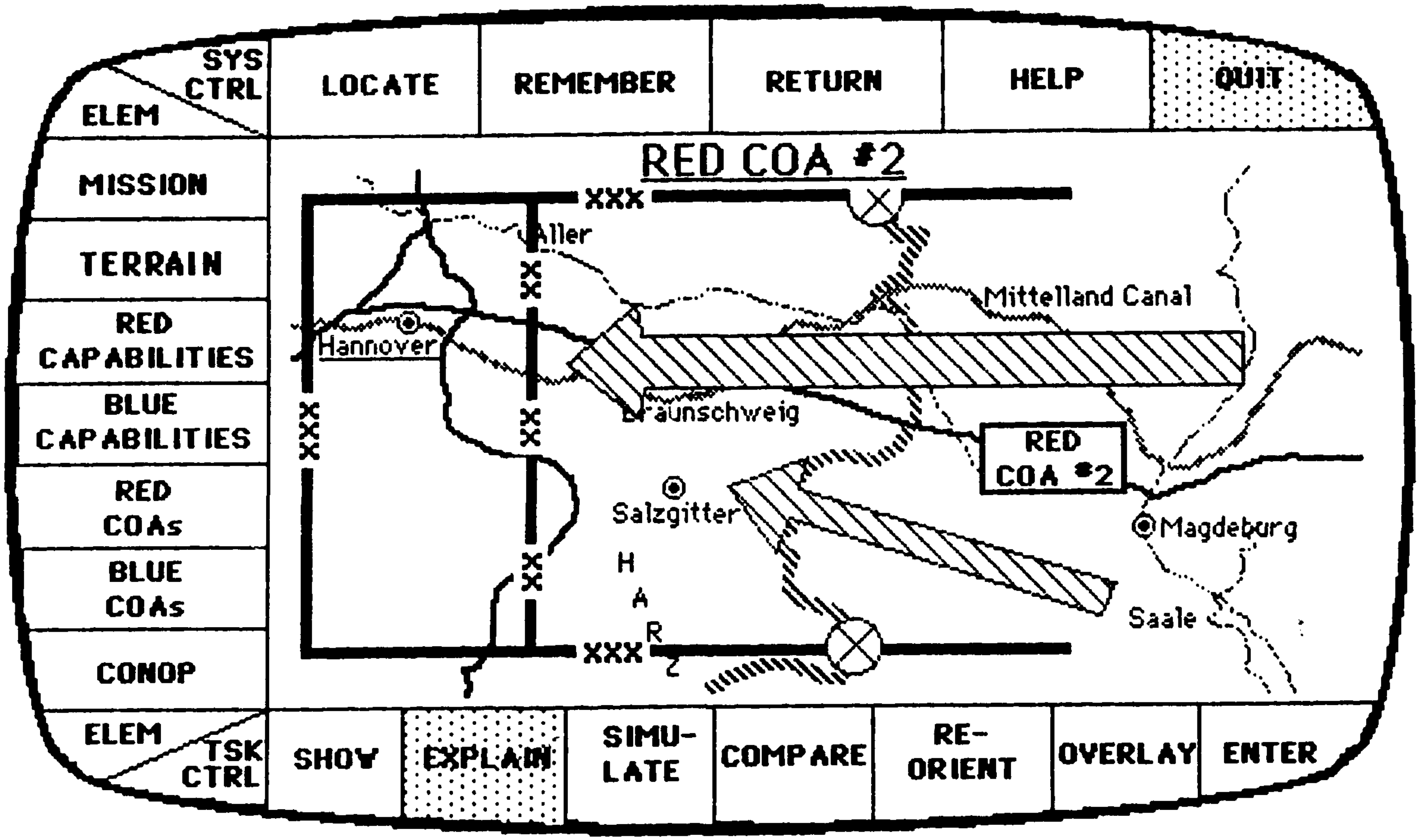
The system permits the planner to change the assumption . . . he changes **Weather** from "clear" to "overcast" . . .



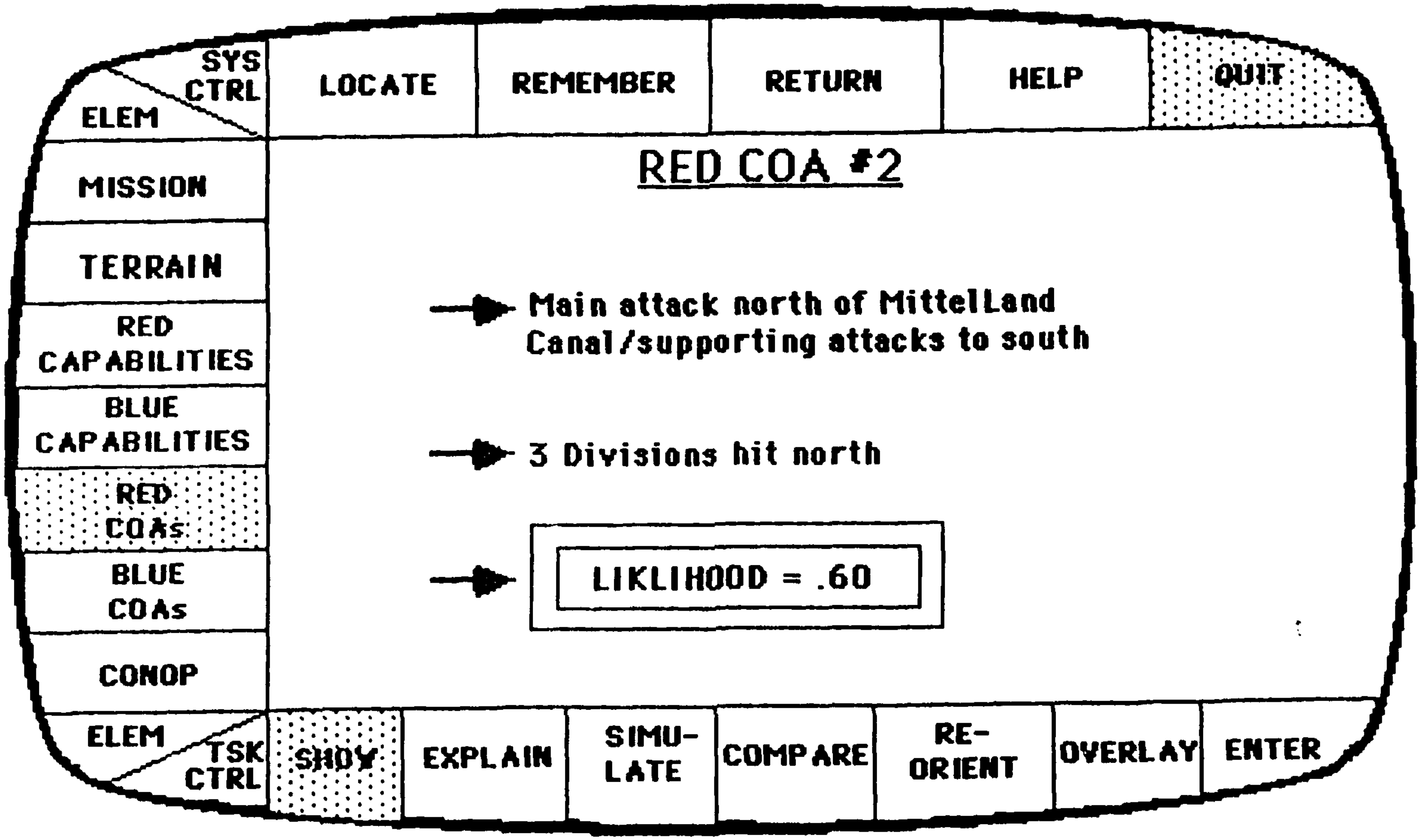
The system responds to the change by changing the probability of the Red COA #1 from .90 to .70. The planner can then change additional assumptions or move on to look at additional system-generated COAs . . .



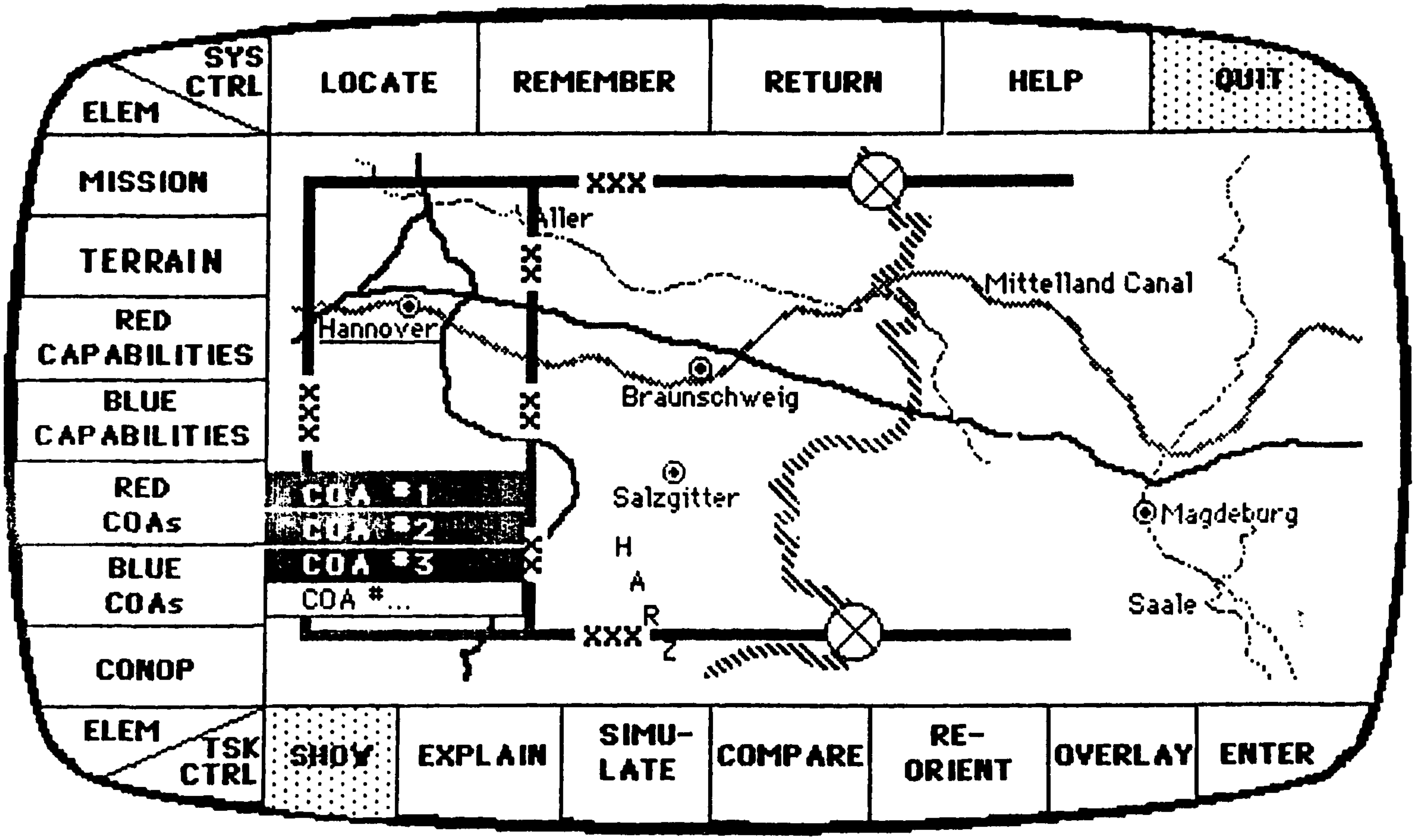
The user then selects additional data about **Red COAs** . . .



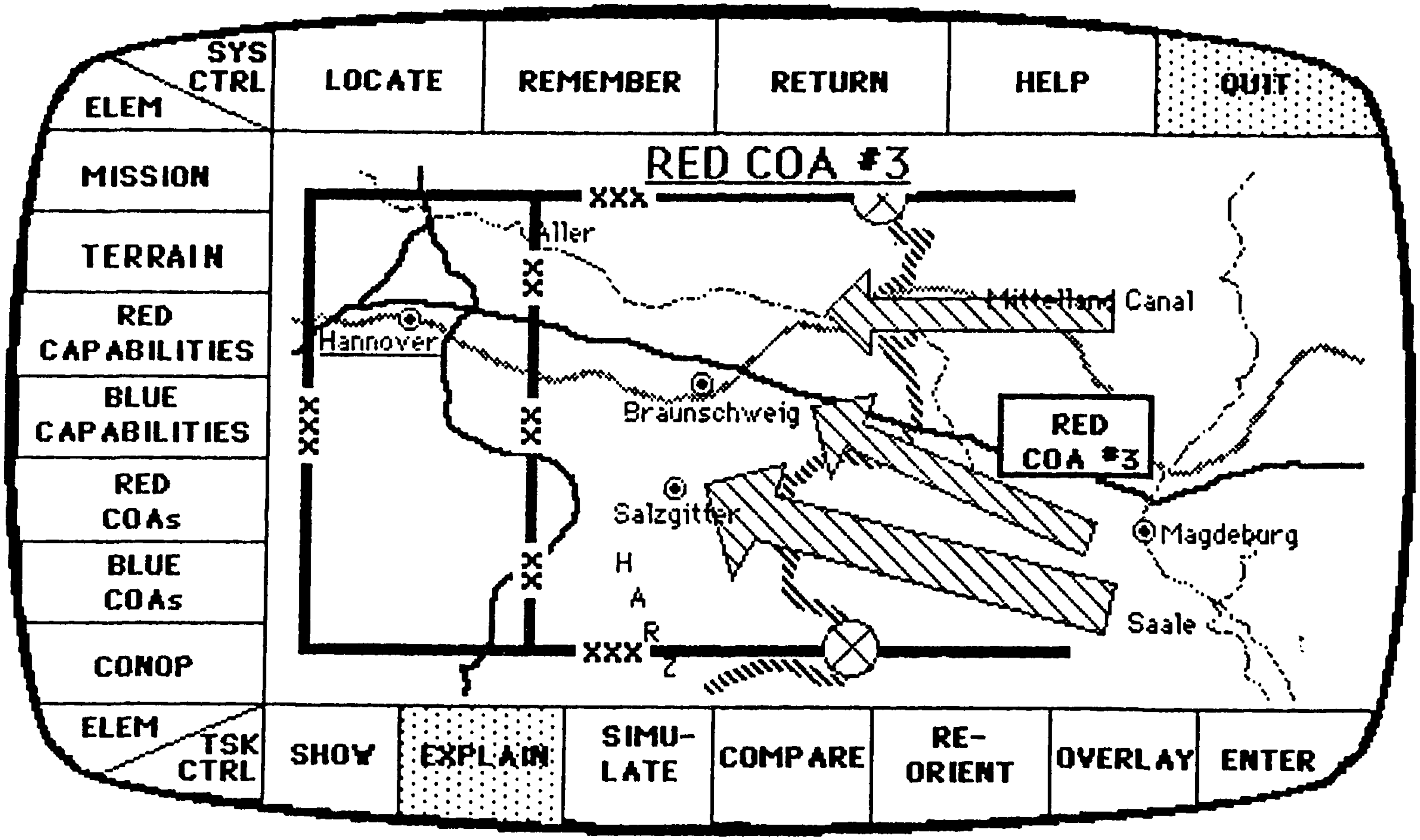
Red COA #2 is displayed . . . the user clicks on **Explain** . . .



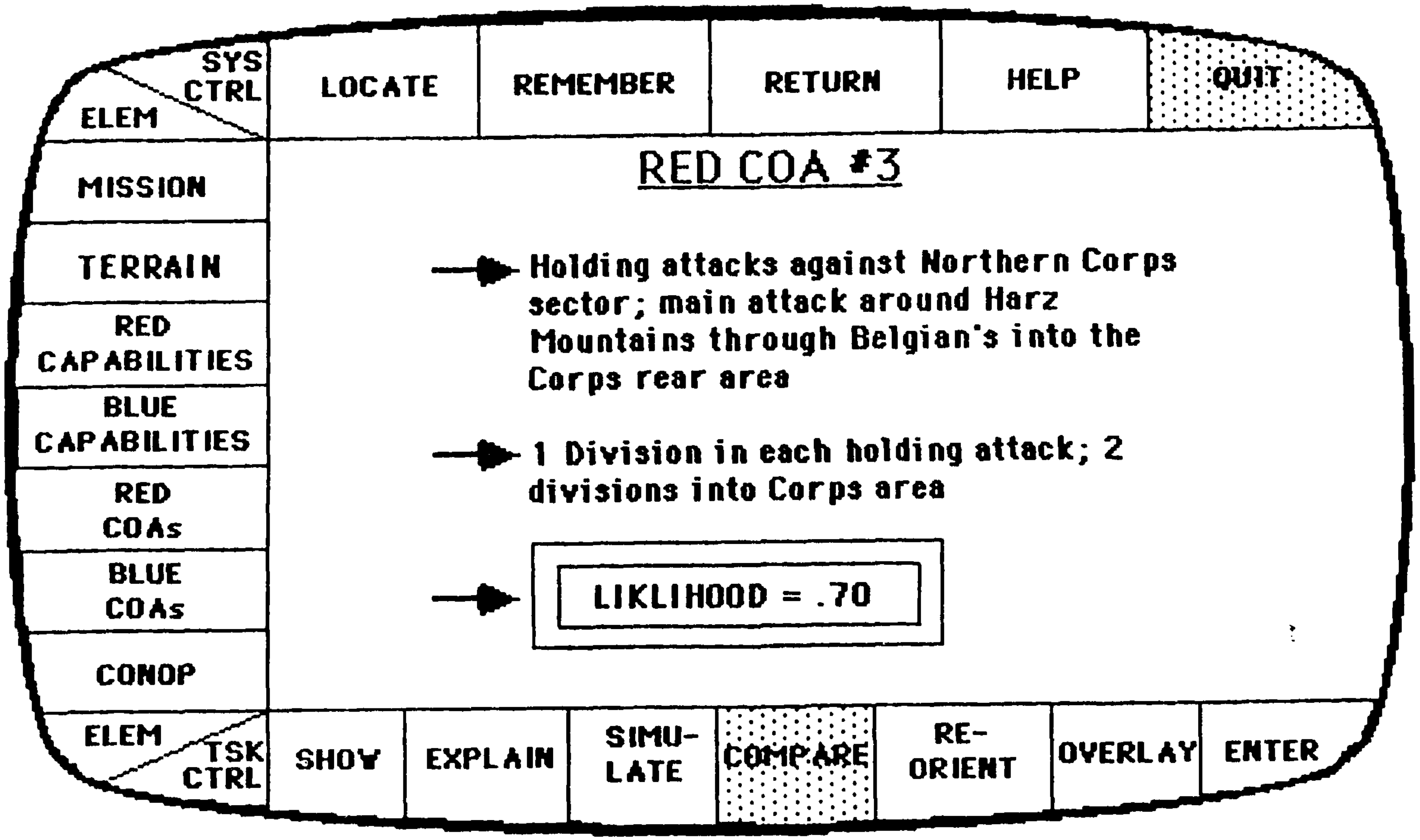
A .60 likelihood is displayed . . .



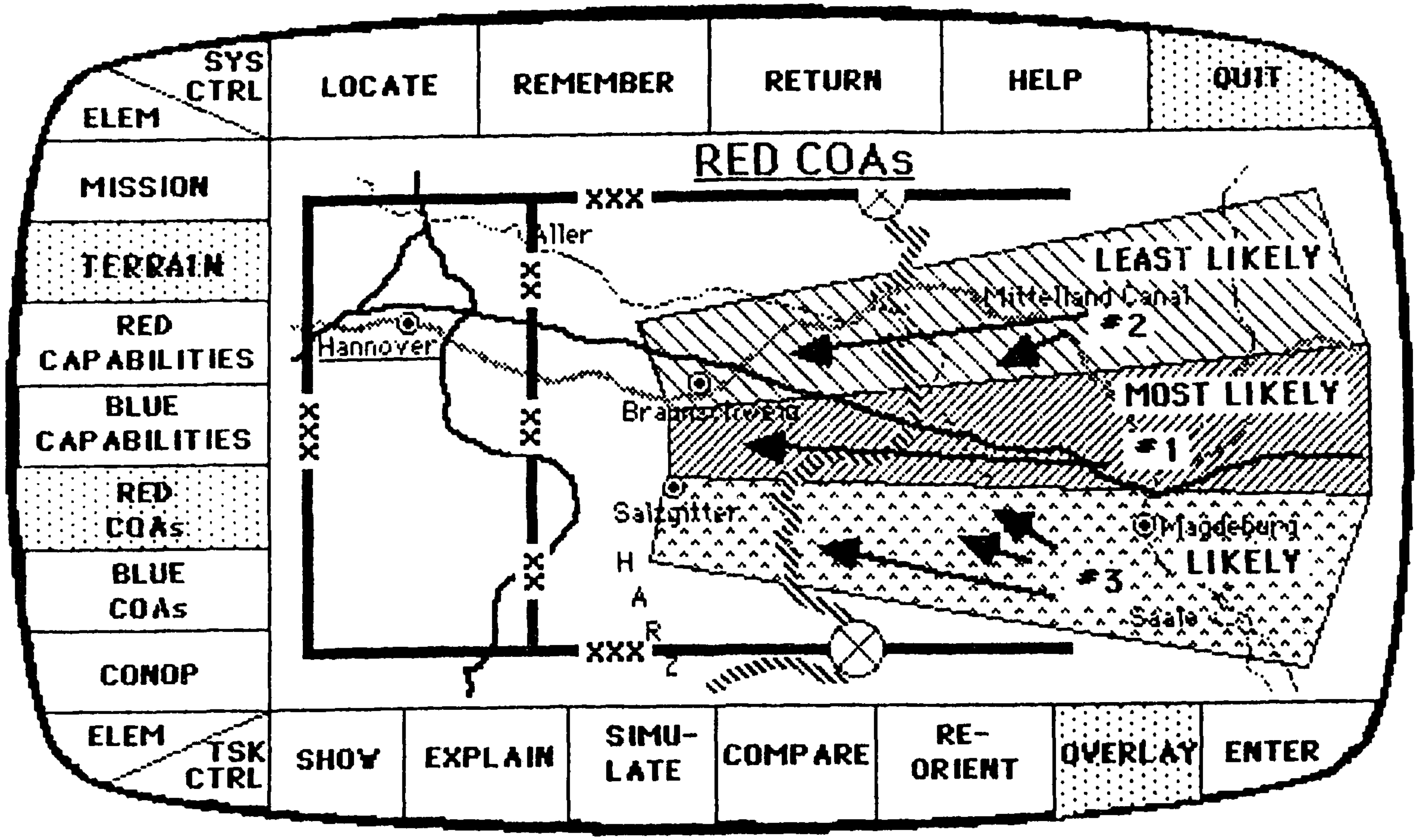
Another COA is requested . . .



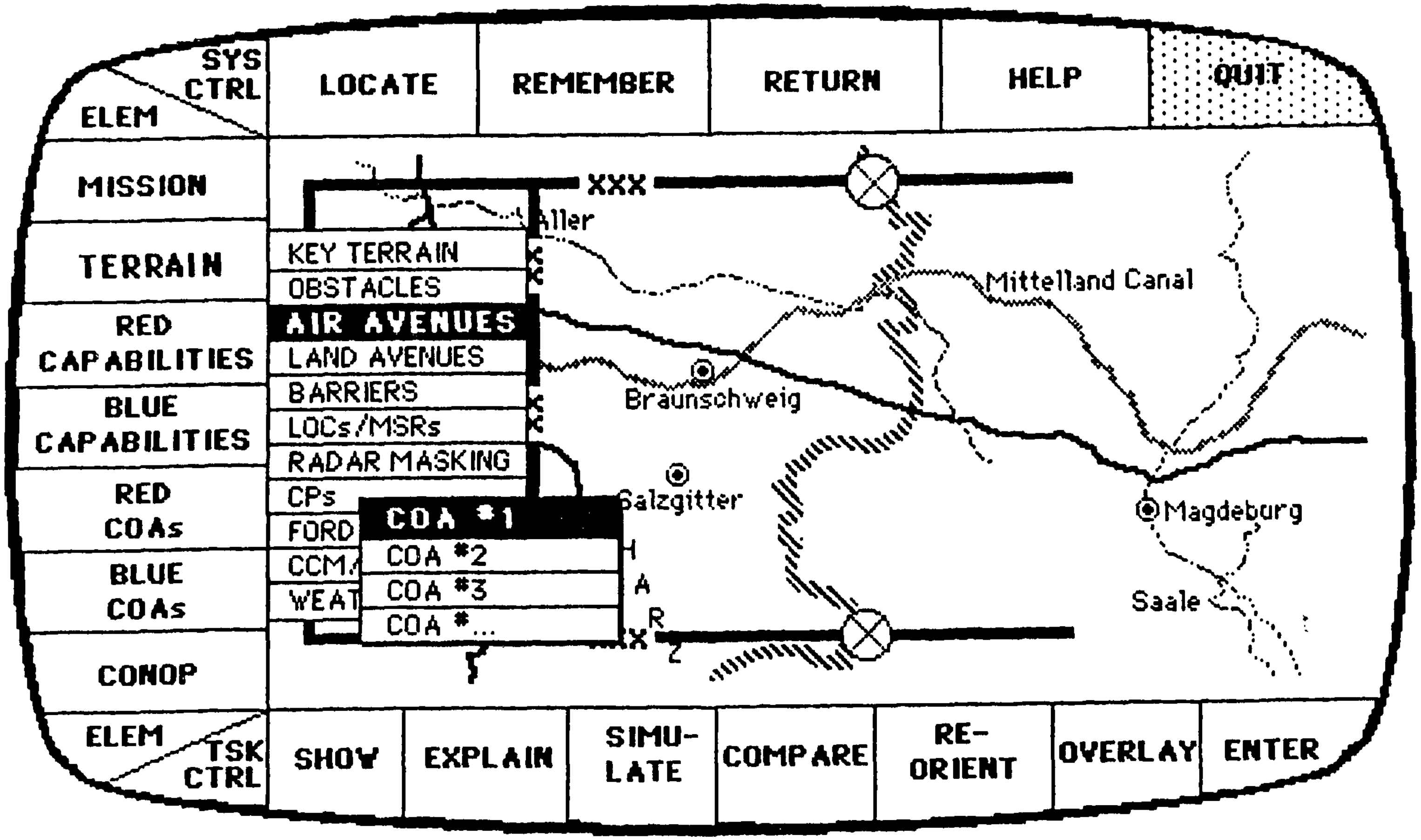
Red COA #3 is displayed . . .



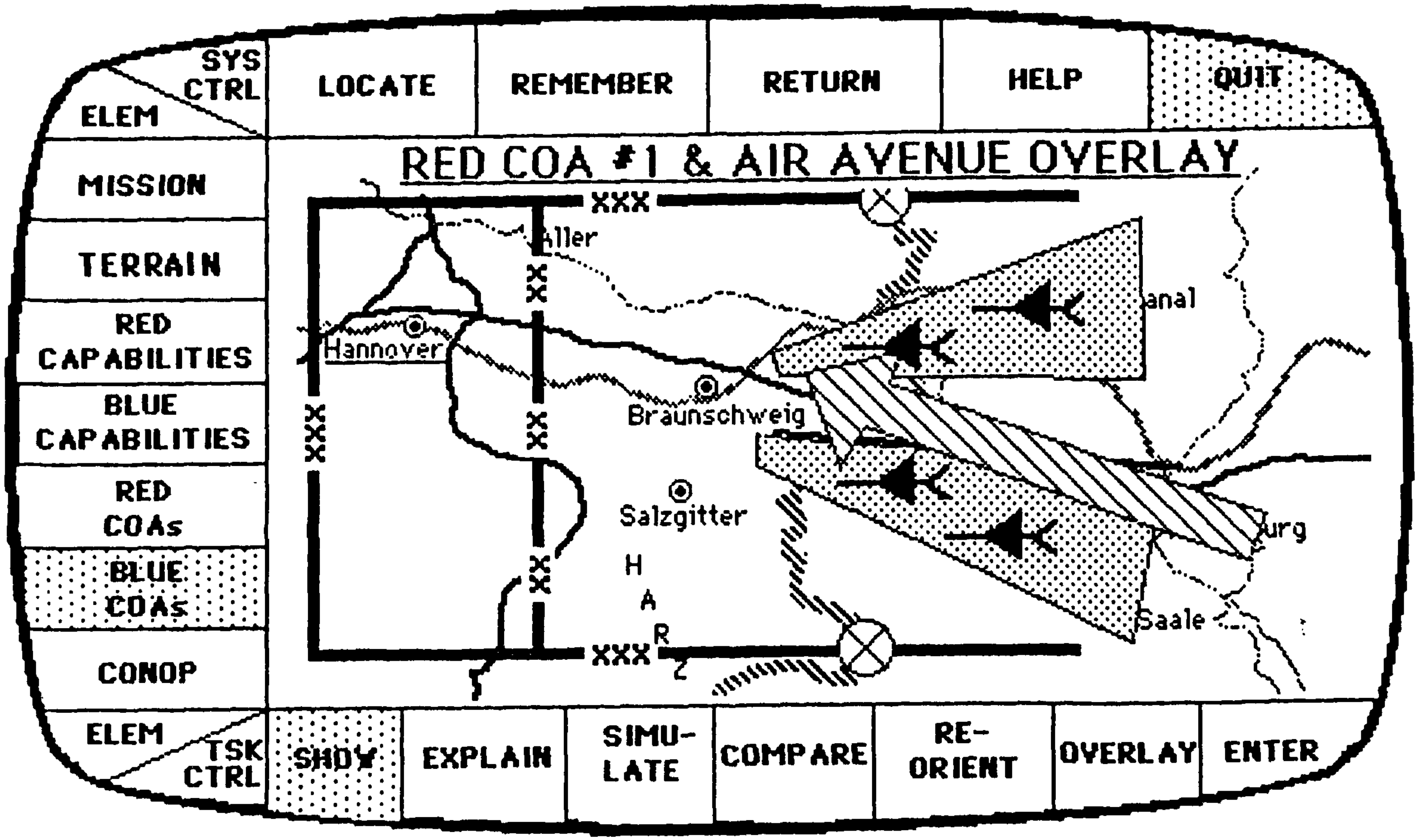
The explanation suggests that the probability is .70; it also suggests how the attack will occur . . . the planner then clicks on the **Compare** option because he wants to see all of the COAs he has seen to thusfar together . . .



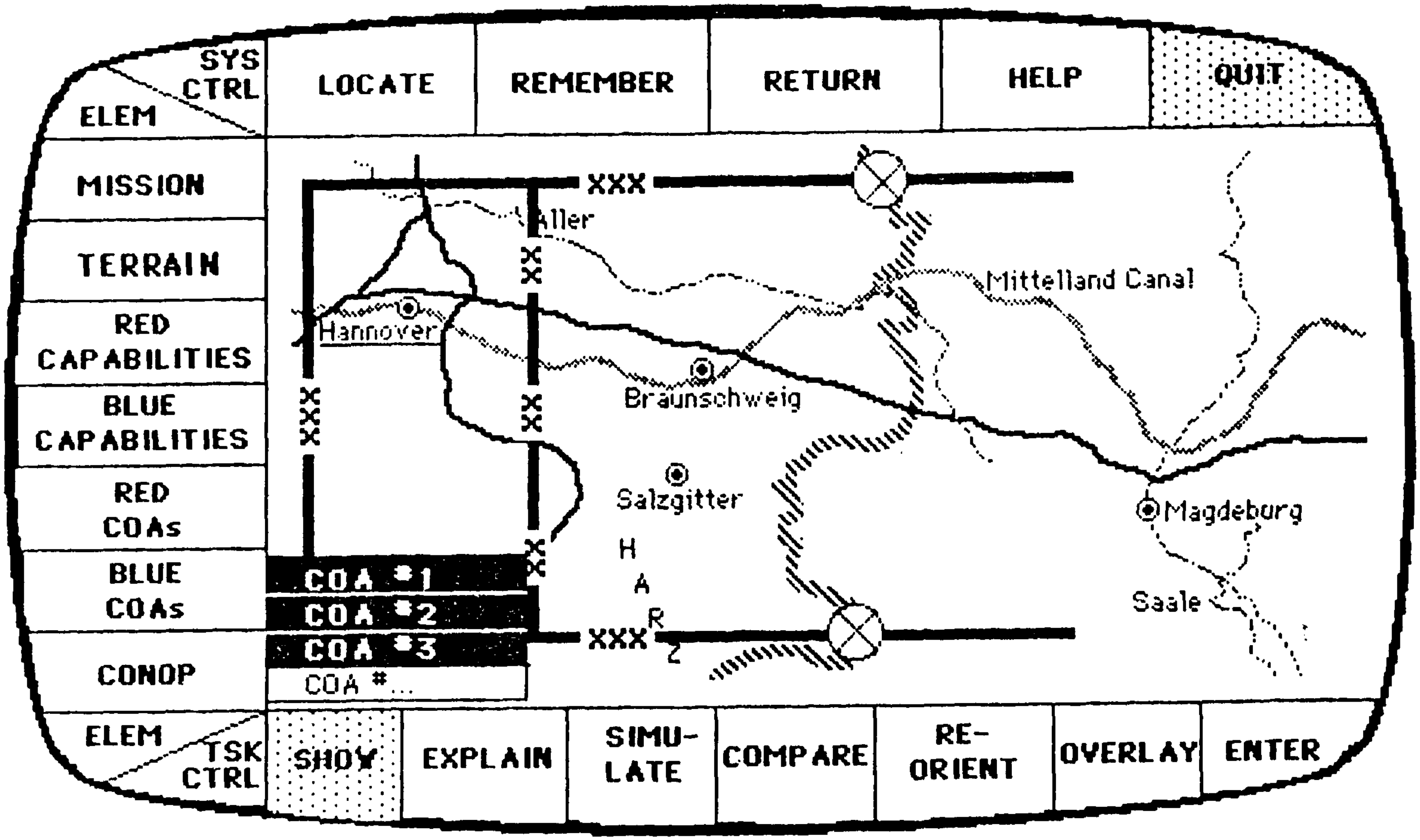
The system then displays -- again via the use of "analytical graphics" -- the three **Red COAs** and their probabilities in a single display . . . he then clicks on **Overlay** to create a hybrid display . . .



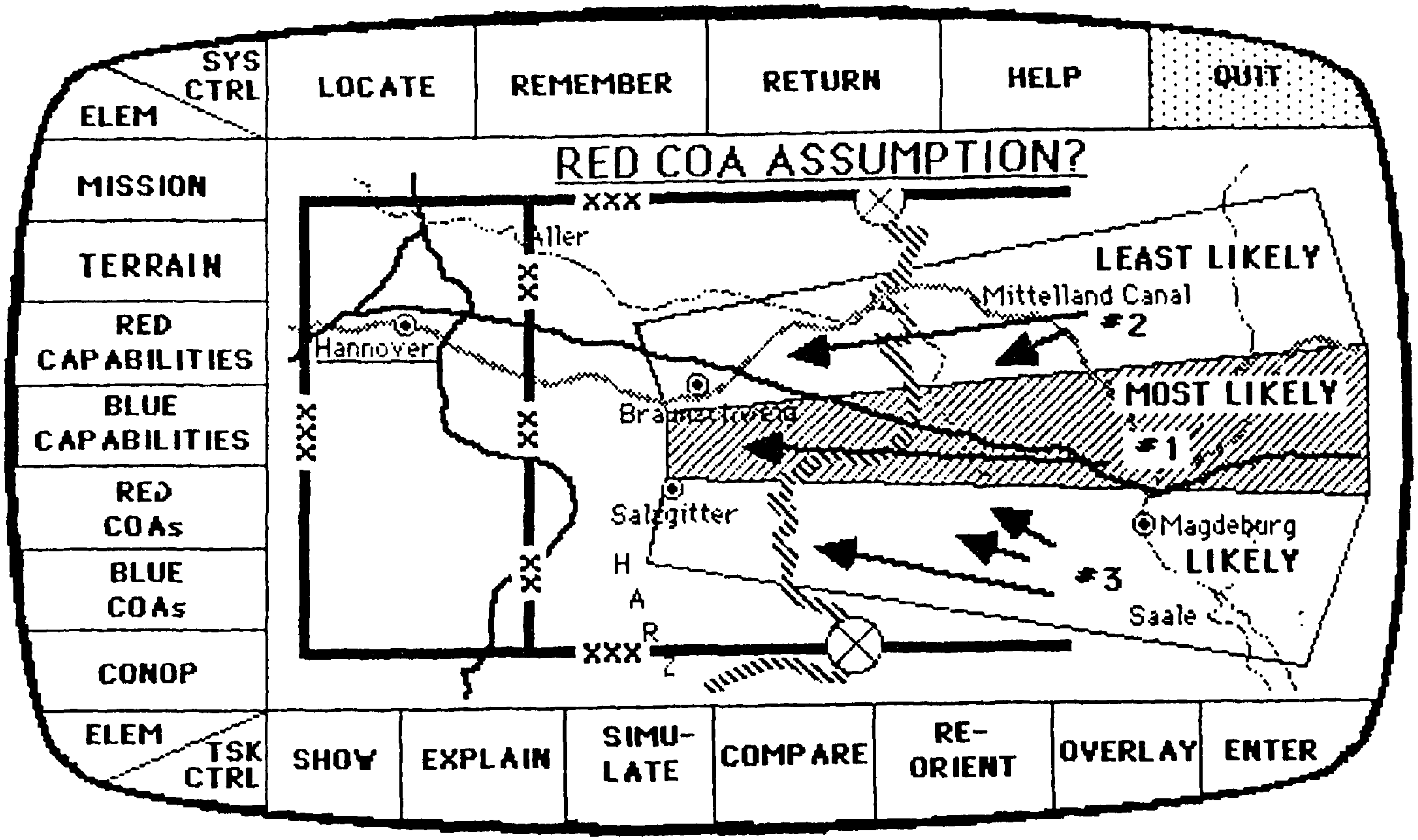
The system asks what he wants to overlay (onto what) and the planner selects among the **COAs** and **Terrain** . . .



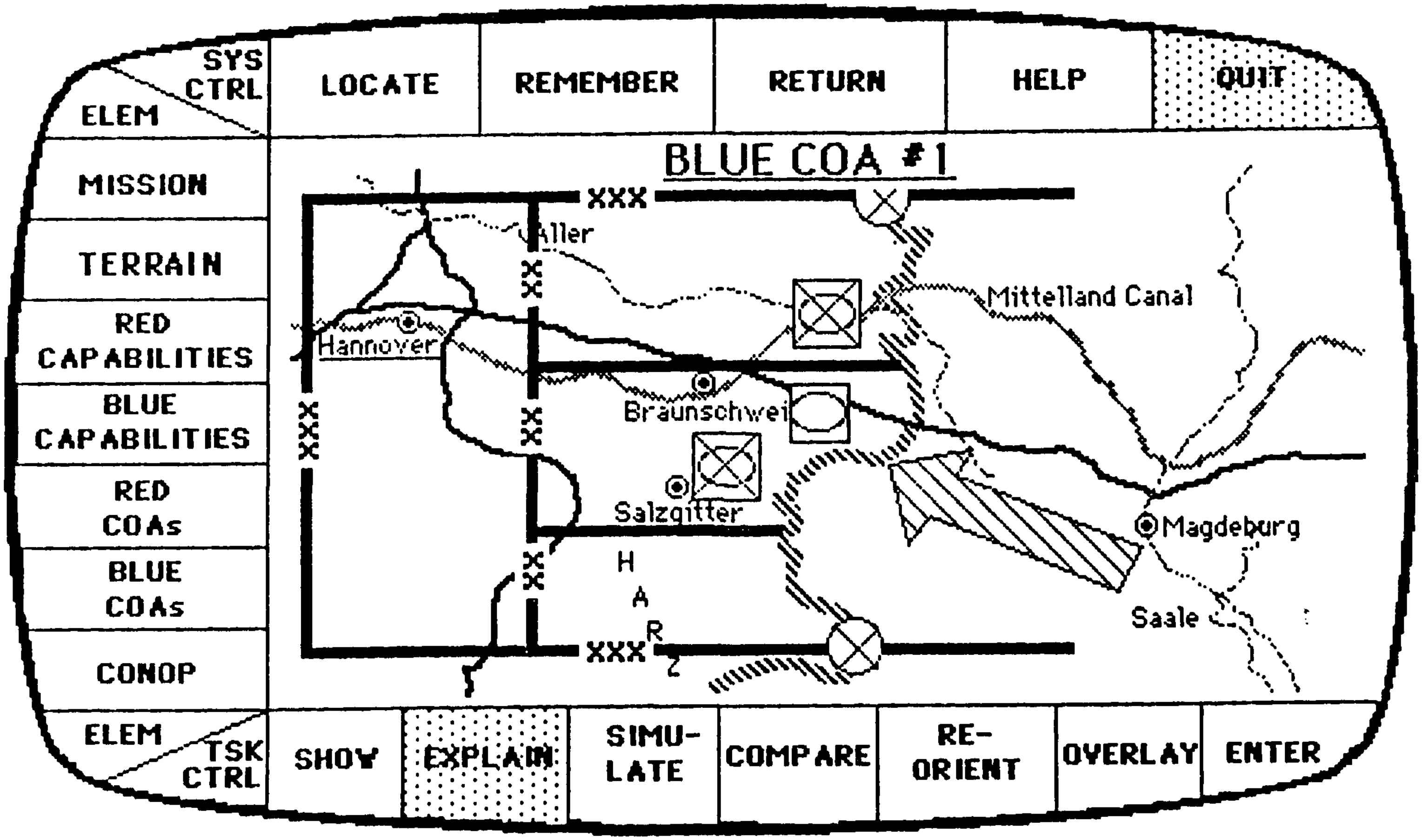
The planner selects **Air Avenues** and **Red COA #1** . . .



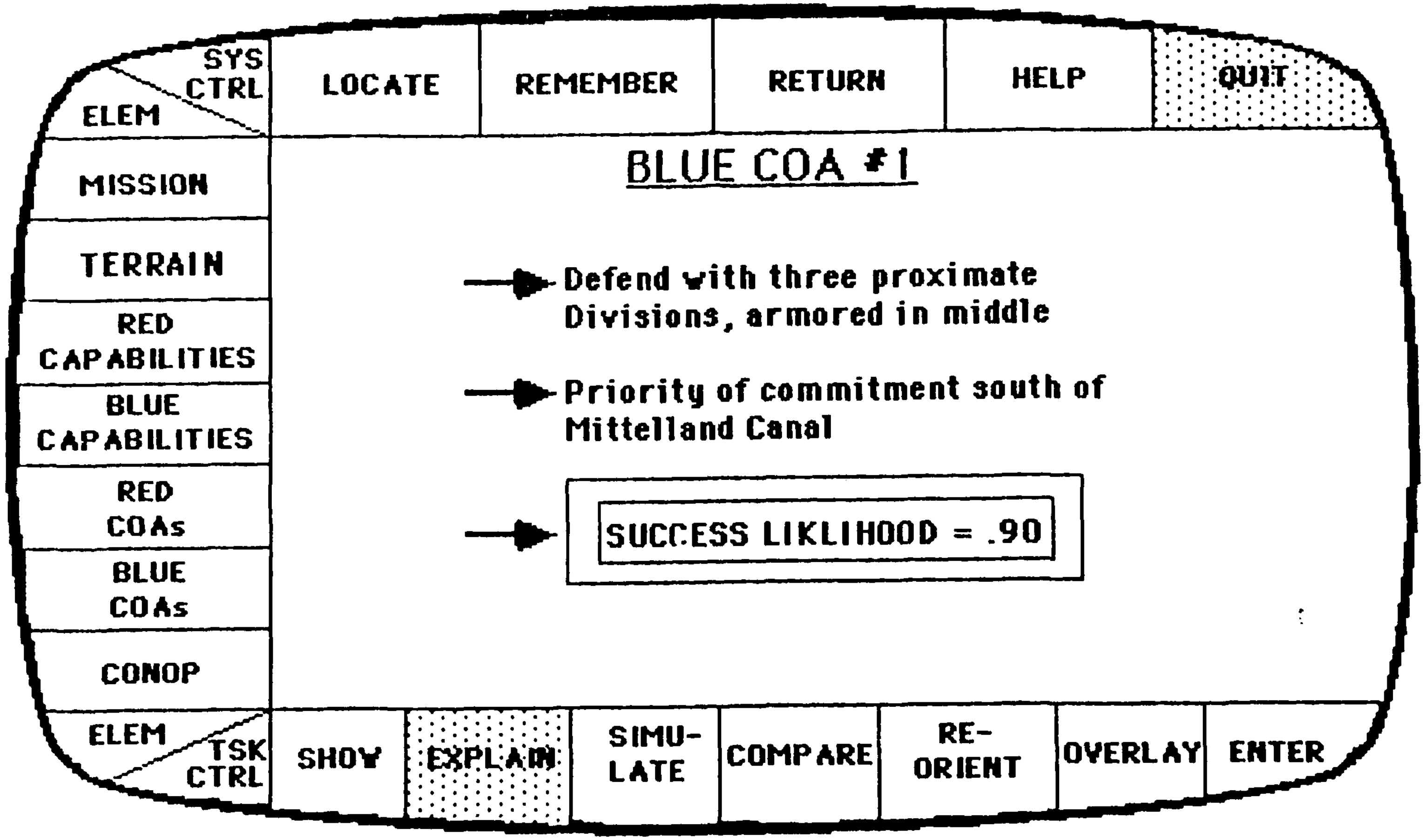
The planner selects **Blue COAs** . . .



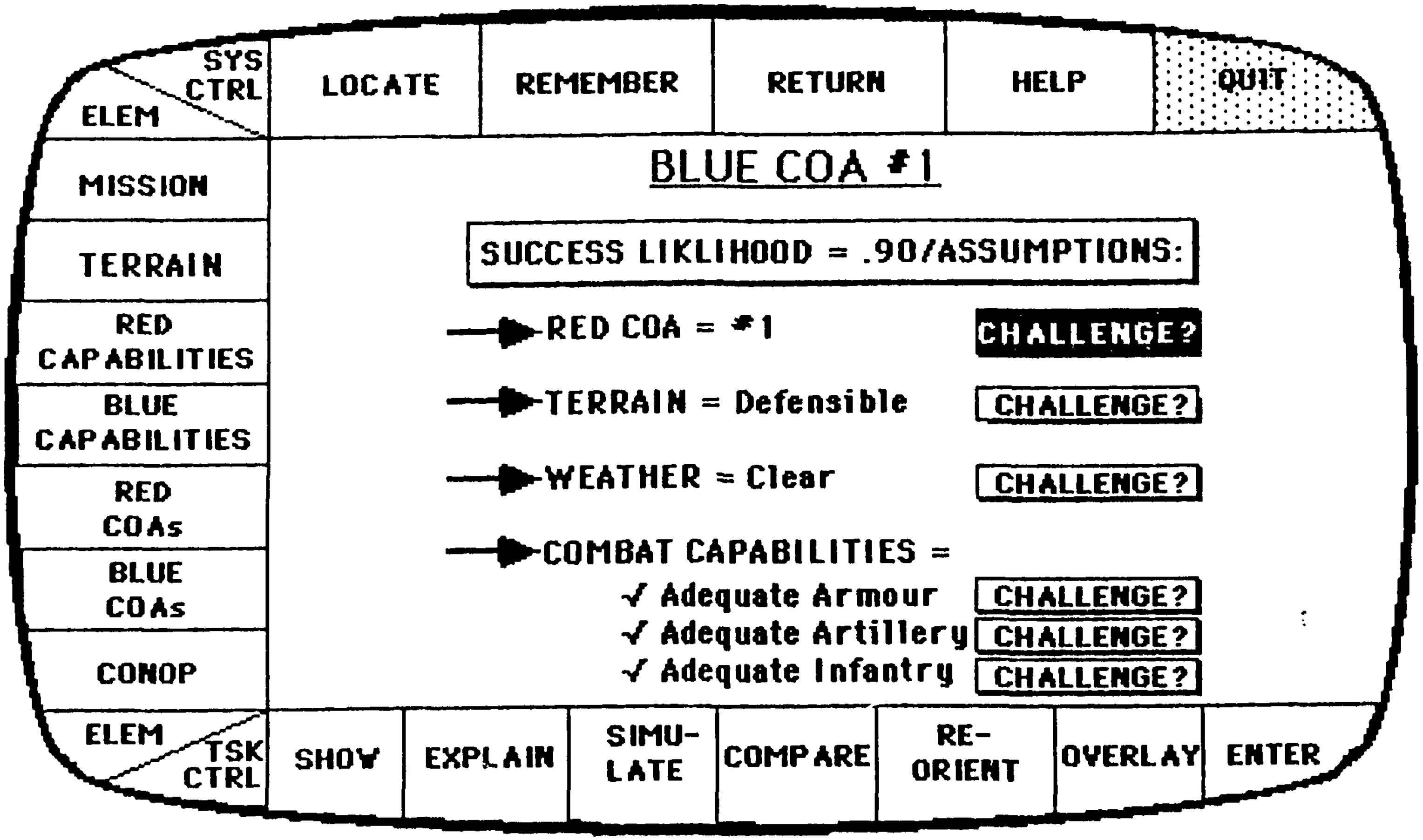
The system first requests information about what the planner thinks Red will do based on his analysis of hypothetical Red COAs . . the planner selects **Red COA #1** by clicking directly on the COA zones previously displayed to the user . . .



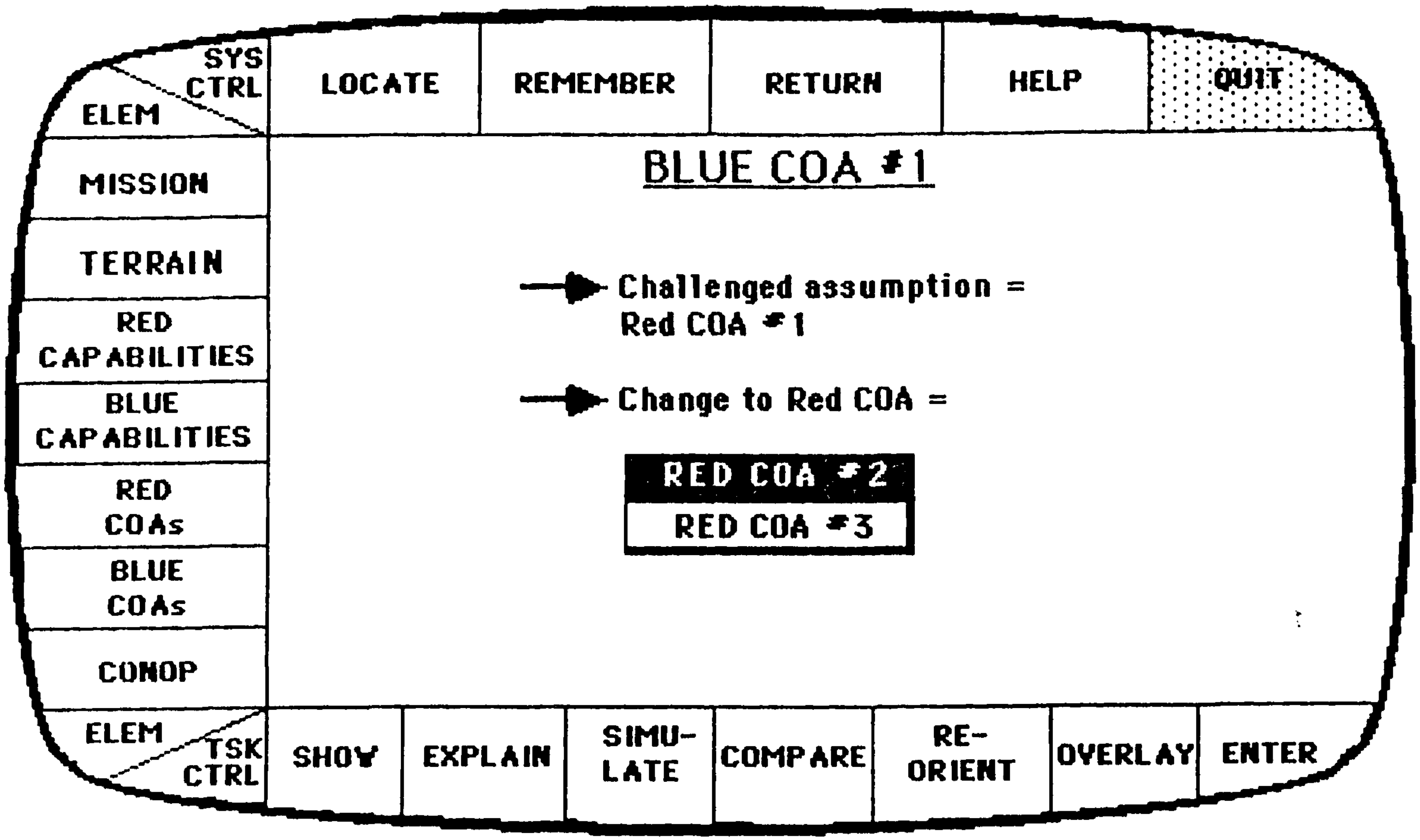
The system then generates **Blue COA #1** for the planner to inspect . . .



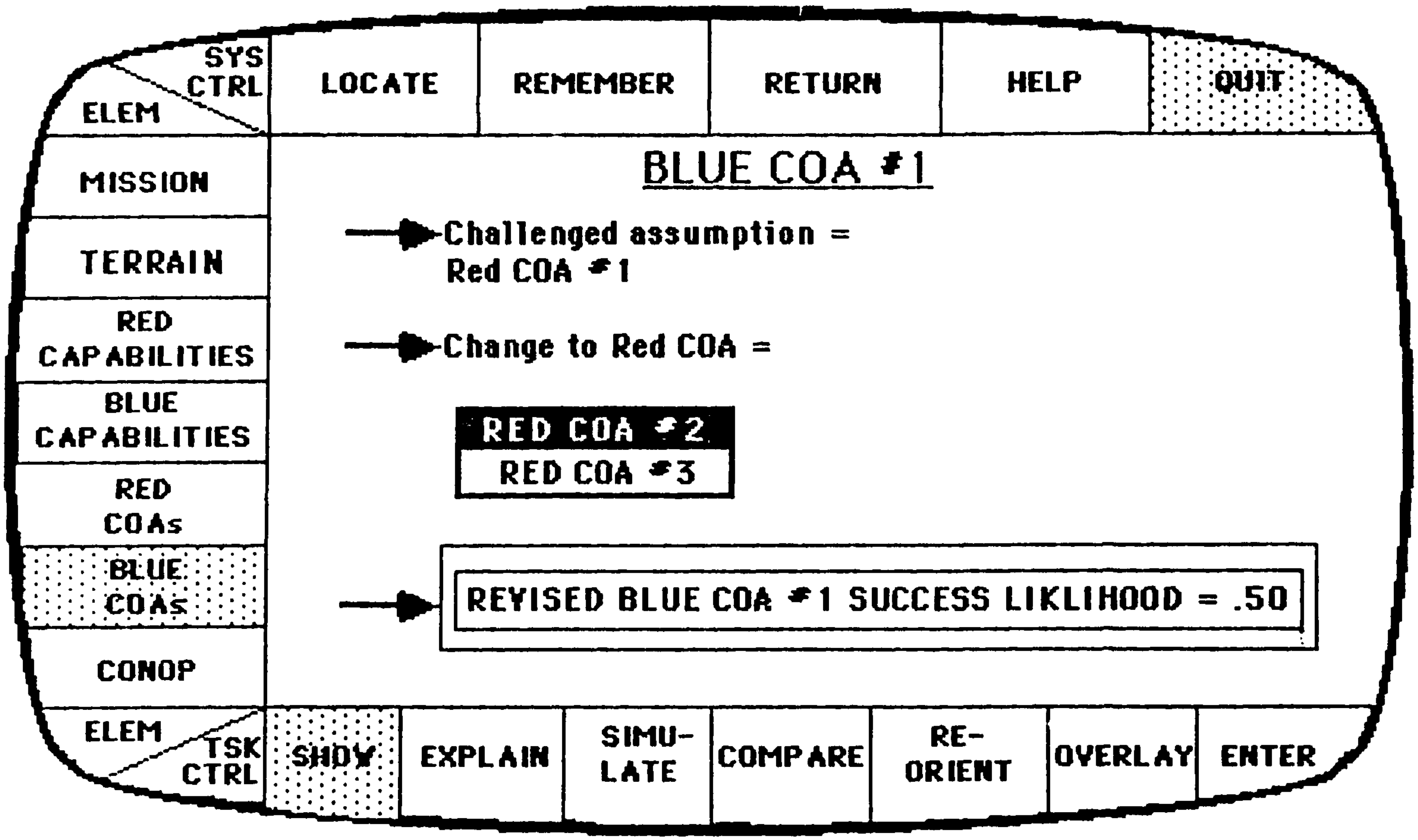
The explanation suggests why the plan might work . . . and provides a .90 probability of success -- given the likelihood of **Red COA #1** . . . the planner wants more information so clicks on **Explain** . . .



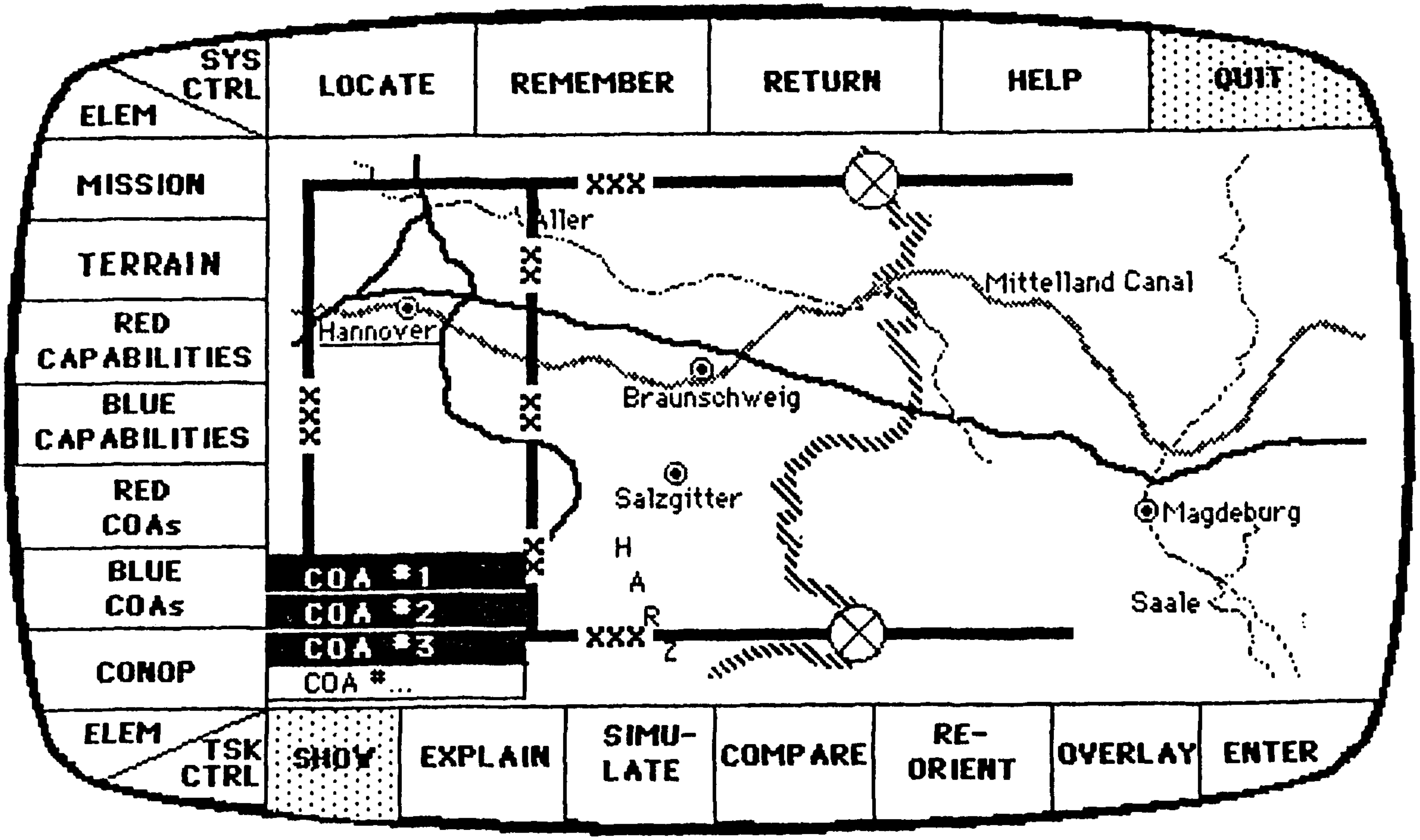
The assumptions that underlie **Blue COA #1** are displayed to the user and he challenges the assumption about the likelihood of **Red COA #1** . . .



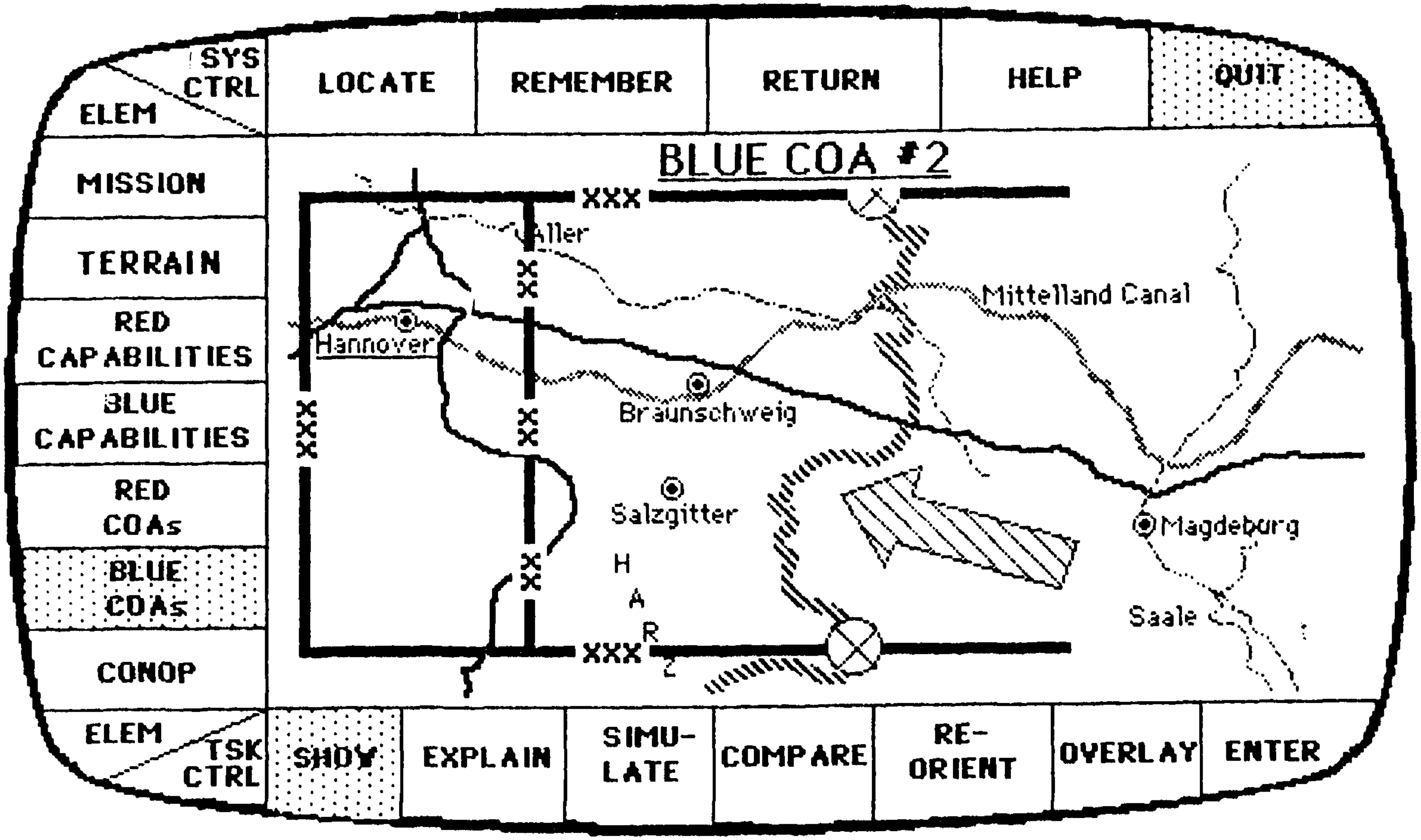
The change occurs and the system calculates the impact . . .



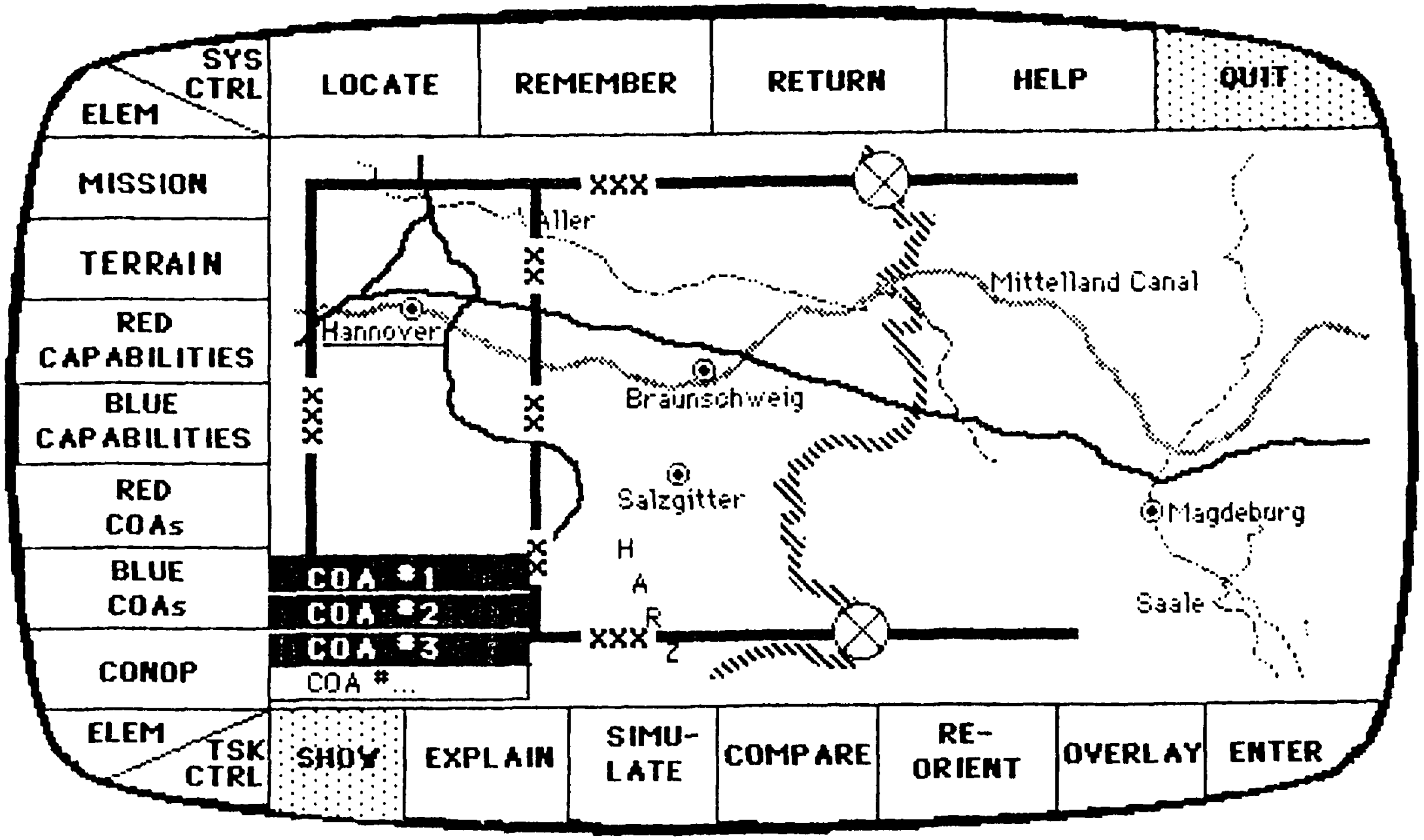
The system decides that the success probability of **Blue COA #1** is now only .50 . . .



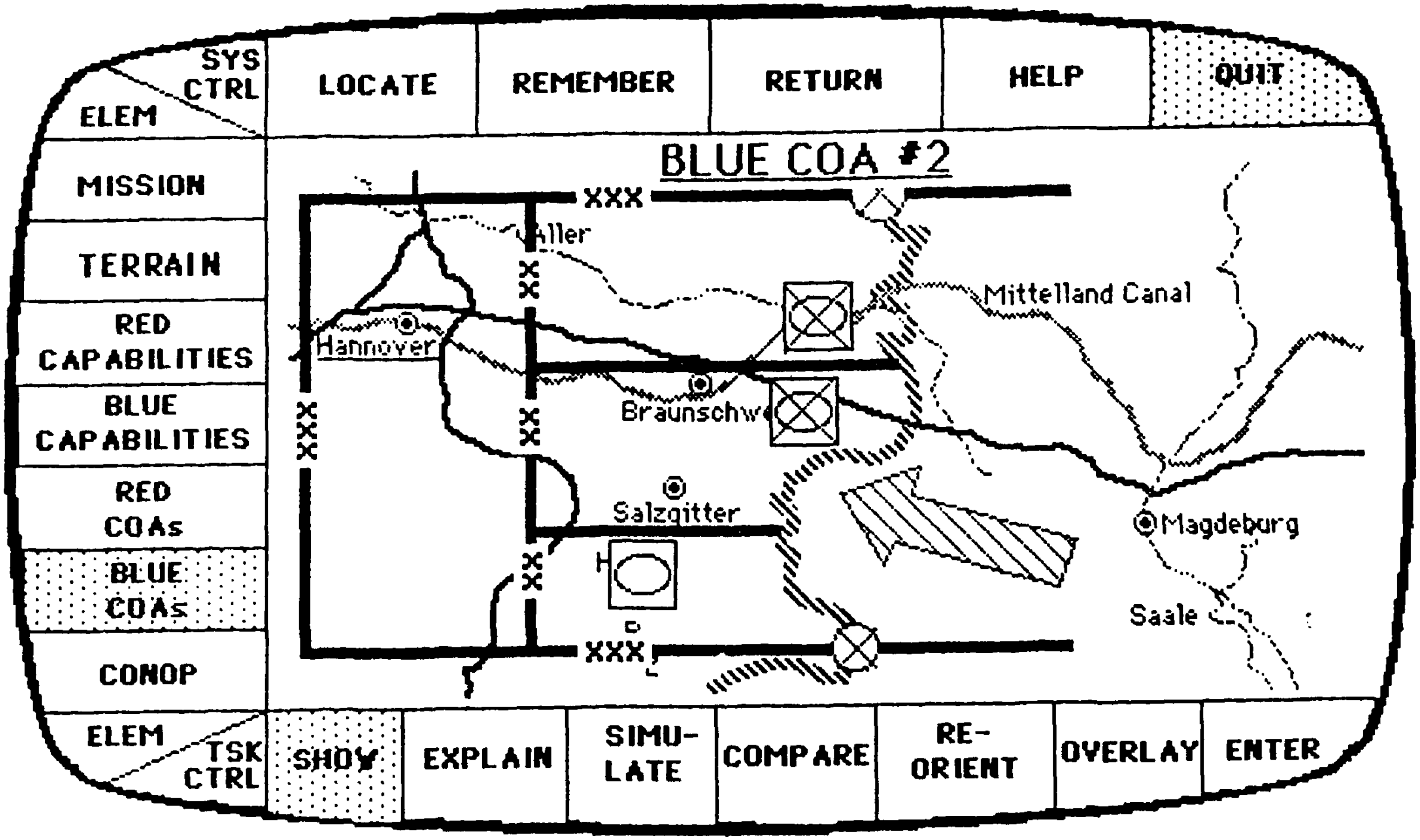
Additional COAs are requested . . .



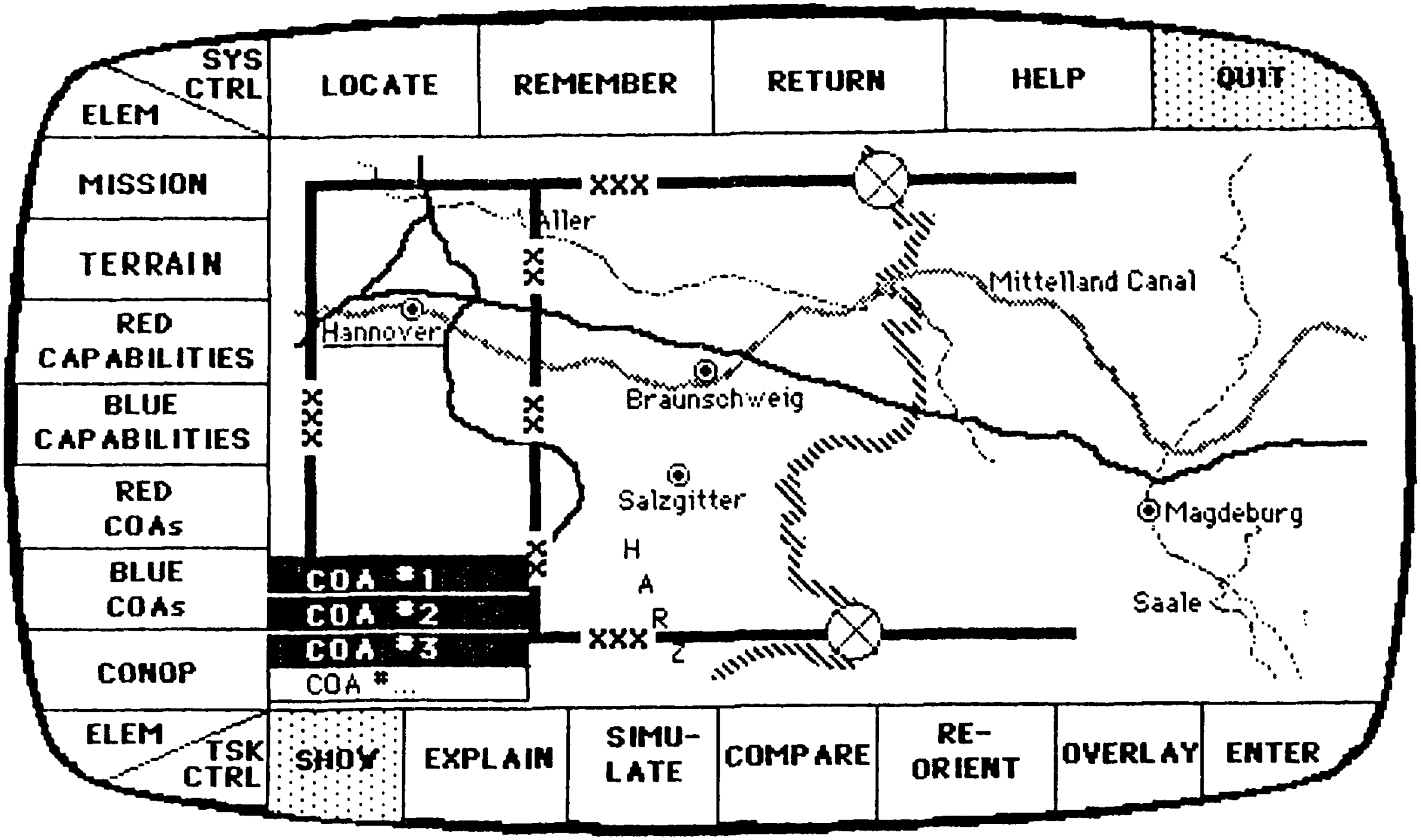
Blue COA #2 is displayed . . .



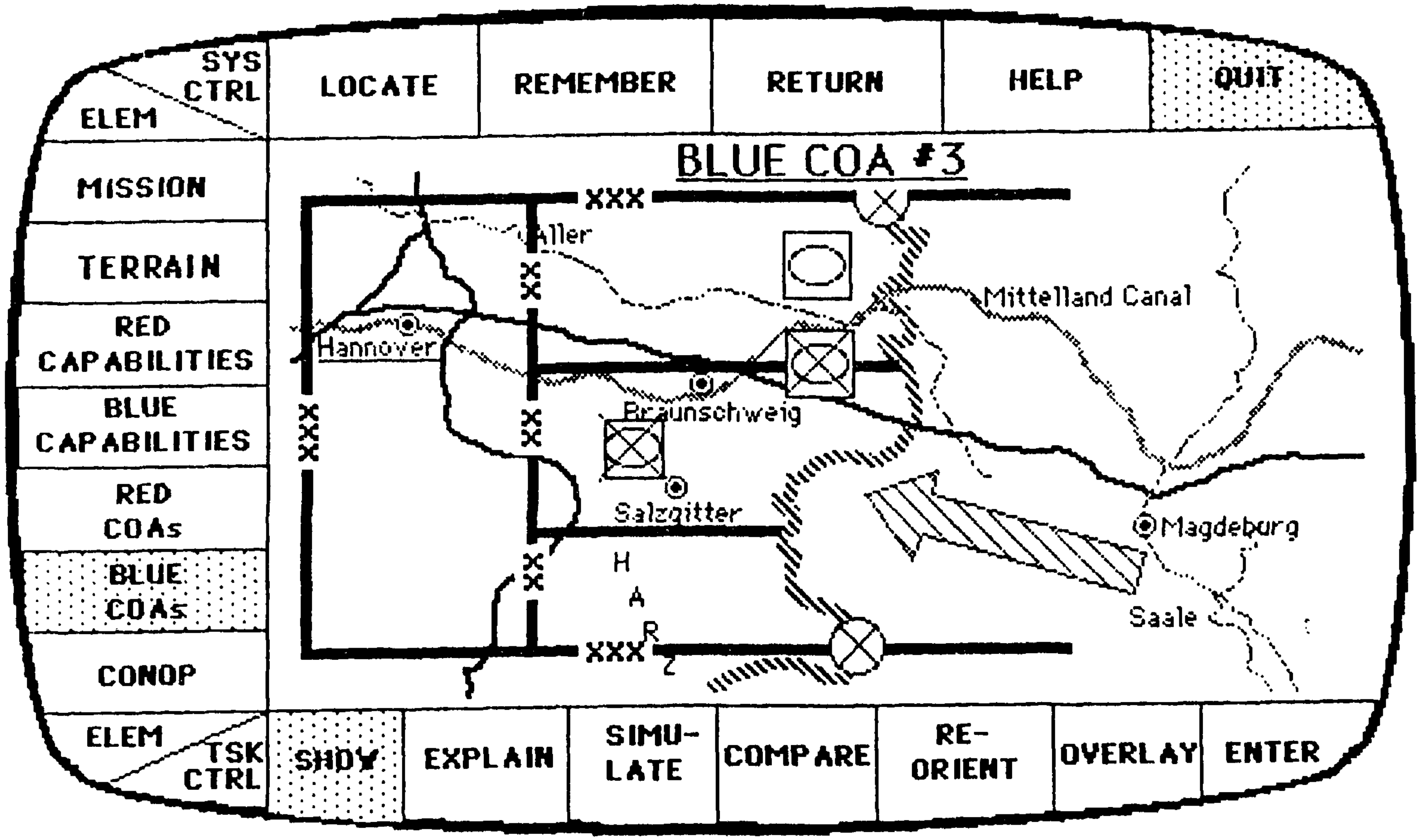
Additional information is requested . . .



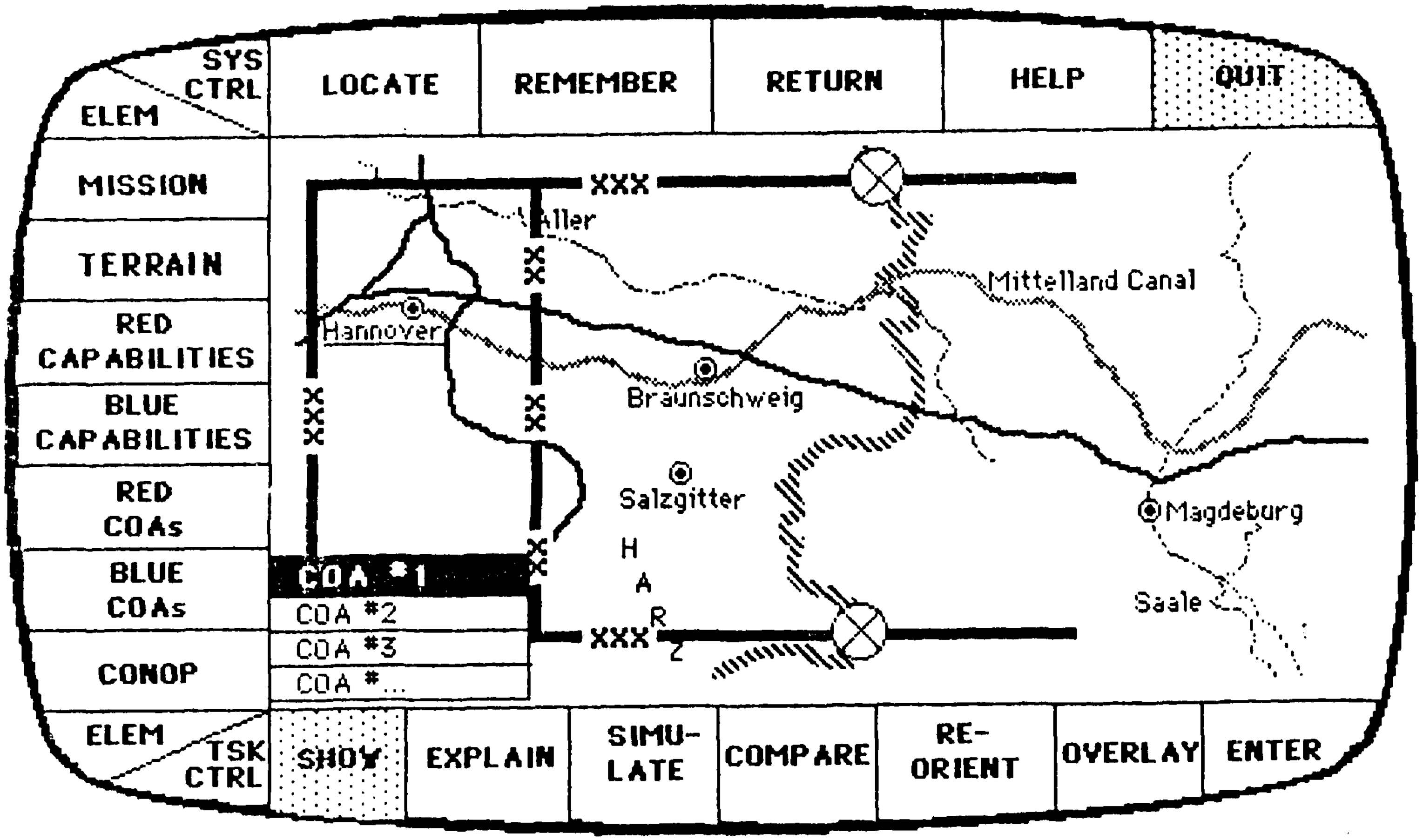
Blue **COA #2** is displayed . . .



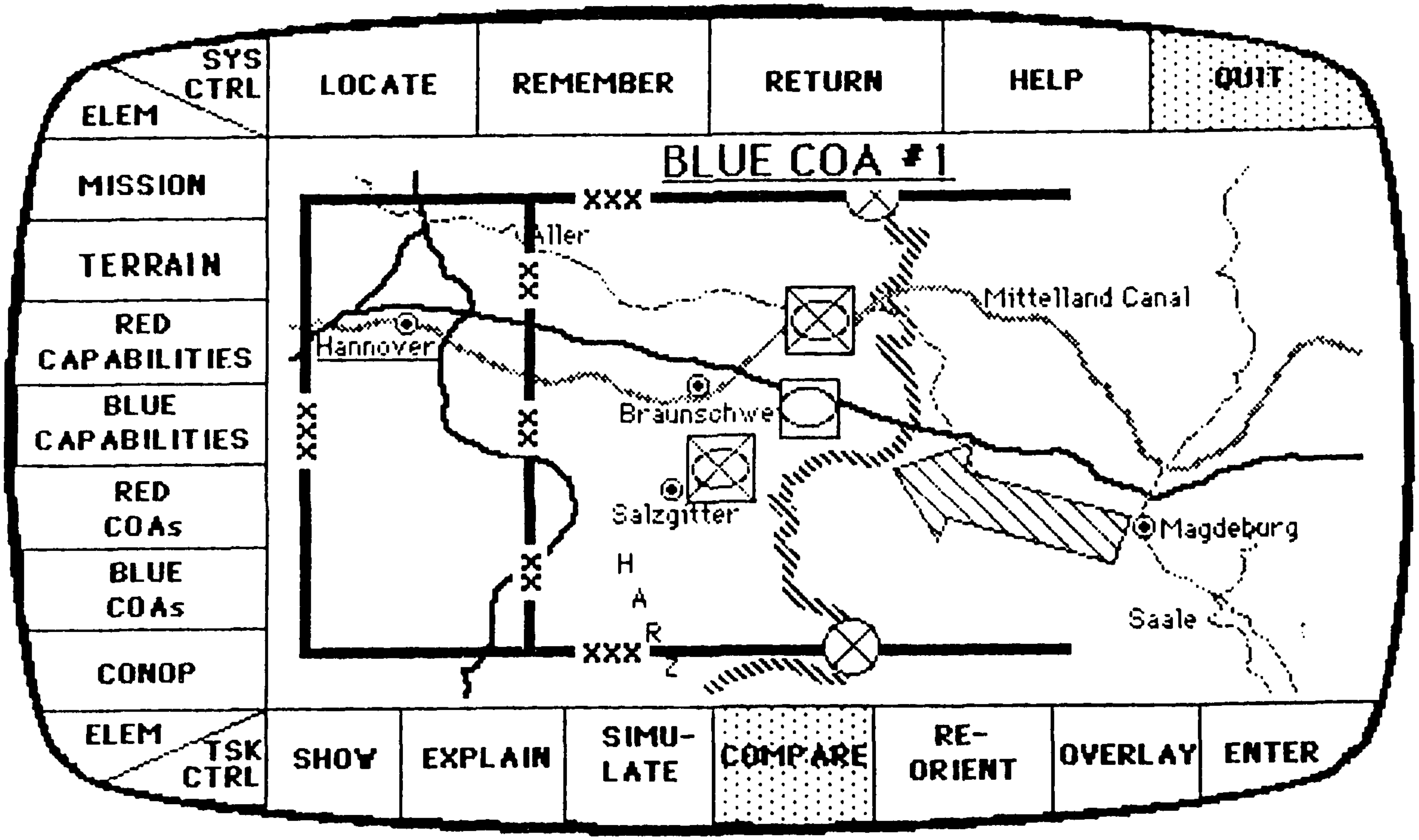
Additional Blue COAs are requested . . .



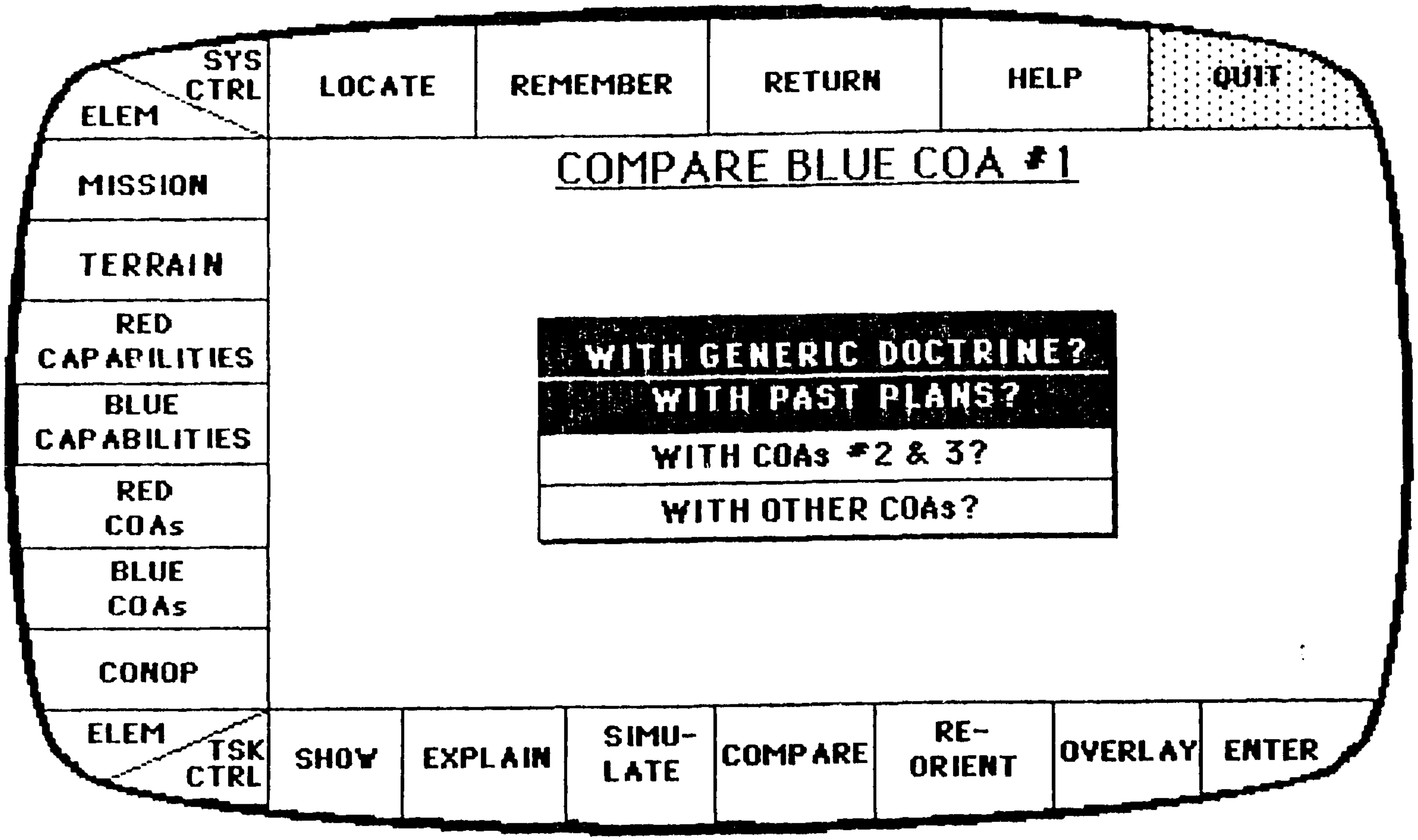
Blue **COA** #3 is displayed . . .



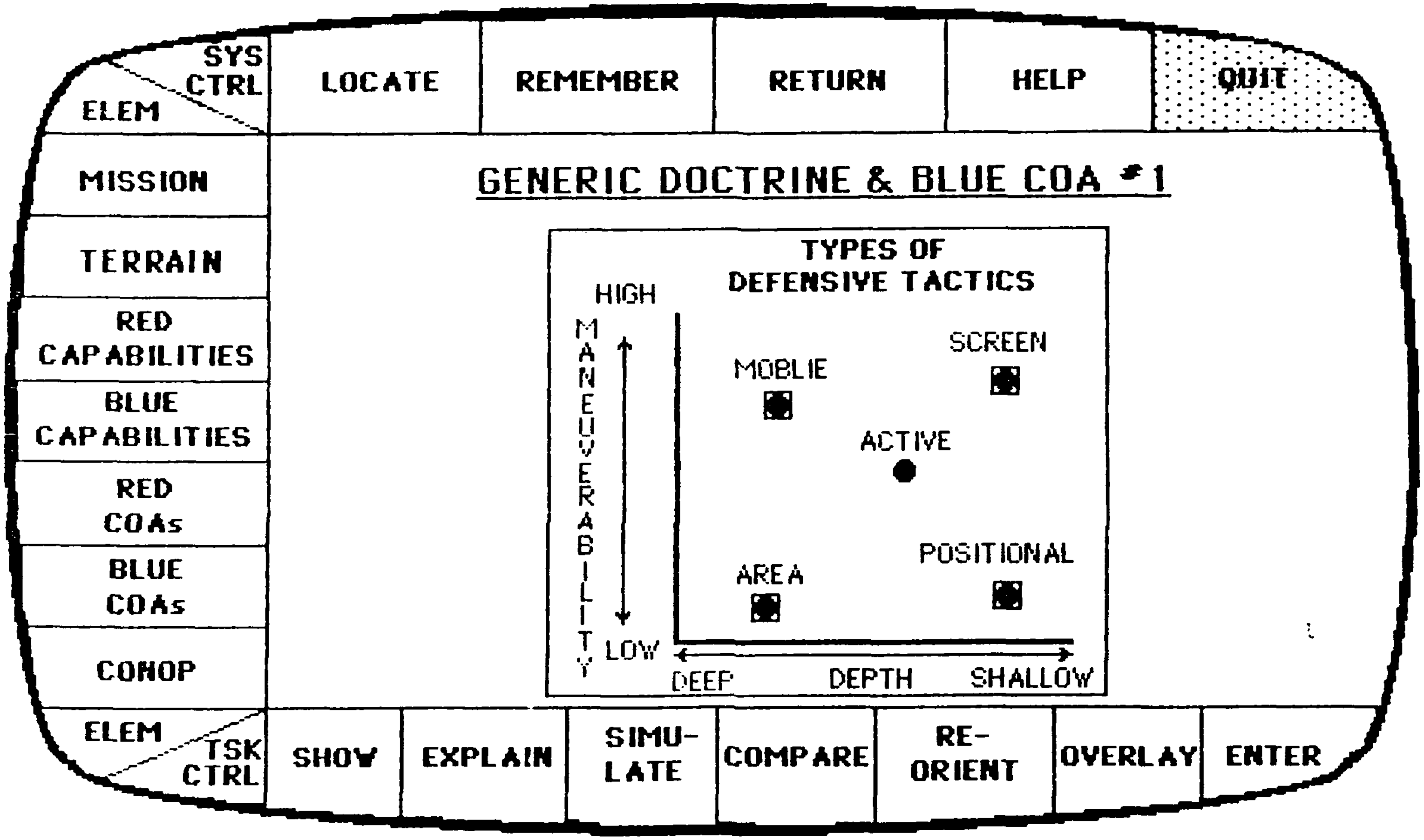
The planner returns to **Blue COAs** and requests to look once again at a specific Blue COA . . .



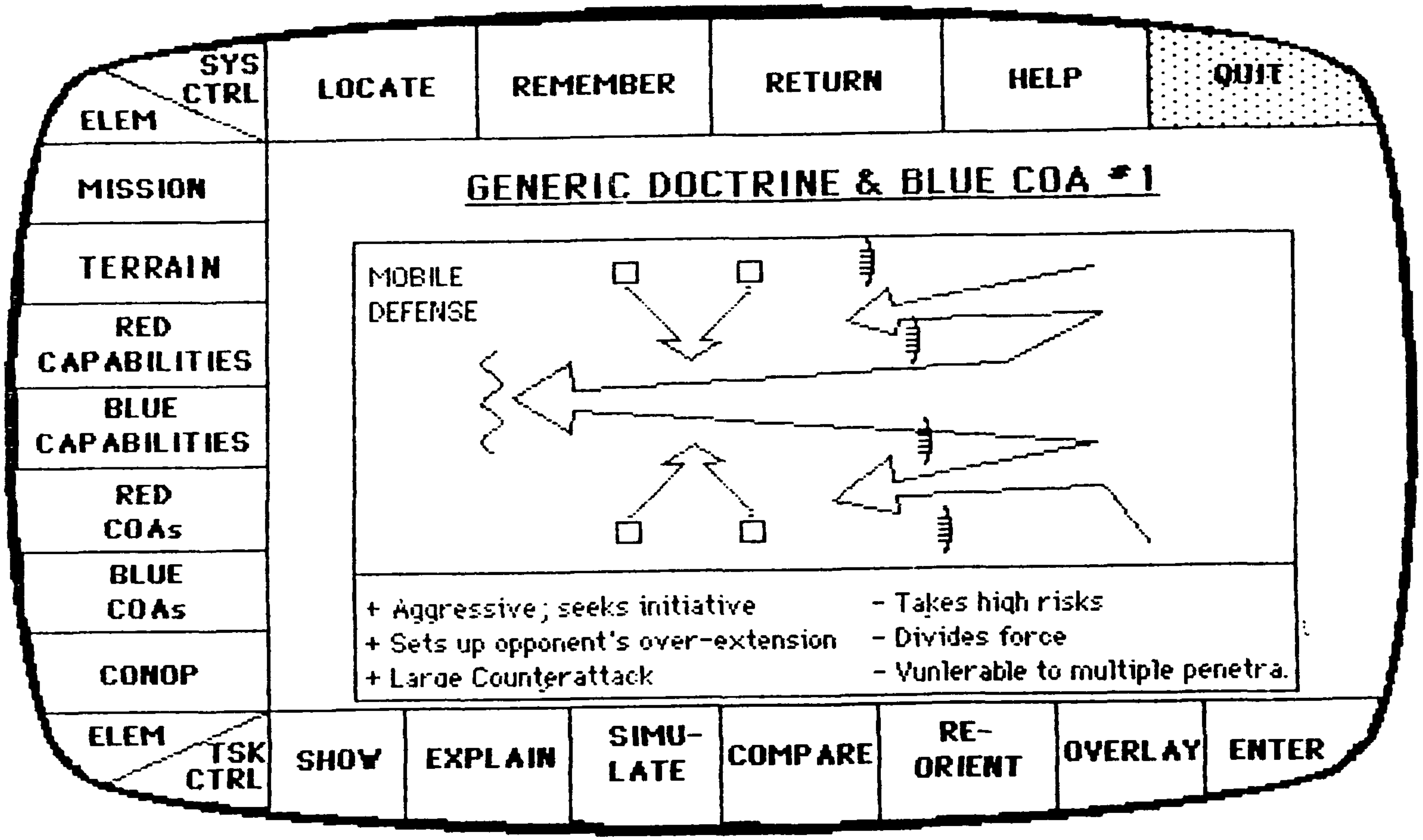
Blue **COA #1** re-appears . . . the planner clicks on **Explain** and **Compare** . .



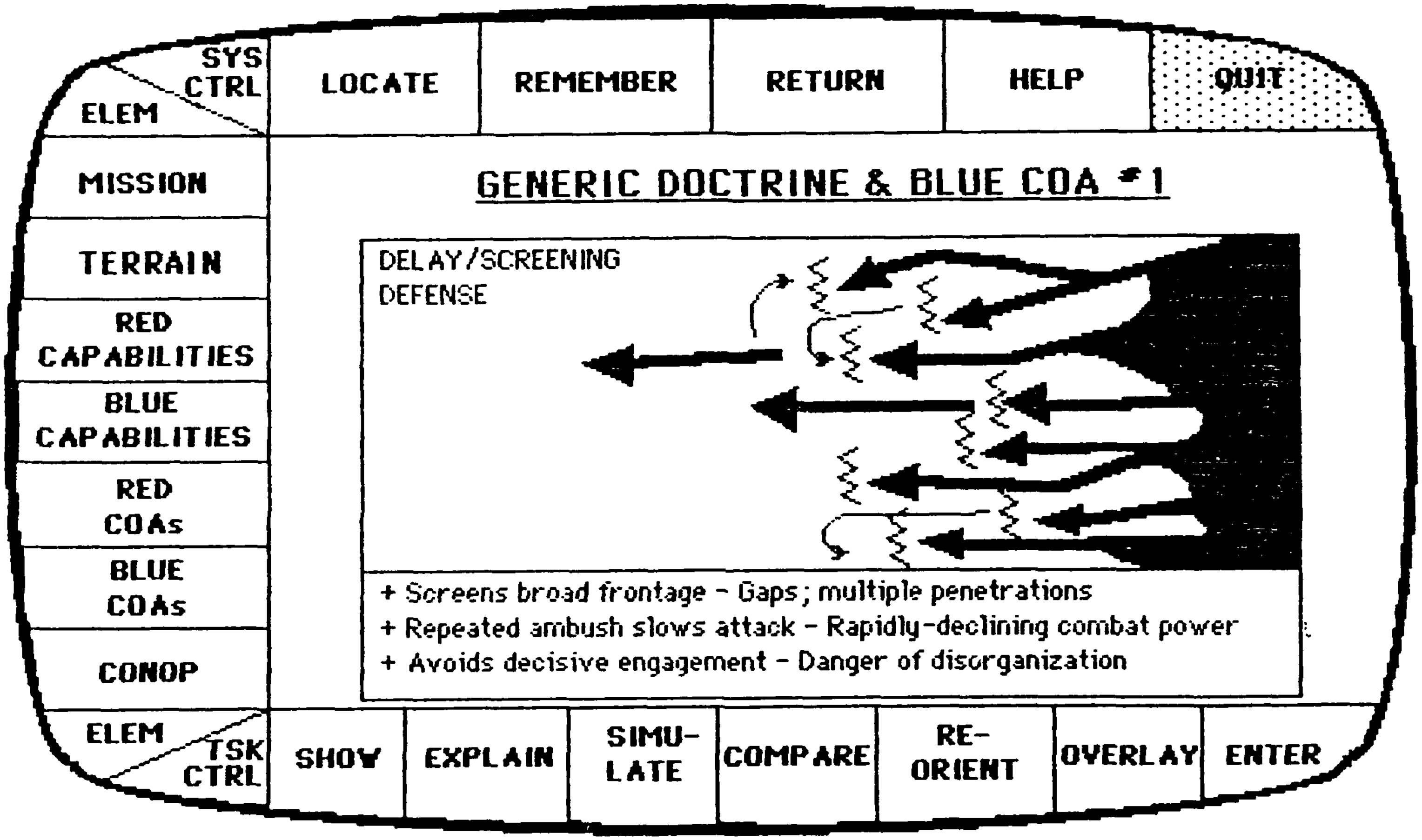
The system presents the user with a set cf comparative options; the planner requests a comparison of **Blue COA #1** with **Generic Doctrine** in order to test his assumptions about how consistent the system's recommendations are vis-a-vis what he has learned at the Army War College, the Command General & Staff College and via his experience . . . this is an important capability of the system since it permits generic analogical reasoning . . .



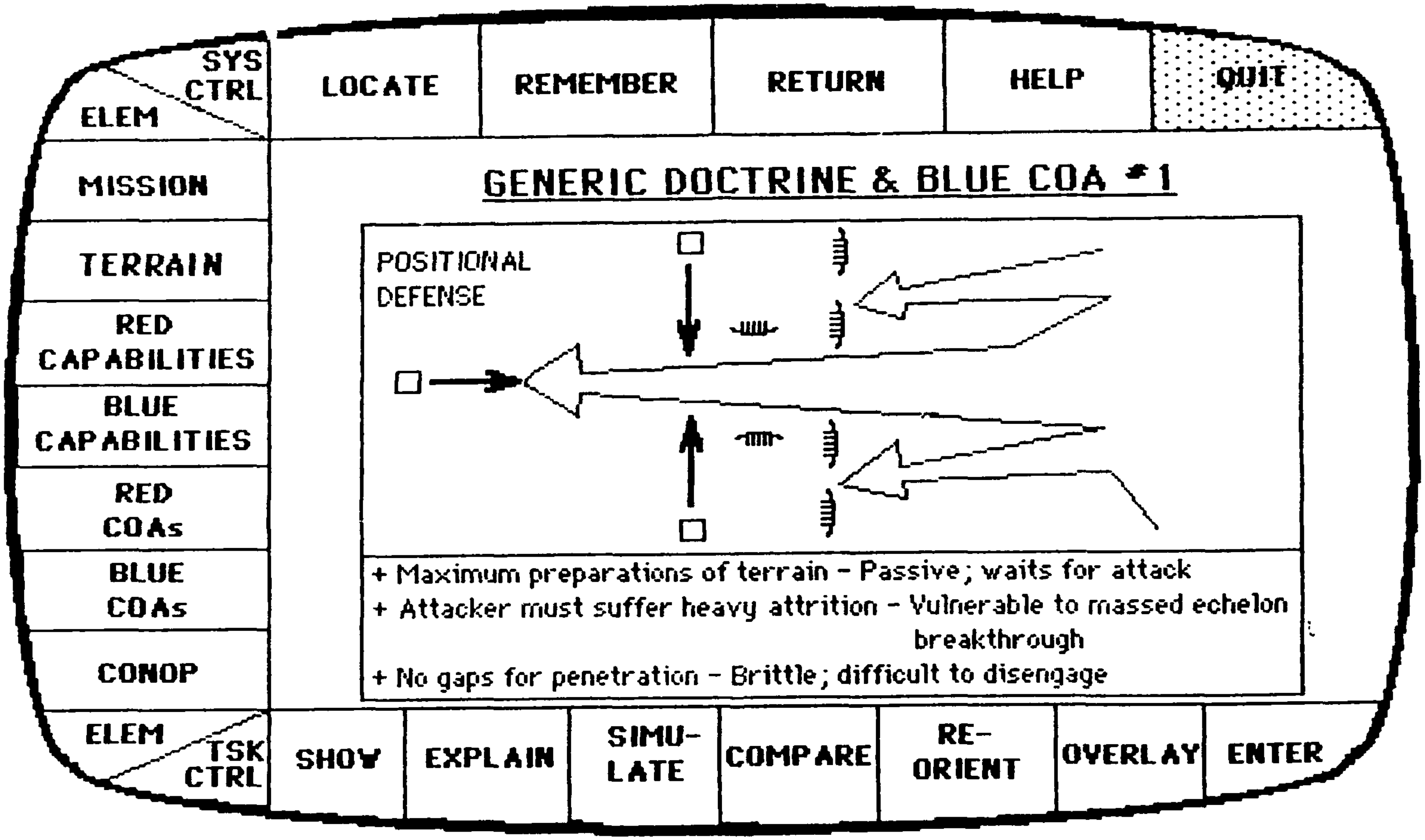
Generic doctrinal display . . .



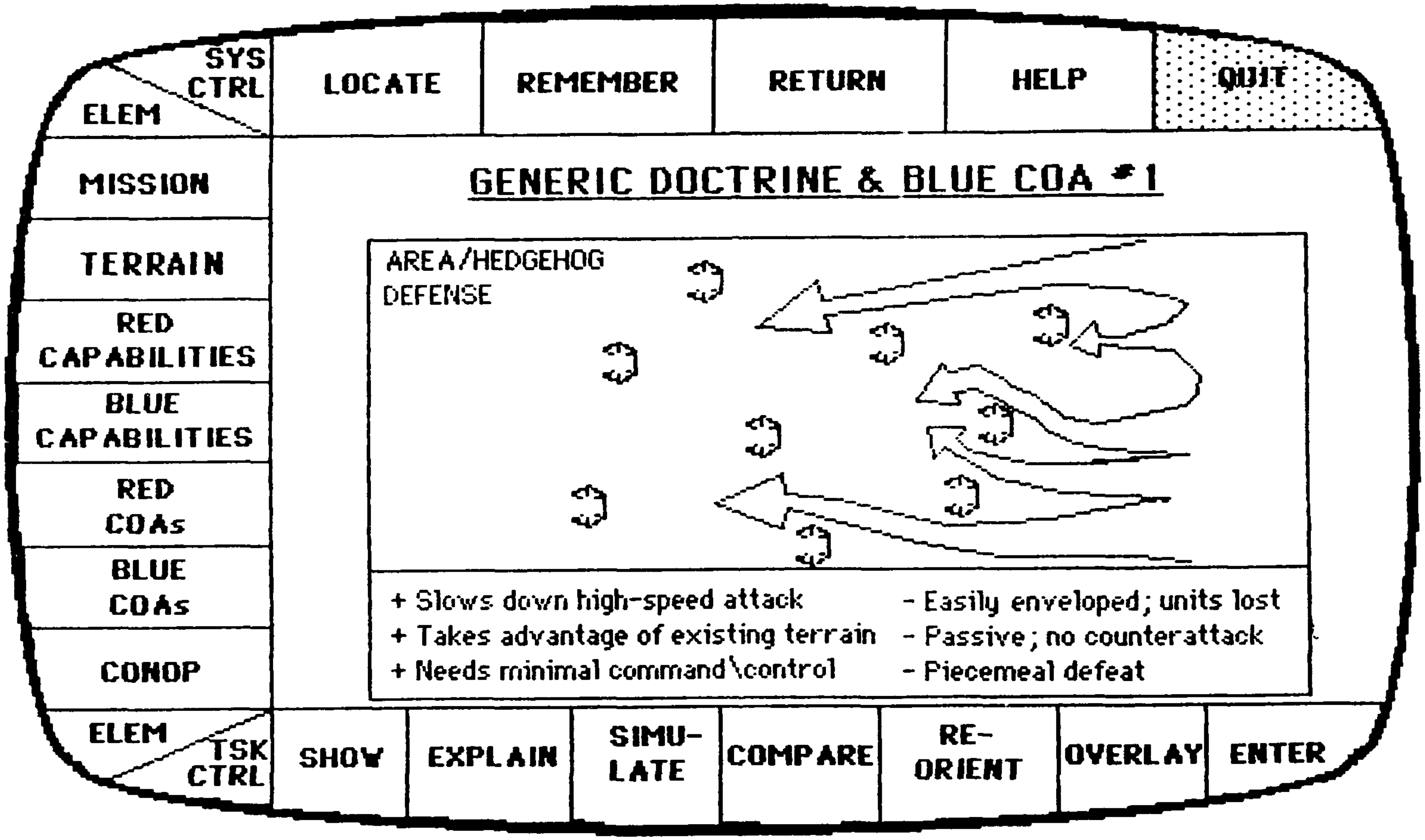
Generic doctrinal display . . .



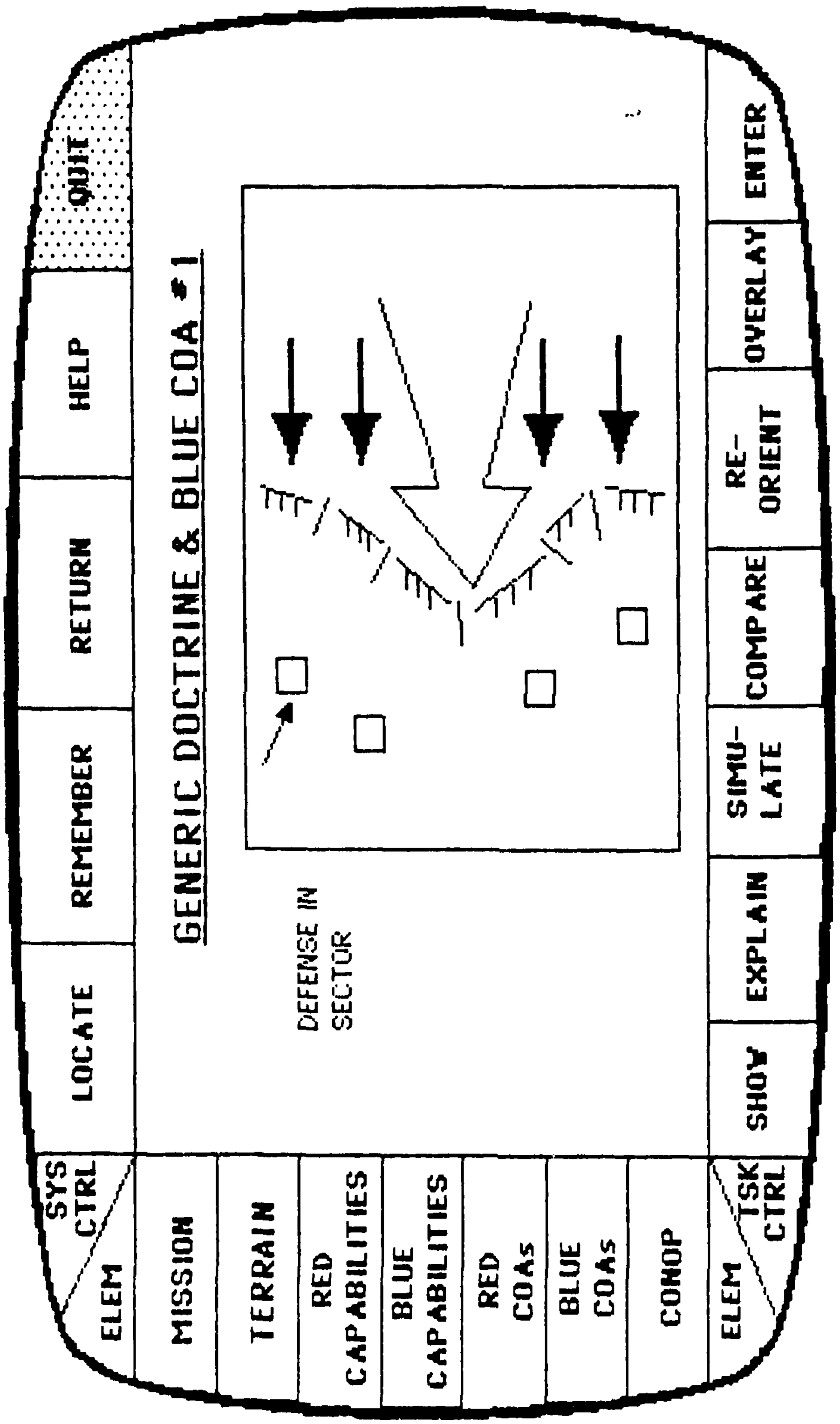
Generic doctrinal display . . .



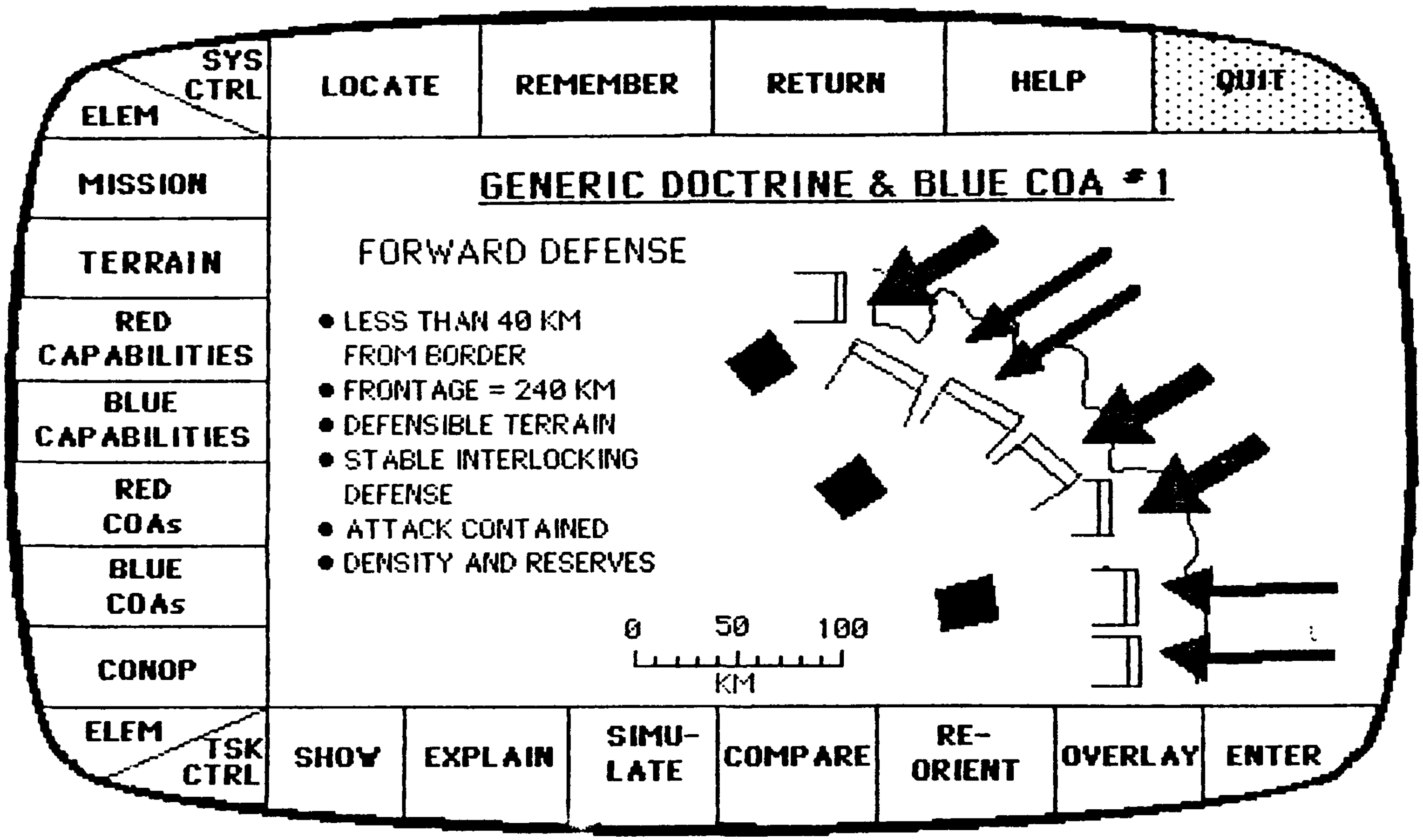
Generic doctrinal display . . .



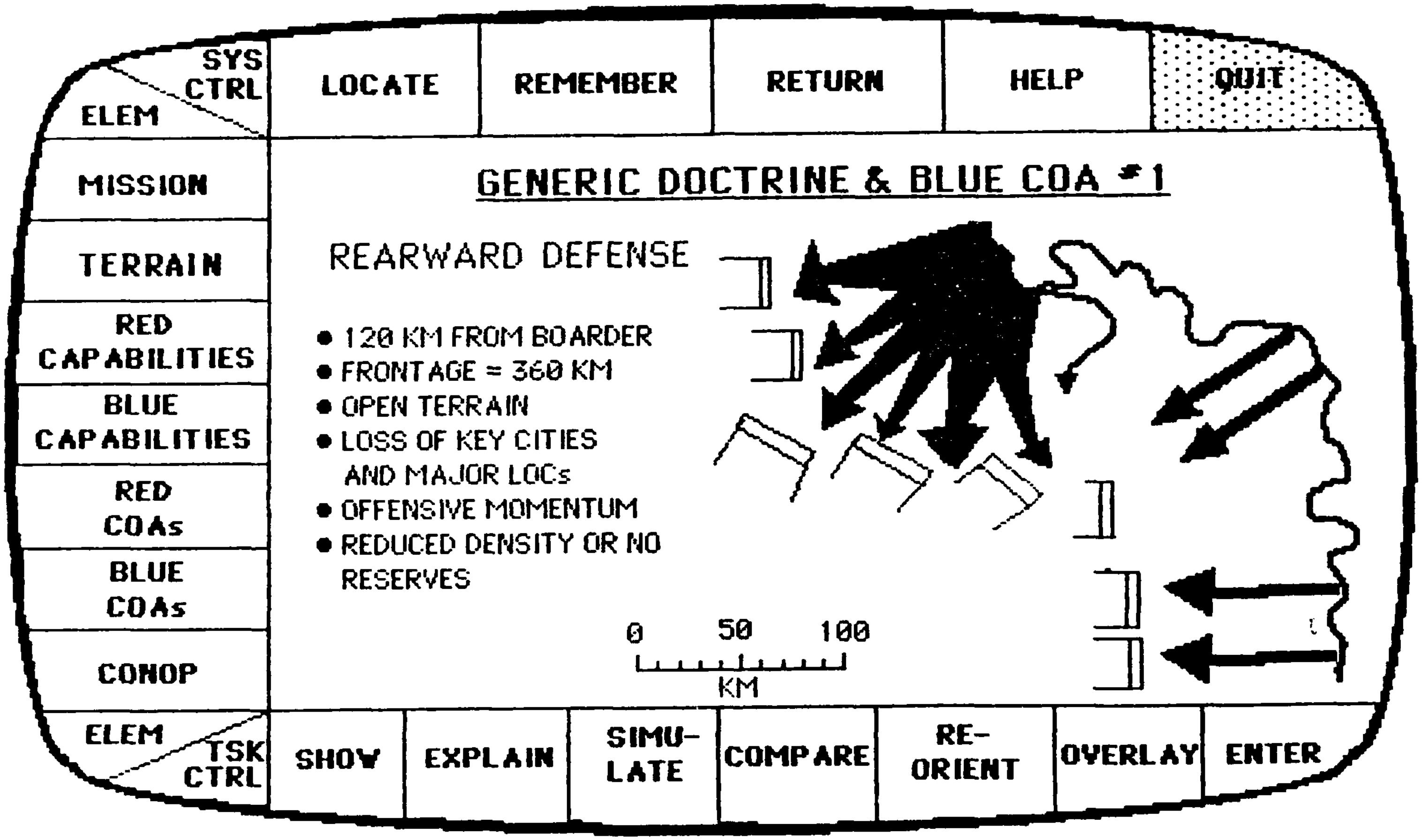
Generic doctrinal display . . .



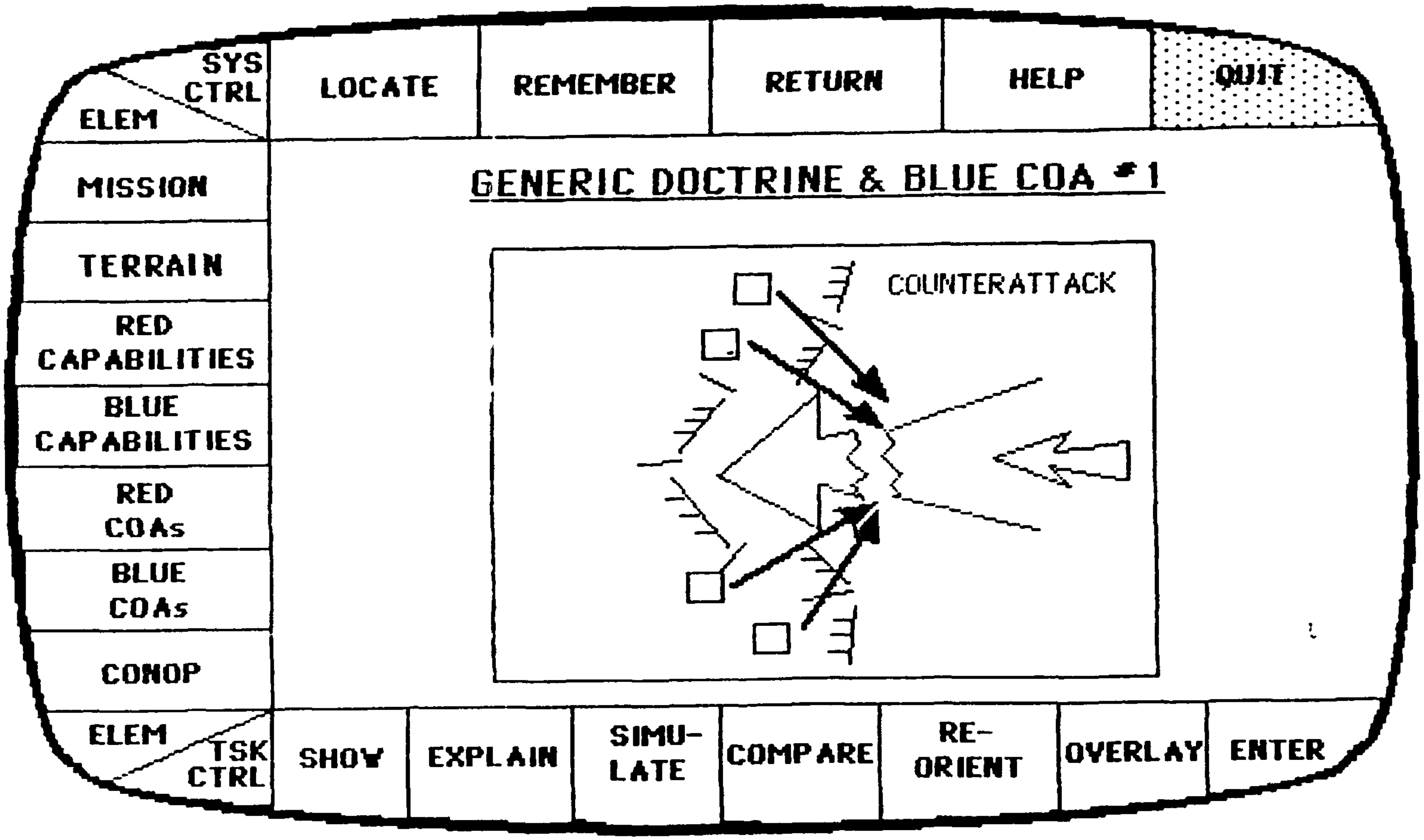
Generic doctrinal display . . .



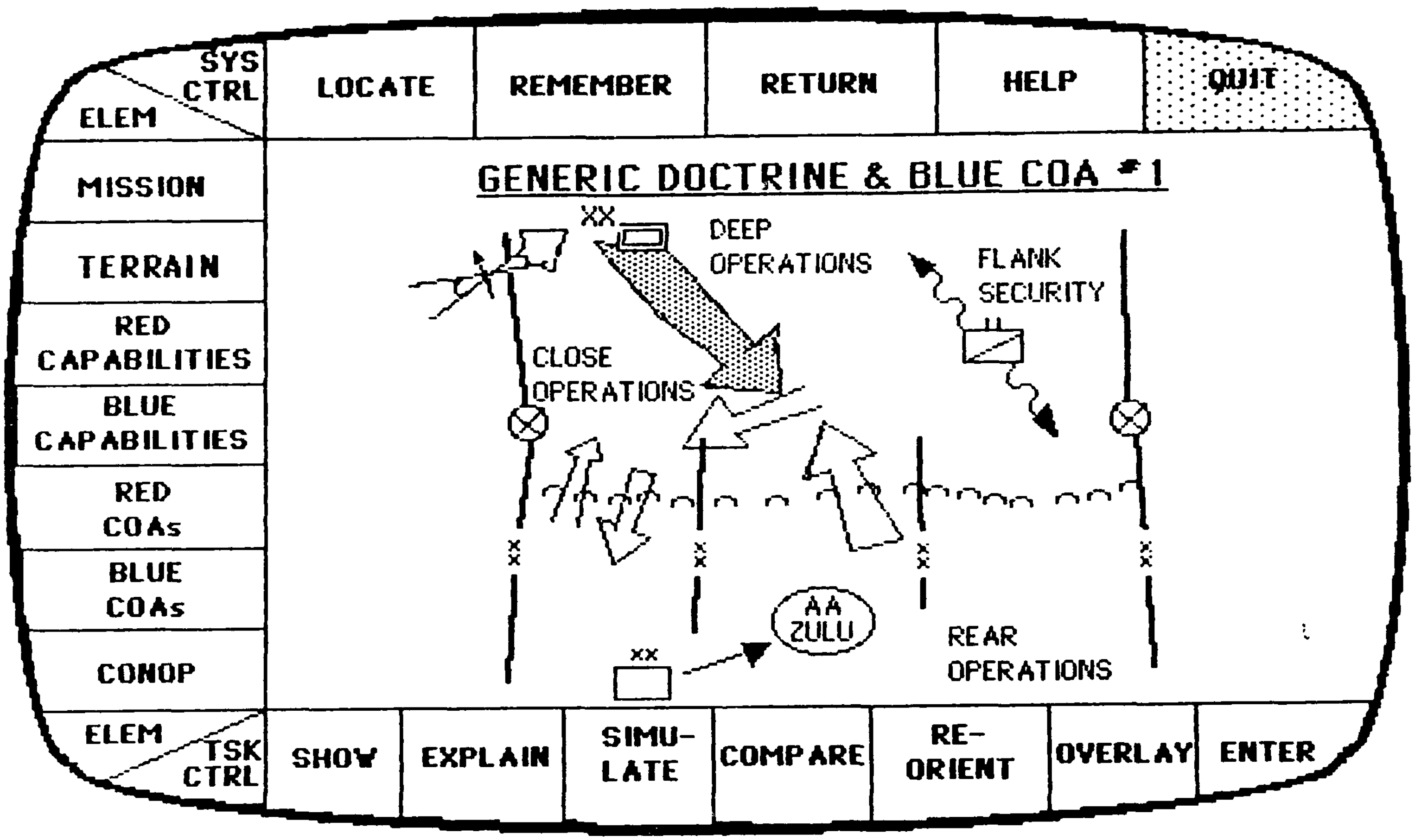
Generic doctrinal display . . .



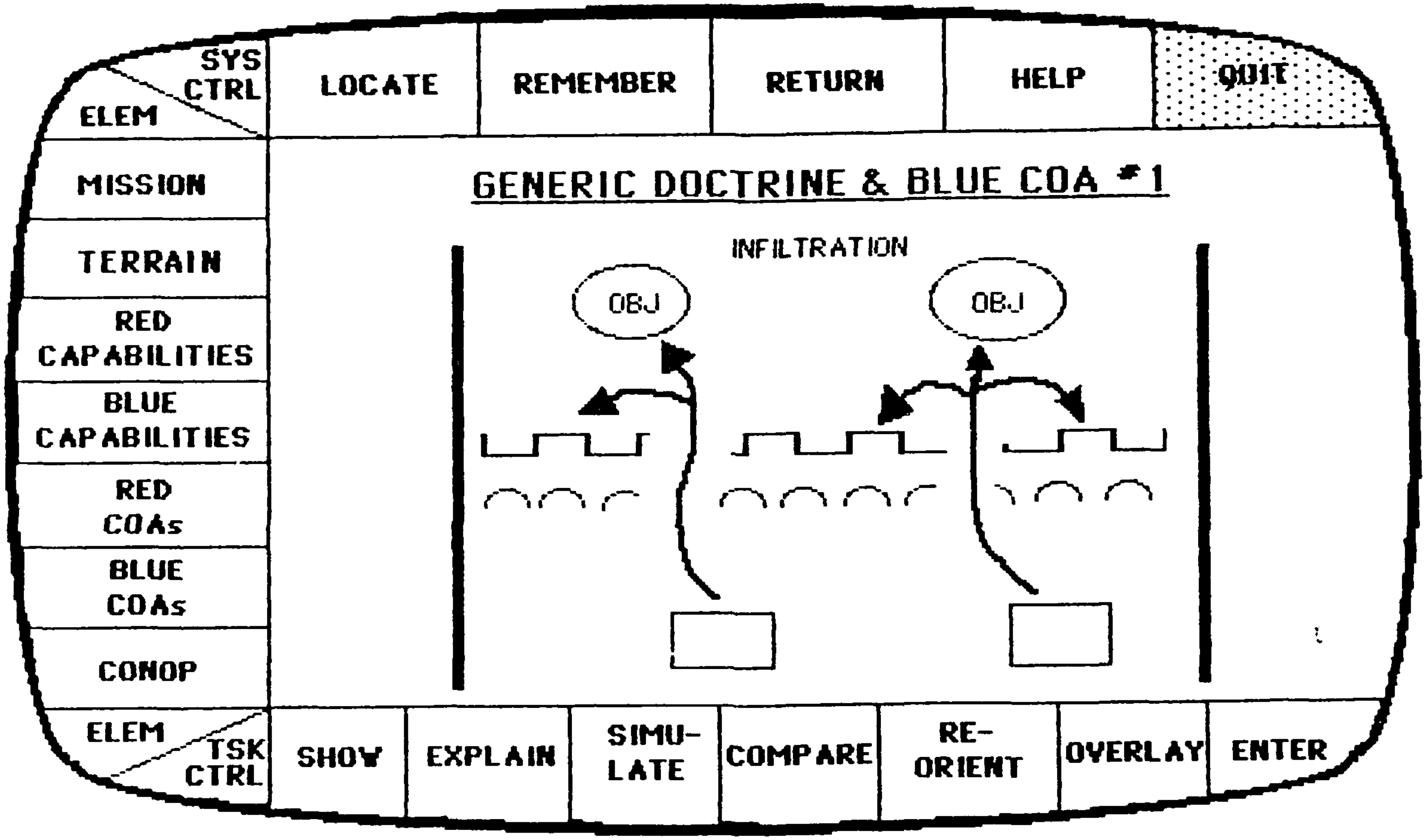
Generic doctrinal display . . .



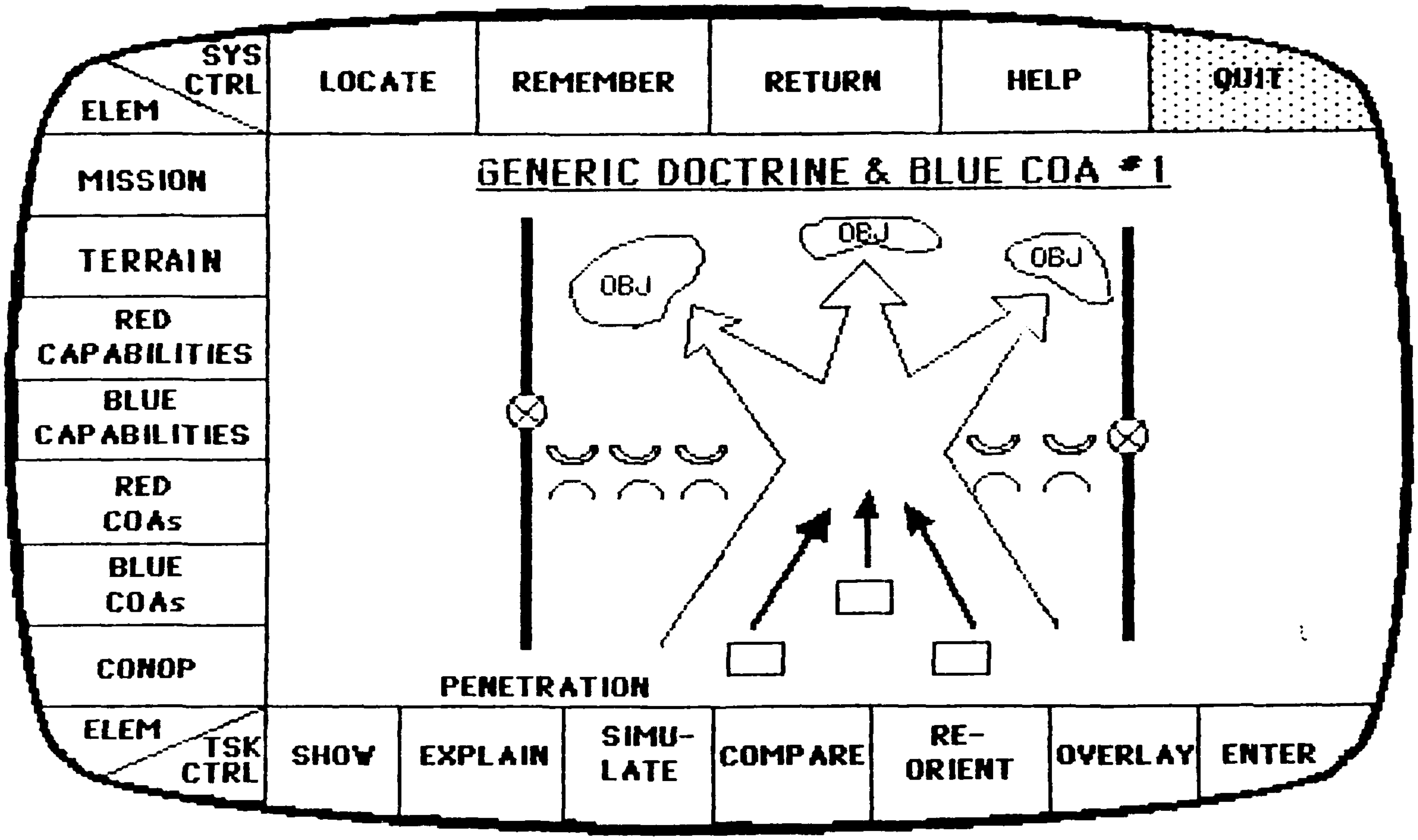
Generic doctrinal display . . .



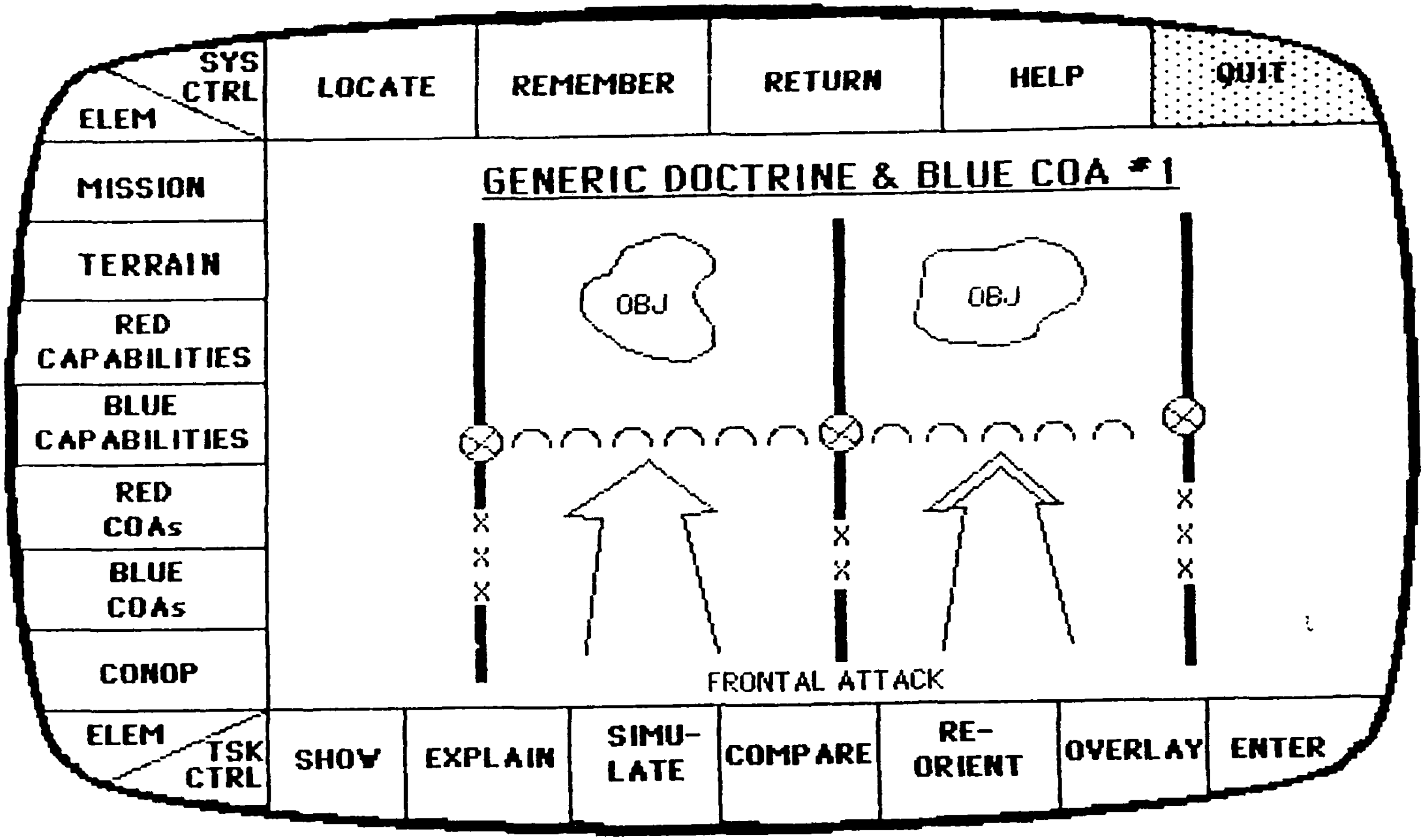
Generic doctrinal display . . .



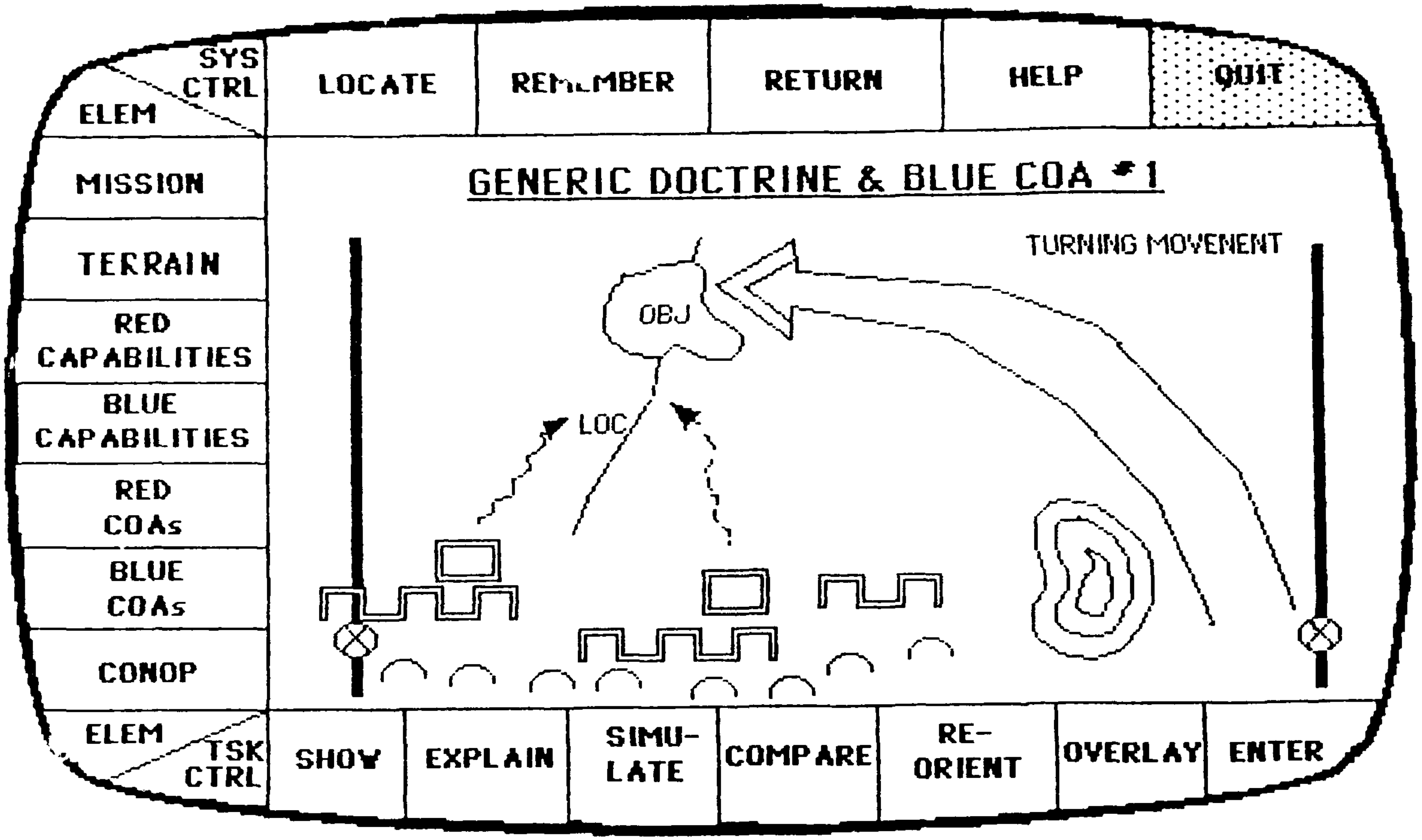
Generic doctrinal display . . .



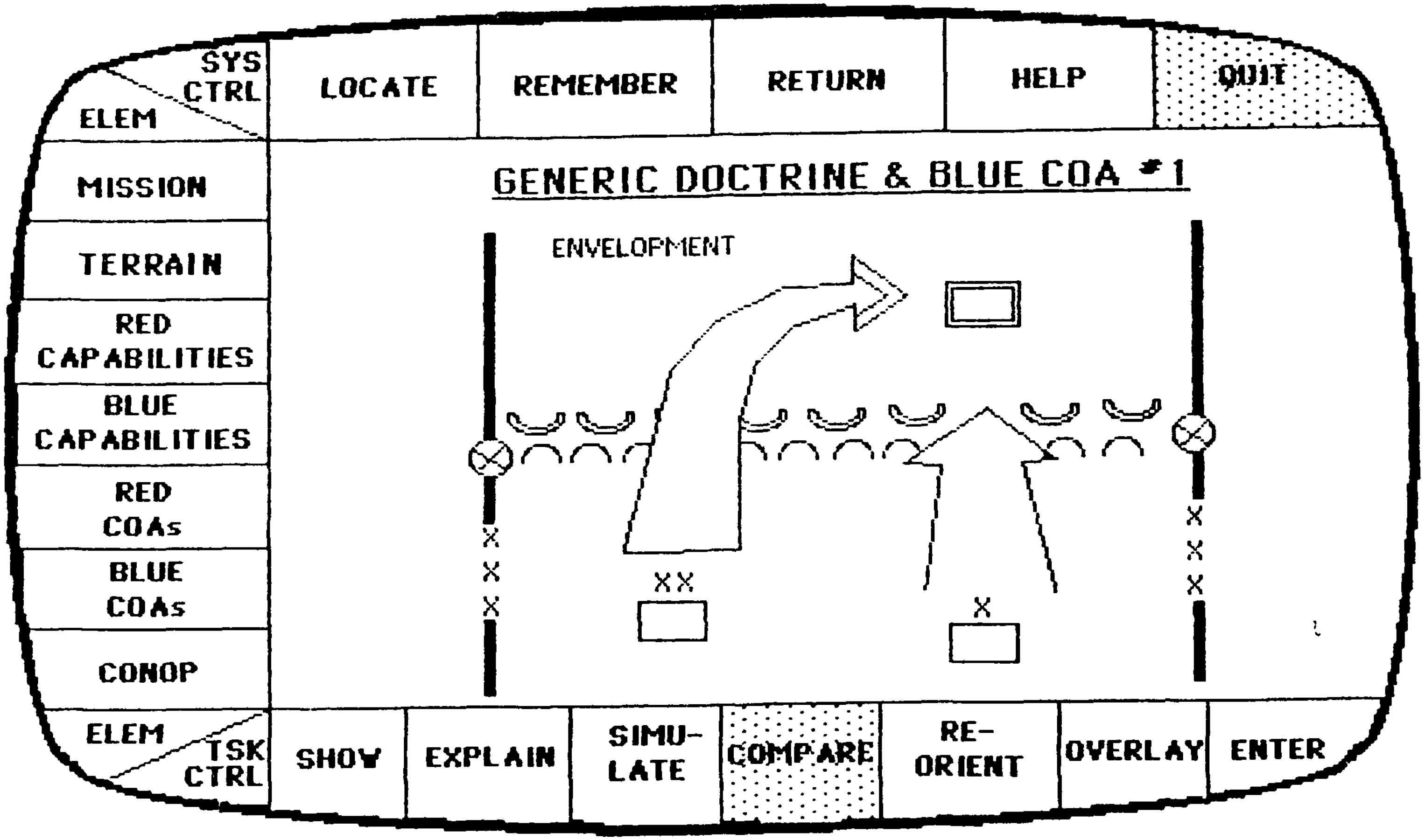
Generic doctrinal display . . .



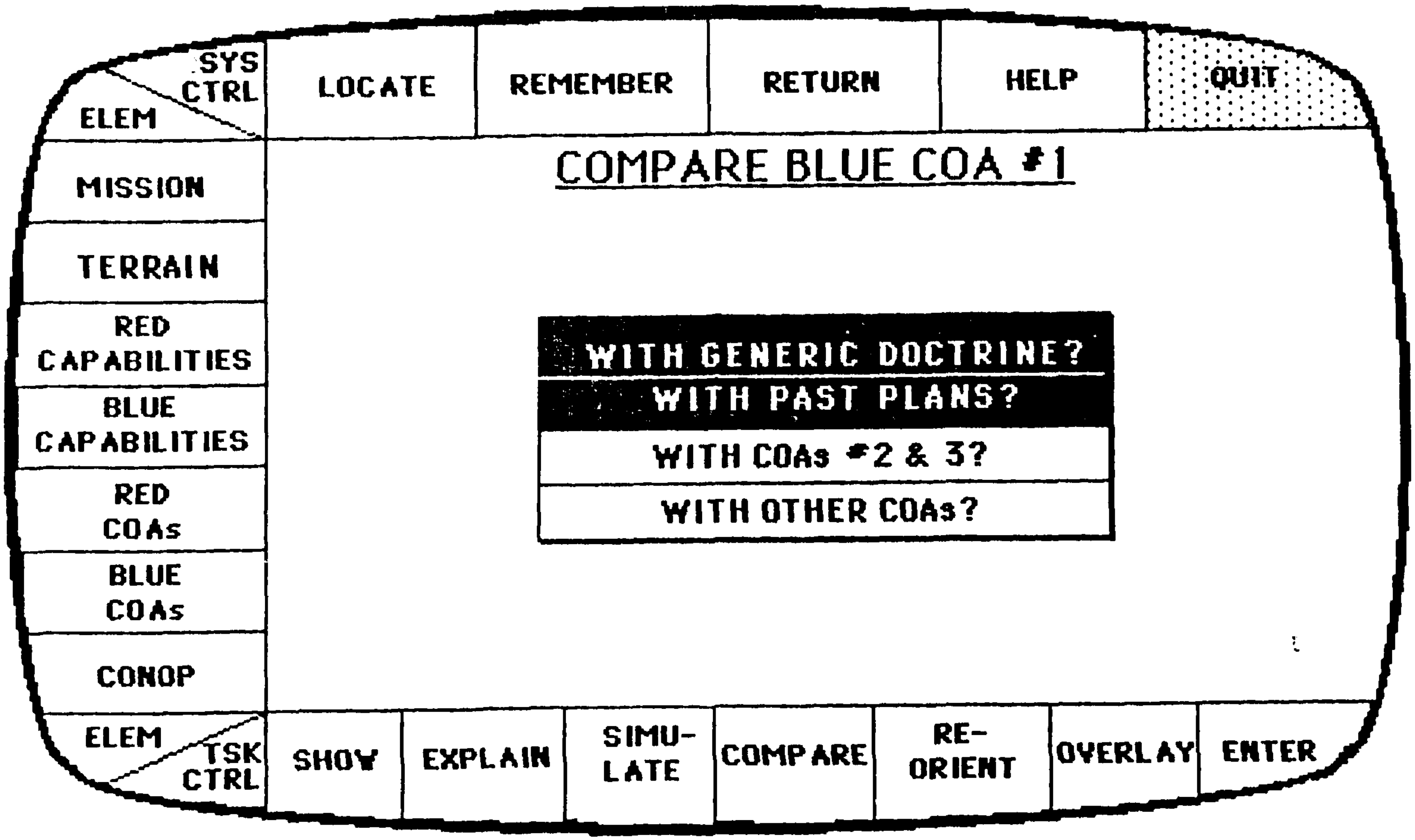
Generic doctrinal display . . .



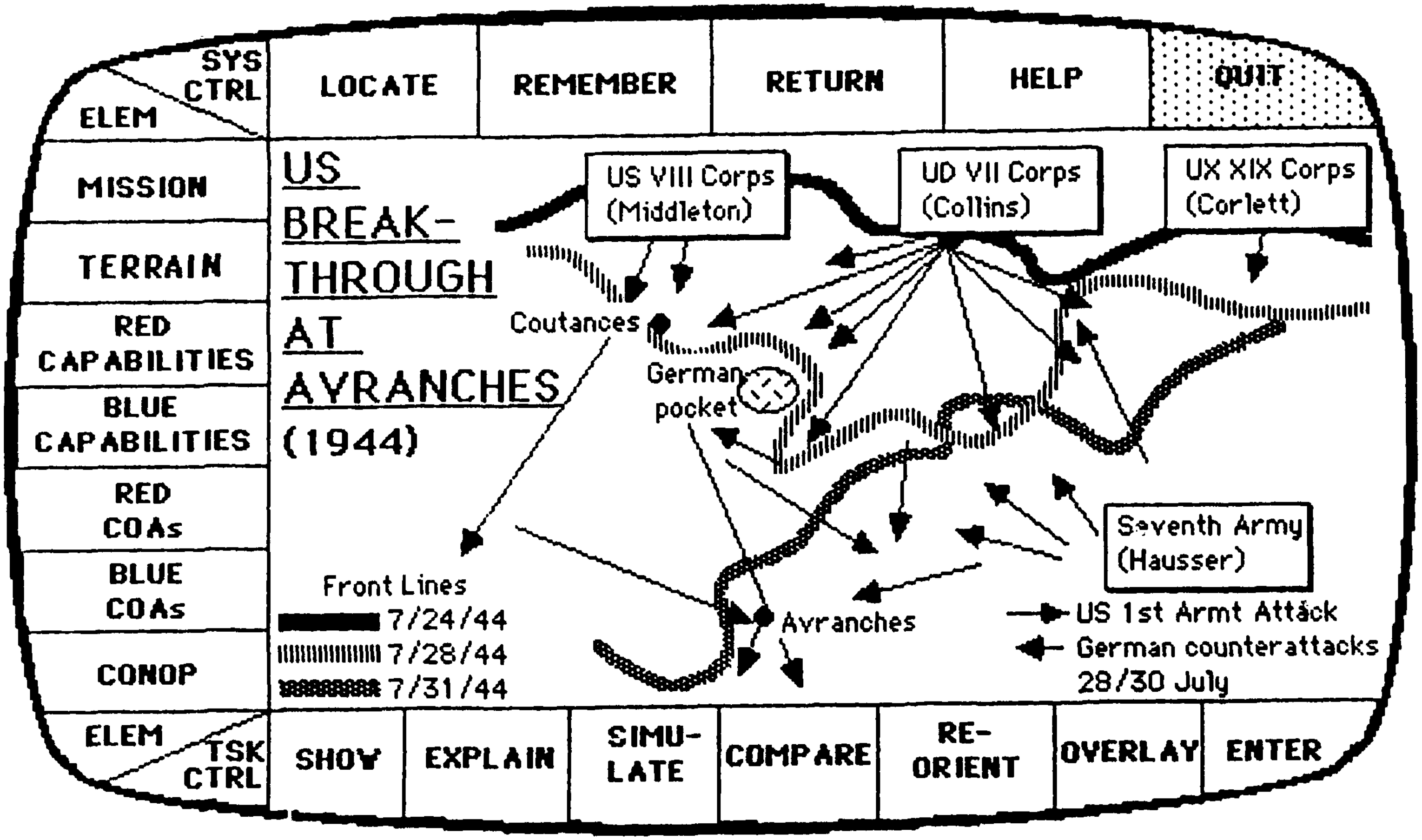
Generic doctrinal display . . .



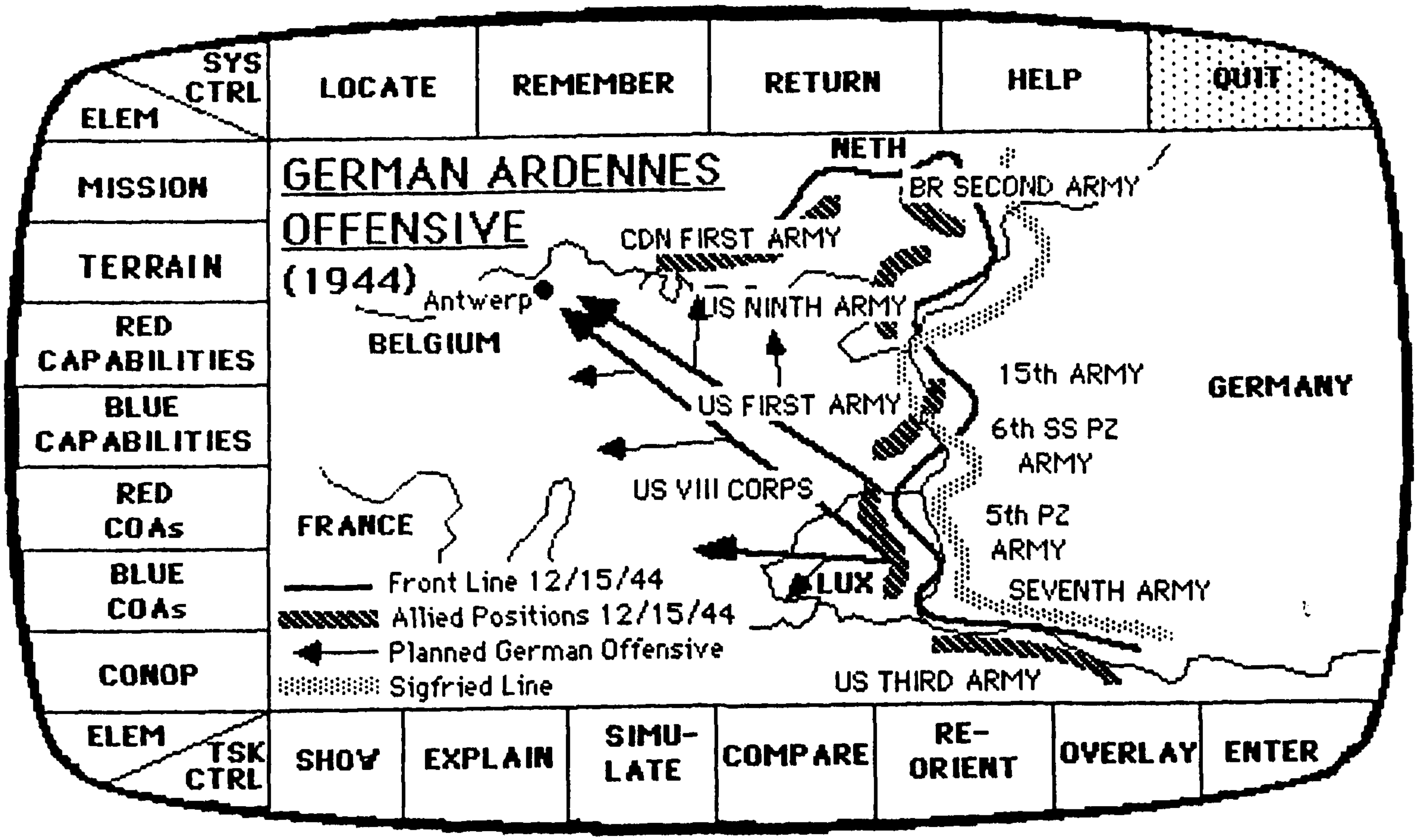
Generic doctrinal display . . .



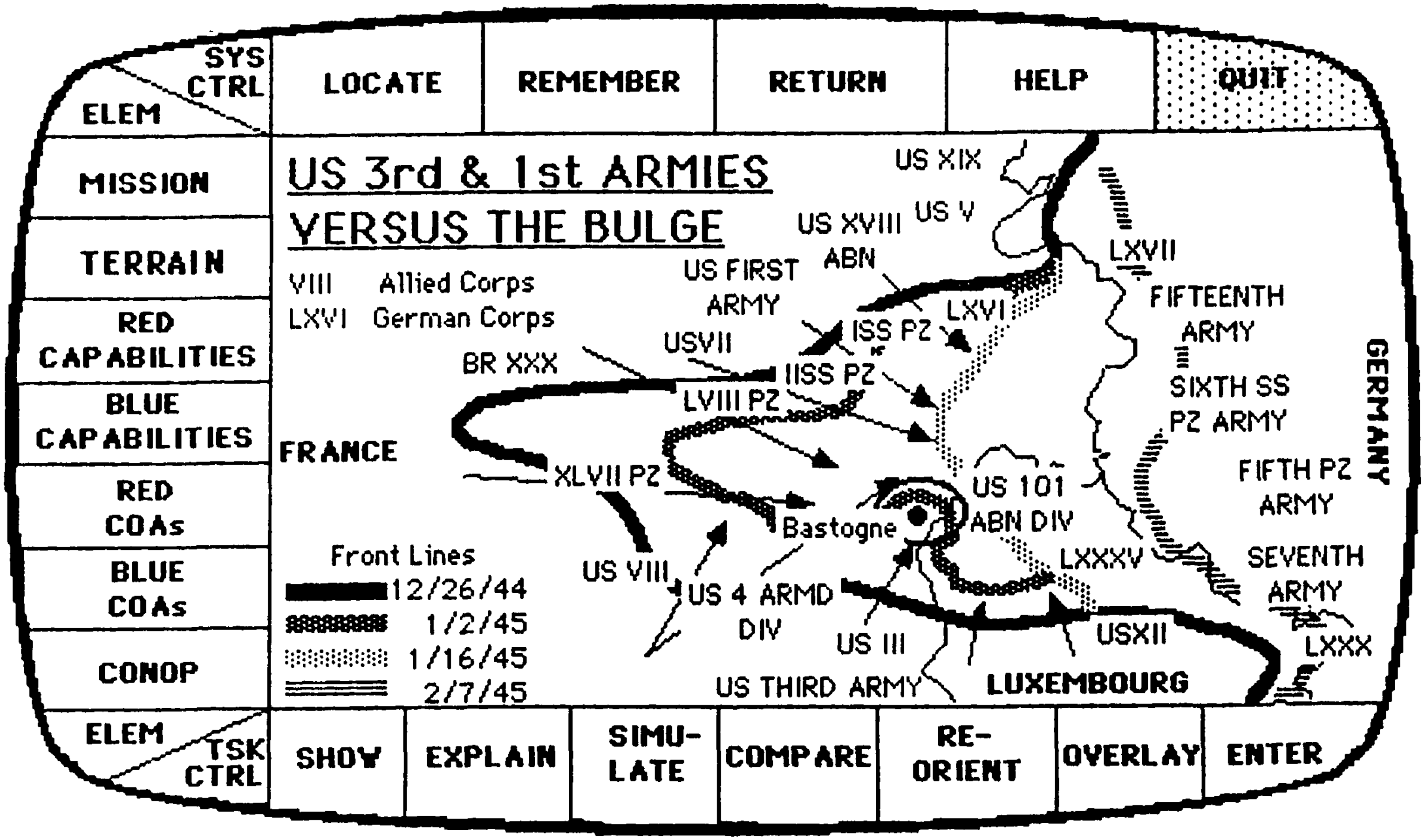
The system permits additional comparative options . . . and the planner selects the option permitting comparison with past plans . . . this option suggests that the system selects from a set of stored plans those that are especially relevant to the planning problem at hand and then presents them to the planner to help him decide if a specific plan is appropriate or not . . . like the previous generic option this one supports analogical reasoning . . .



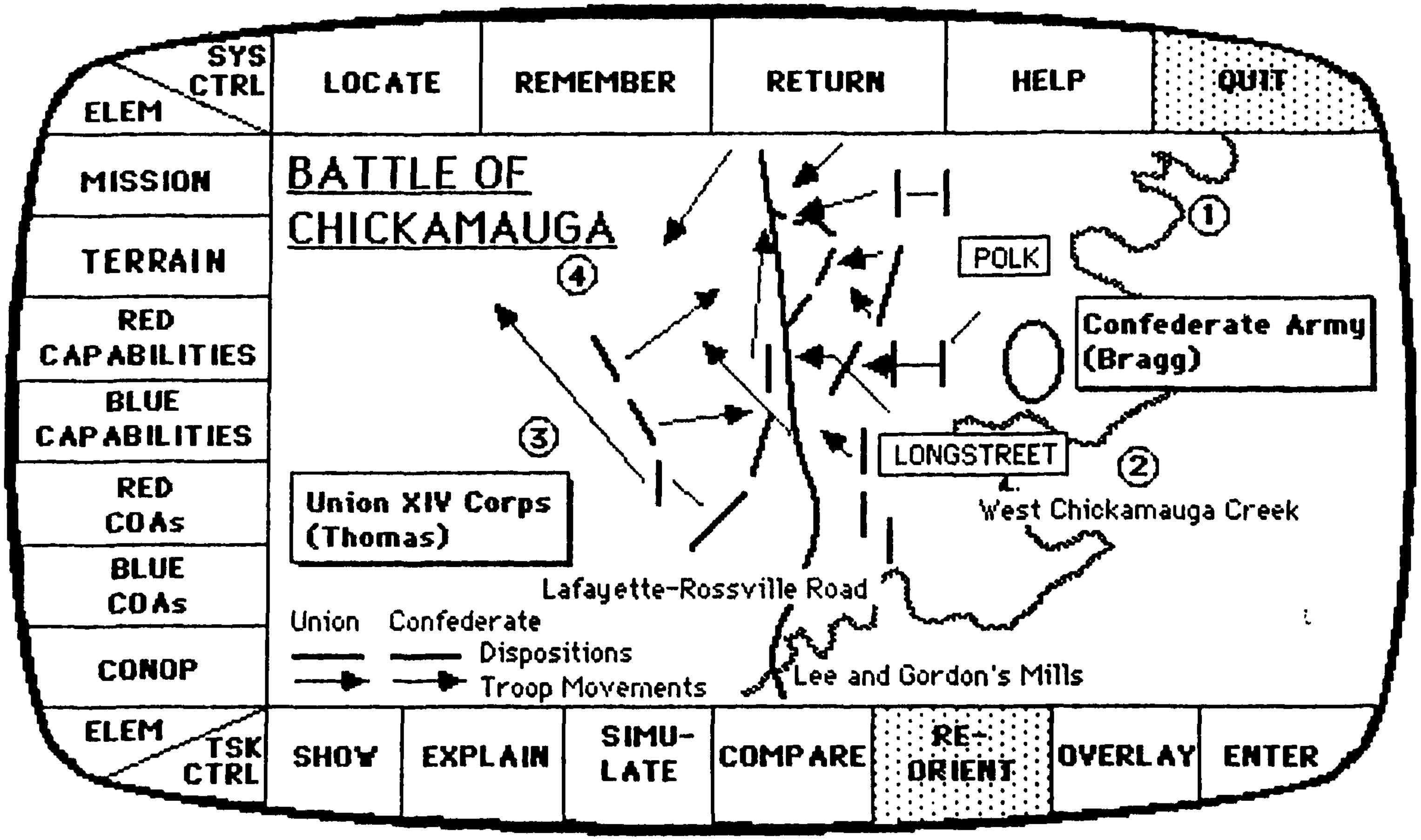
The system presents a set of plans that are relevant to the planning problem at hand . . .



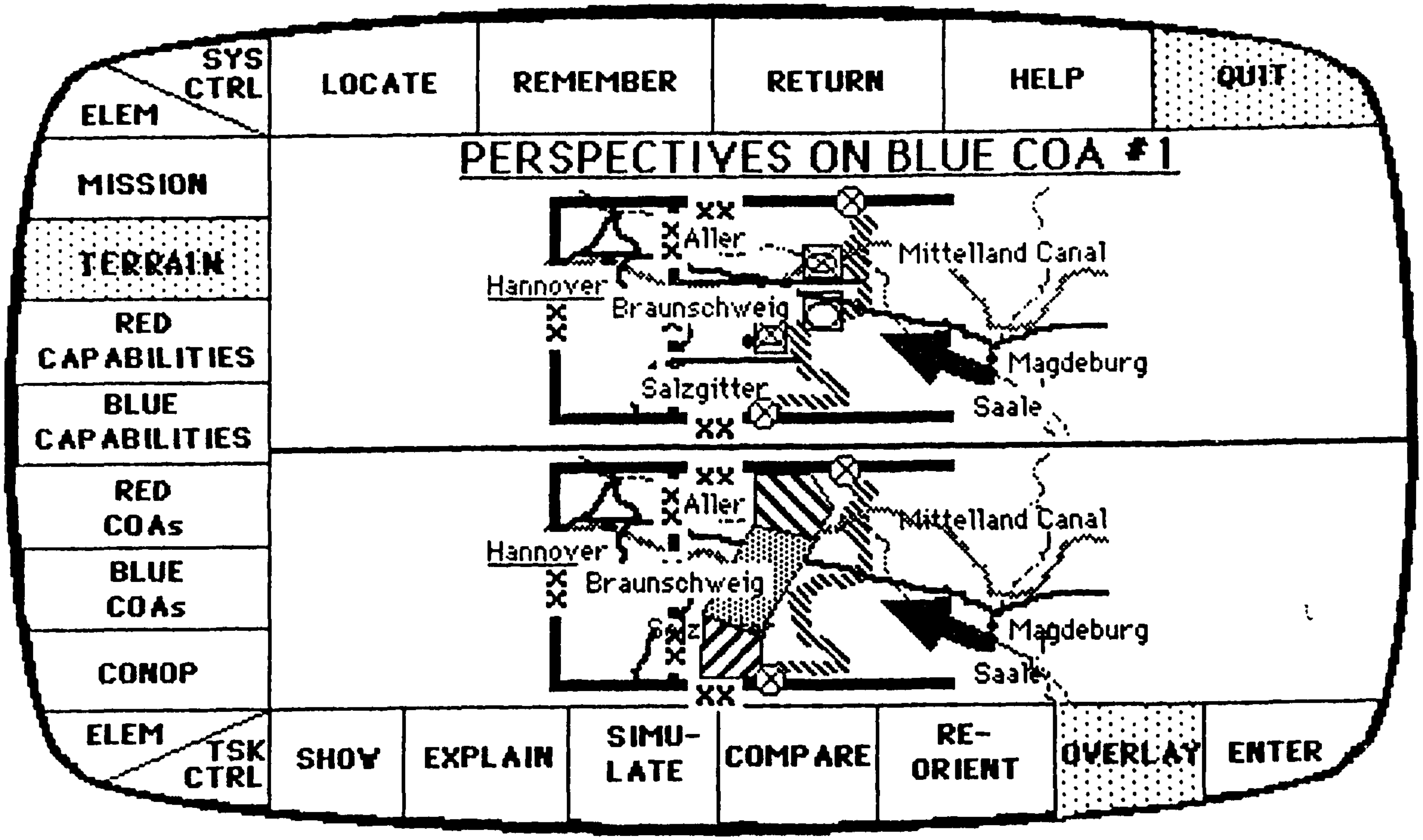
Additional past plans are displayed . . .



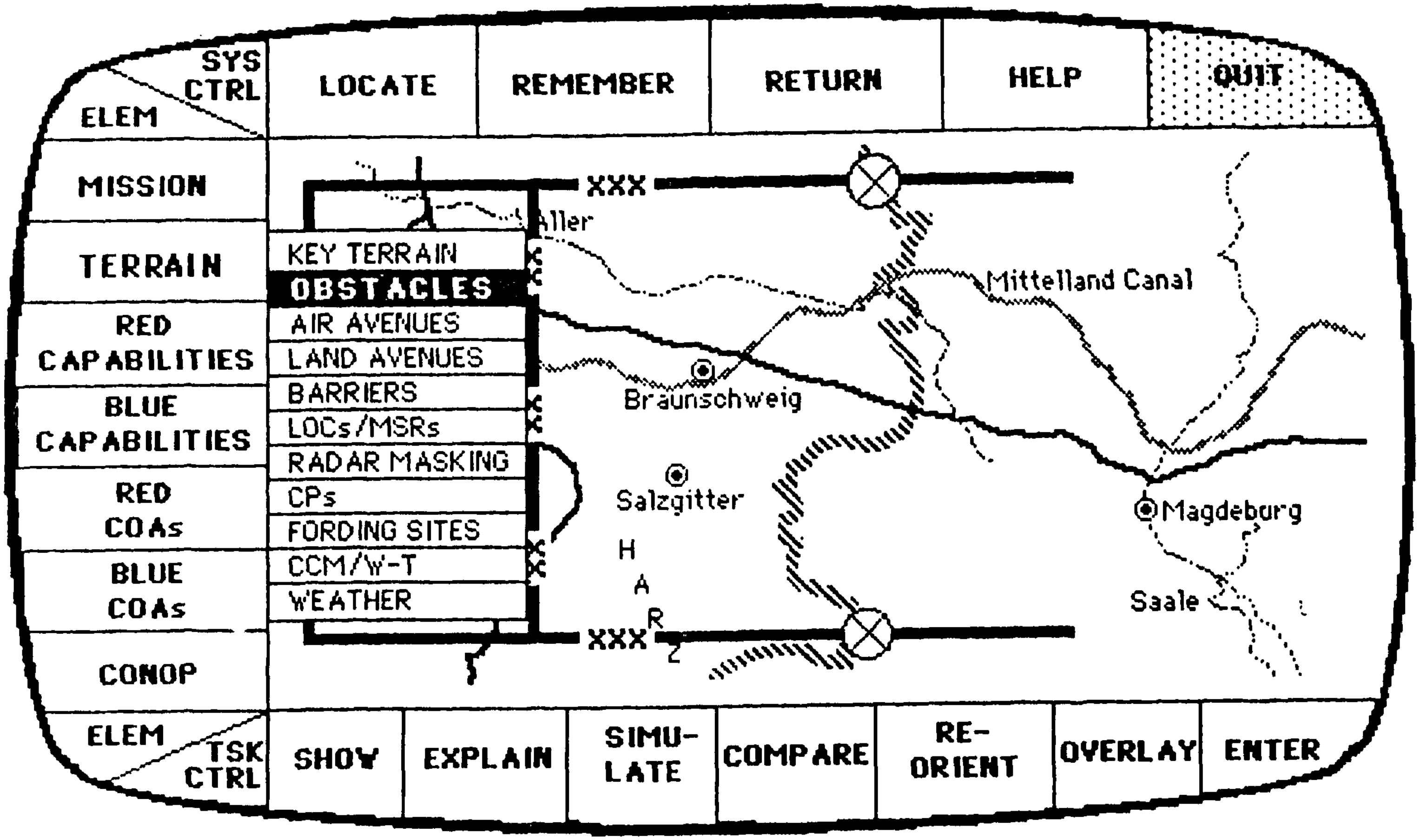
Additional past plans are displayed . . .



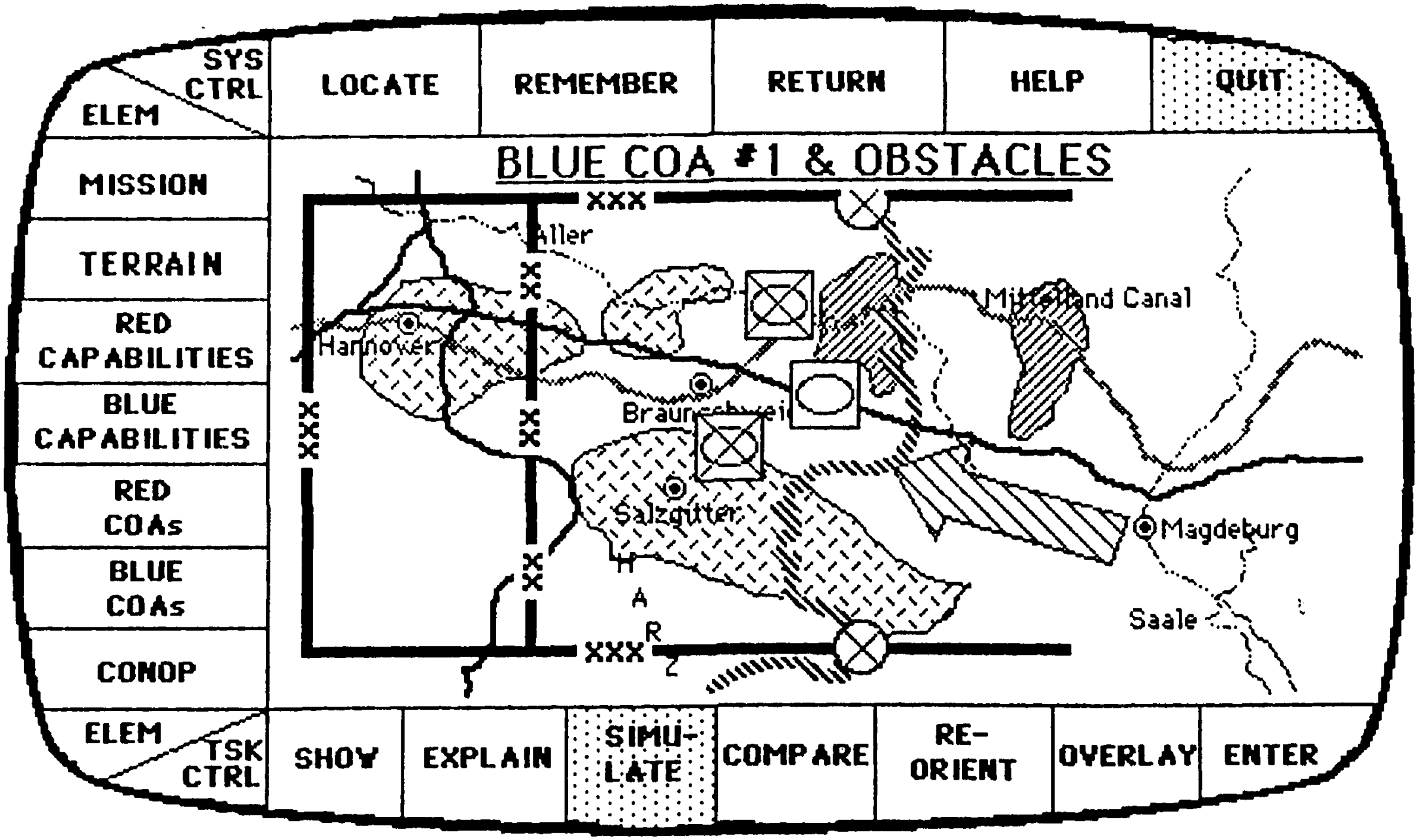
Additional past plans are displayed . . .



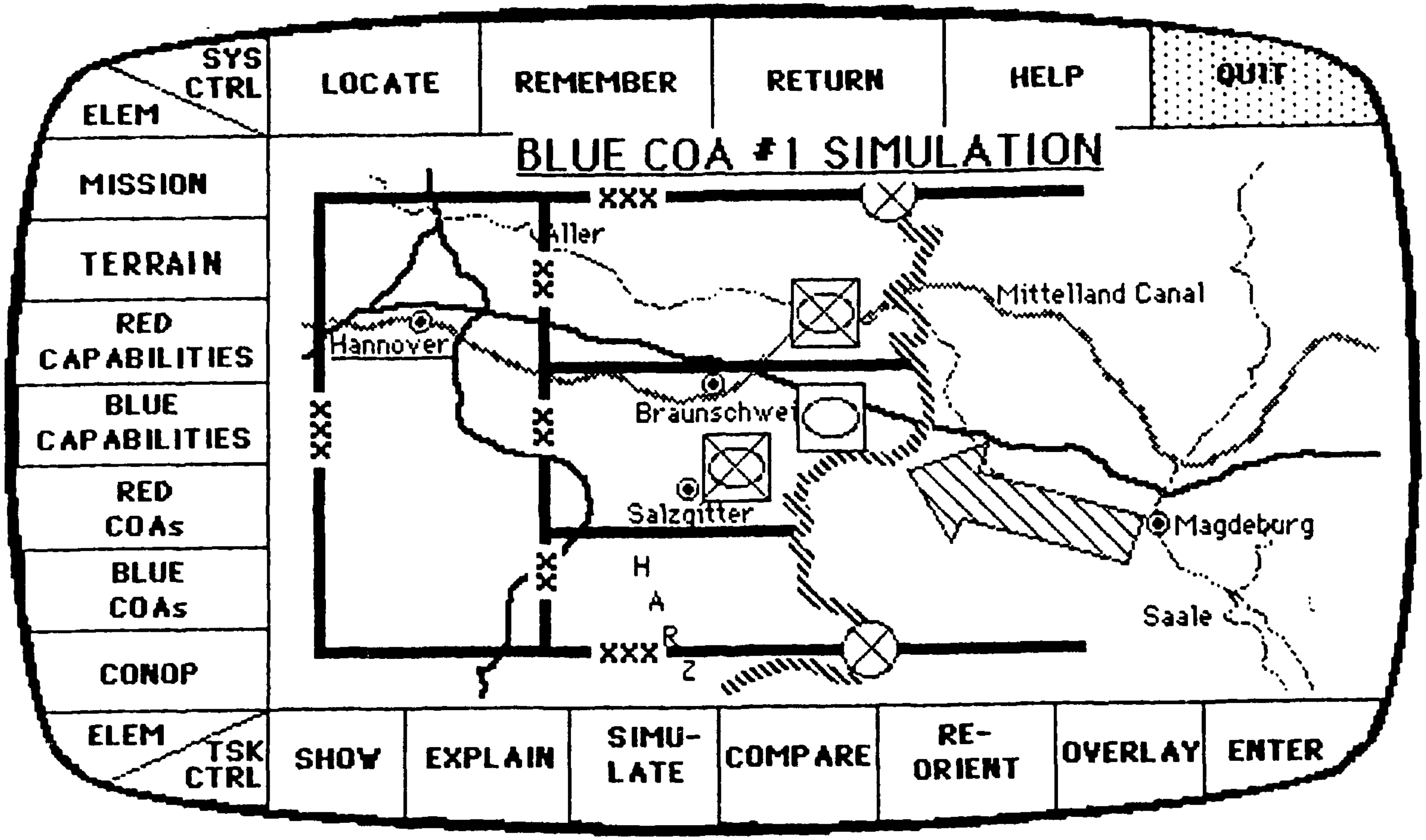
Having selected the **Re-Orient** option the system presents a split screen of **Blue COA #1** illustrating that a COA is more than the location of certain units on a battlefield but rather a "wall" that will prevent Red from penetrating certain areas . . . the planner then clikcs on **Overlay** . . .



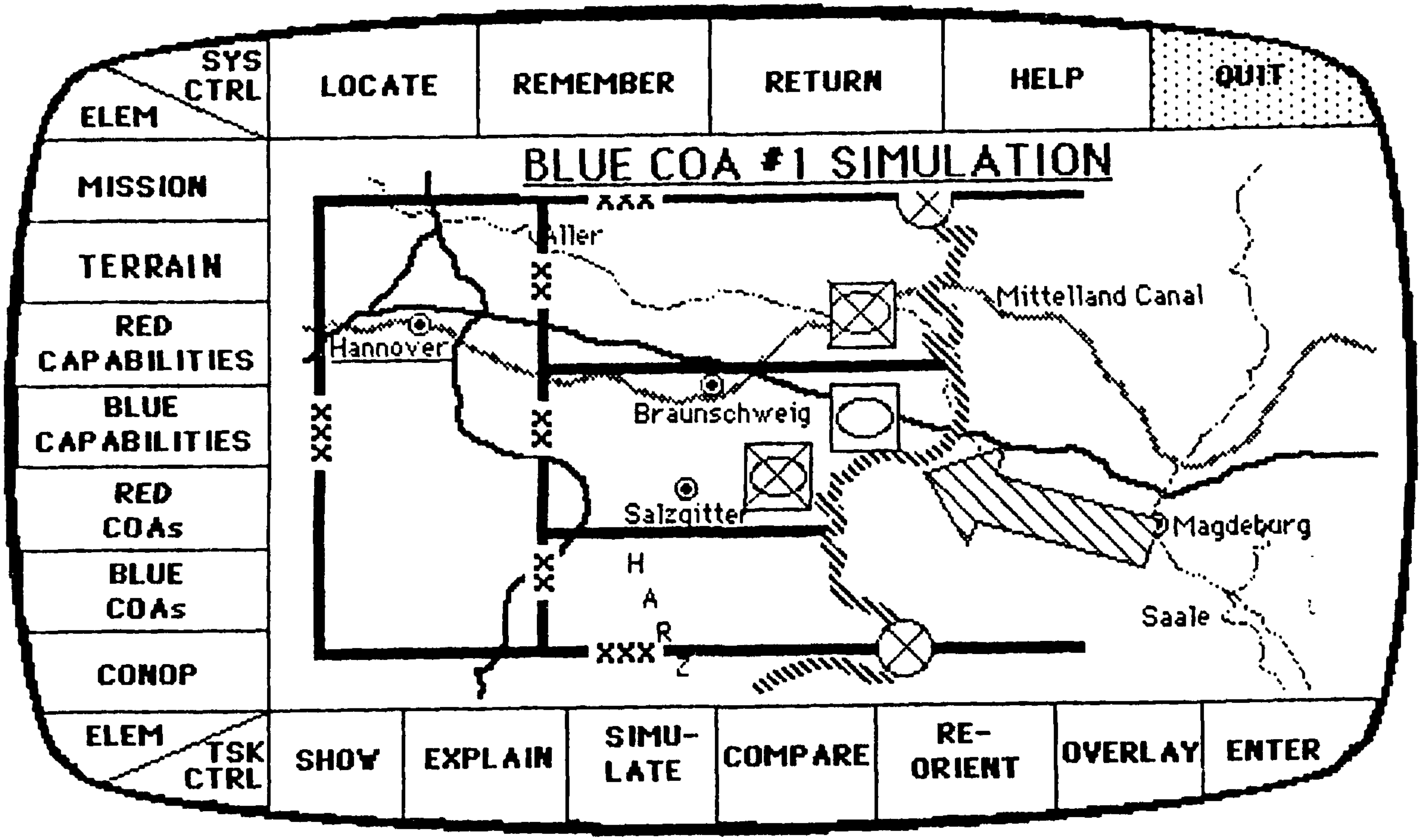
The system asks him to select what he wants to **Overlay** . . .



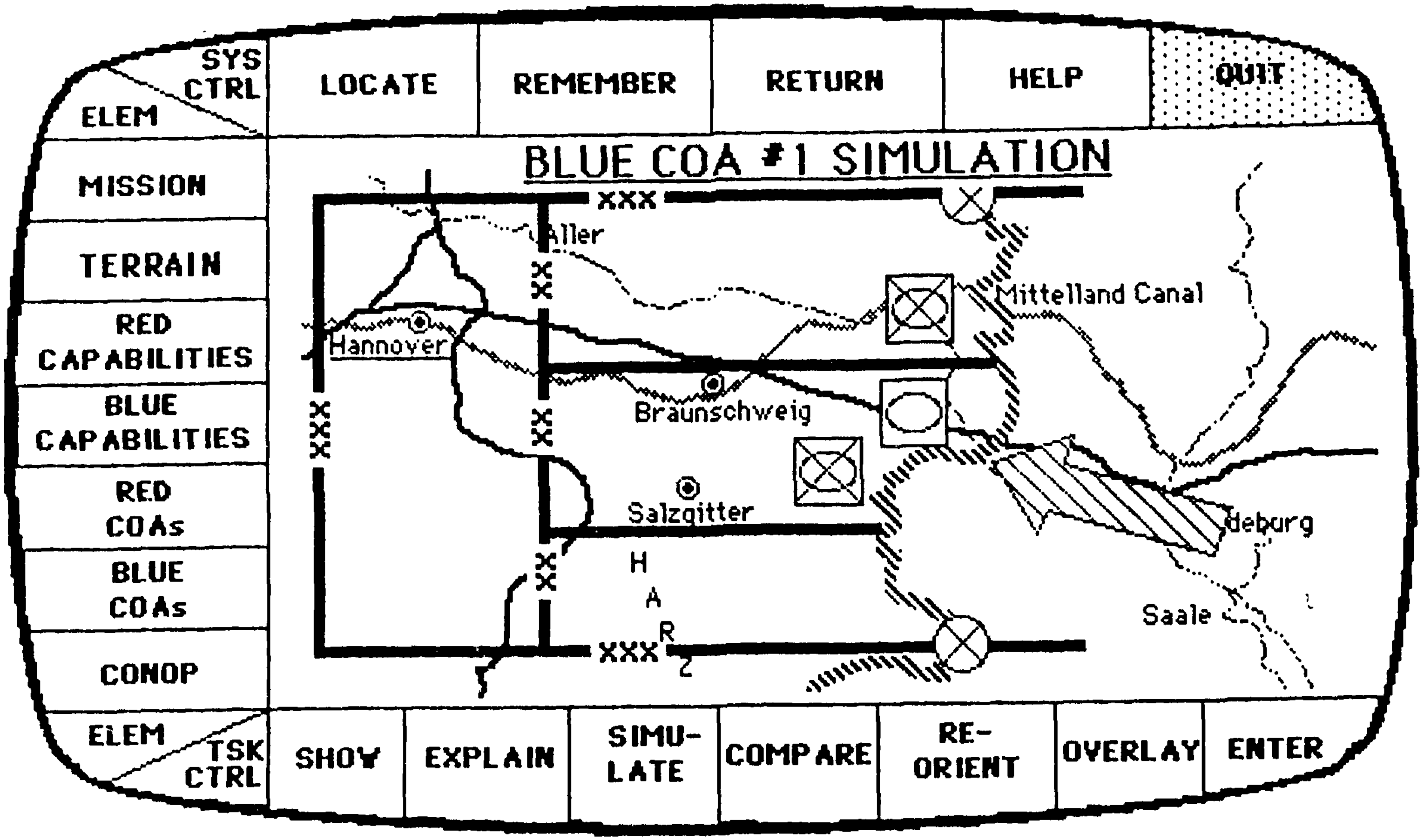
The system displays **Blue COA #1** and **Terrain Obstacles** . . . the planner then clicks on **Simulate** to "see" the COA played out in a simulated battle . . .



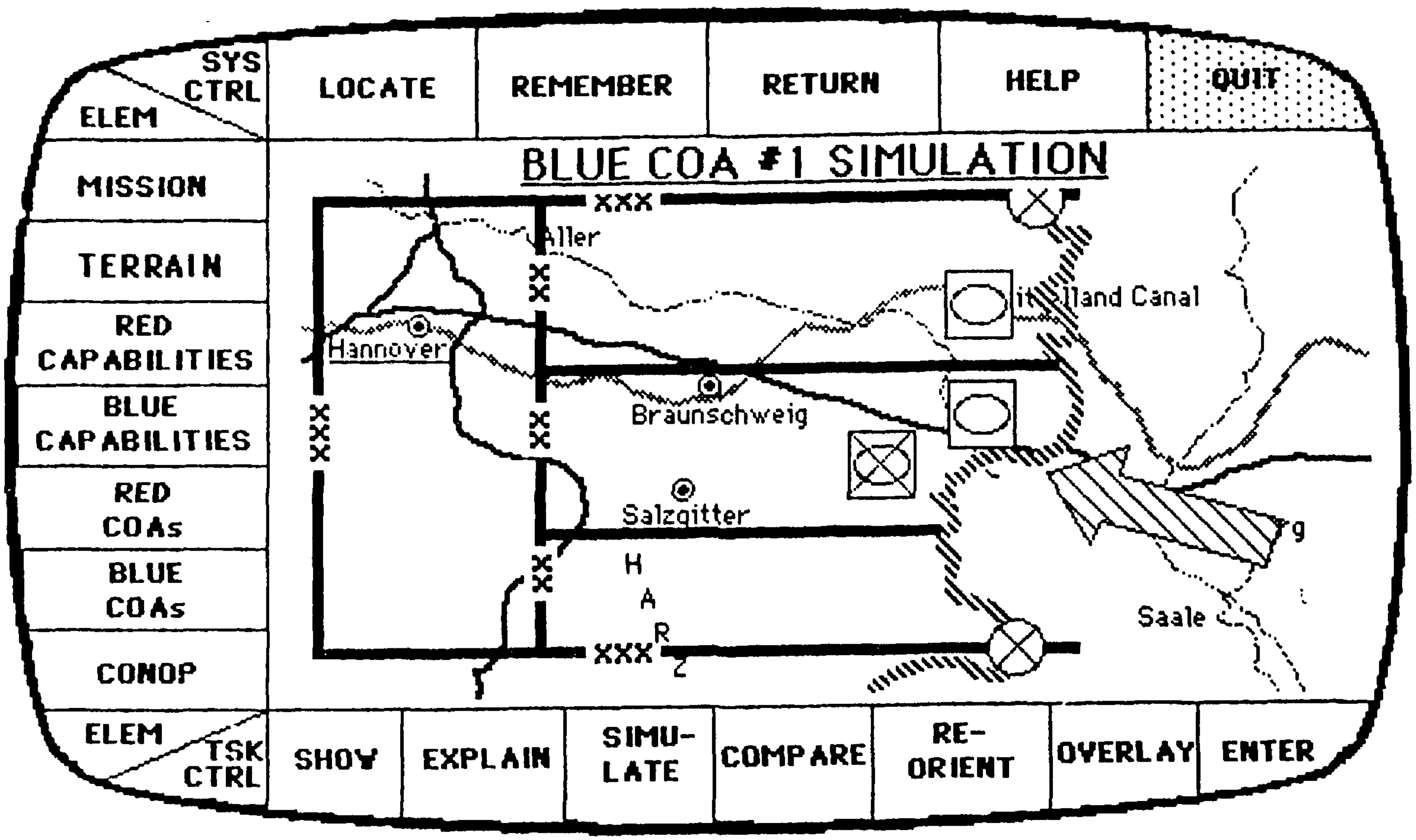
The system displays the animated battlefield . . .



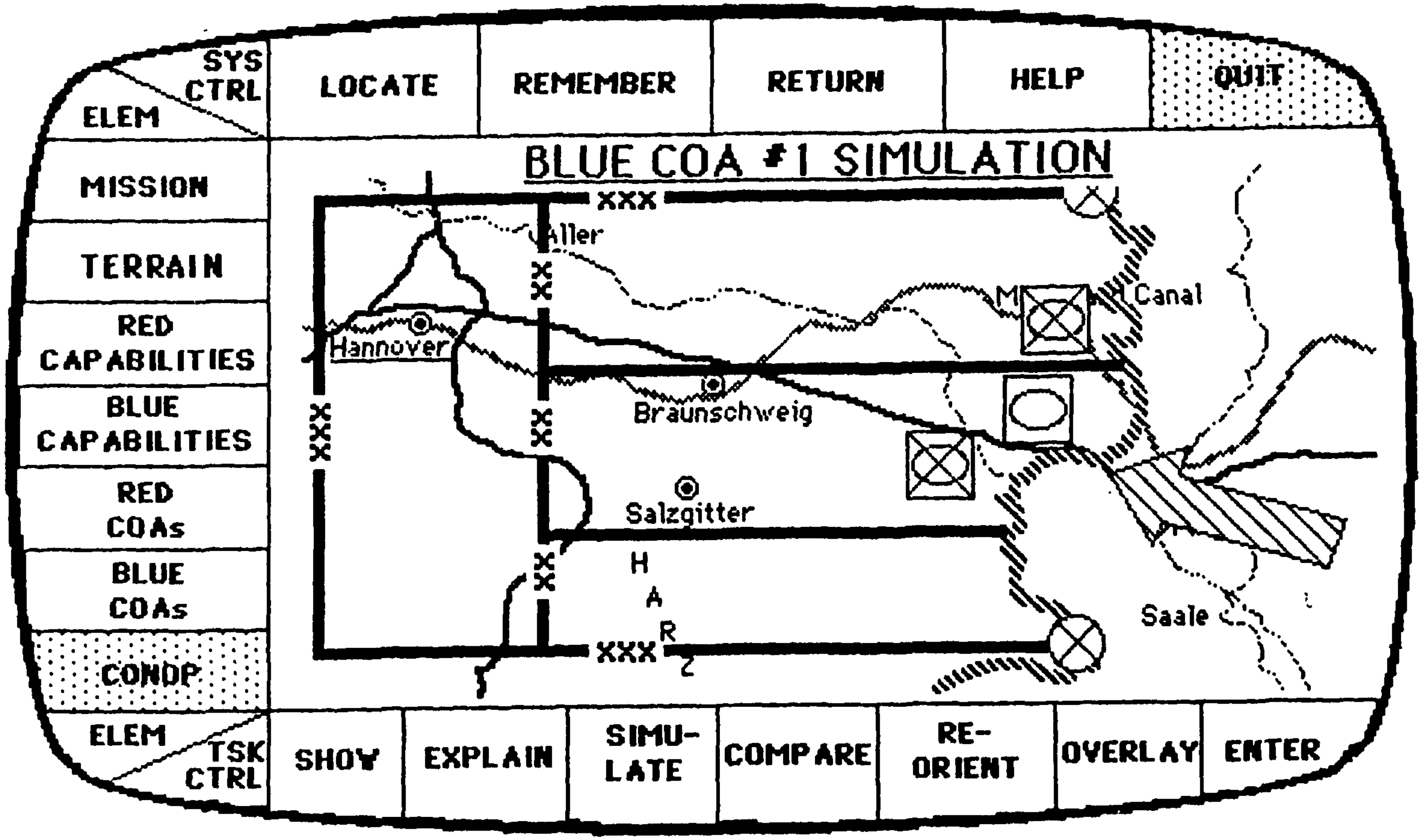
The system displays the animated battlefield . . .



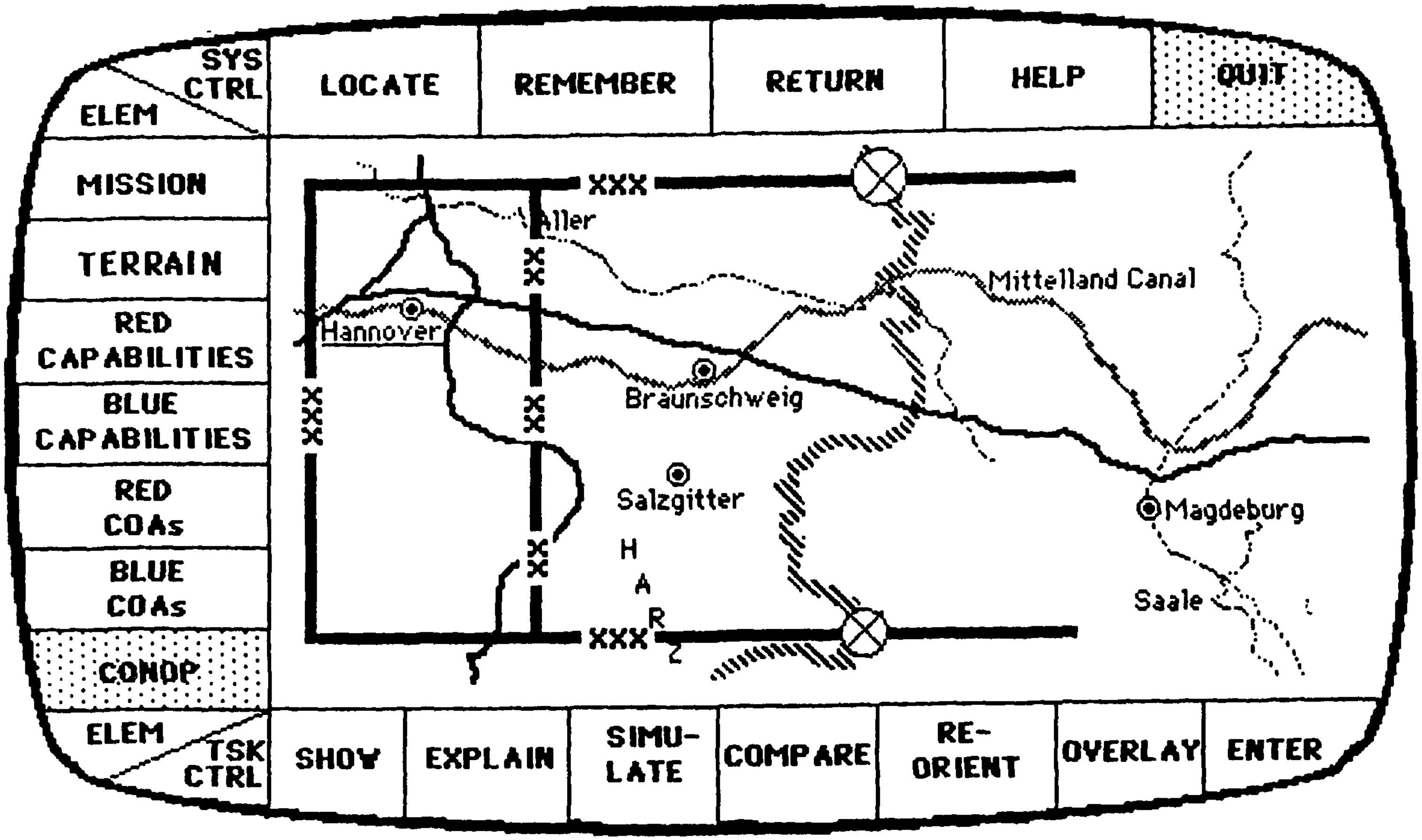
The system displays the animated battlefield . . .



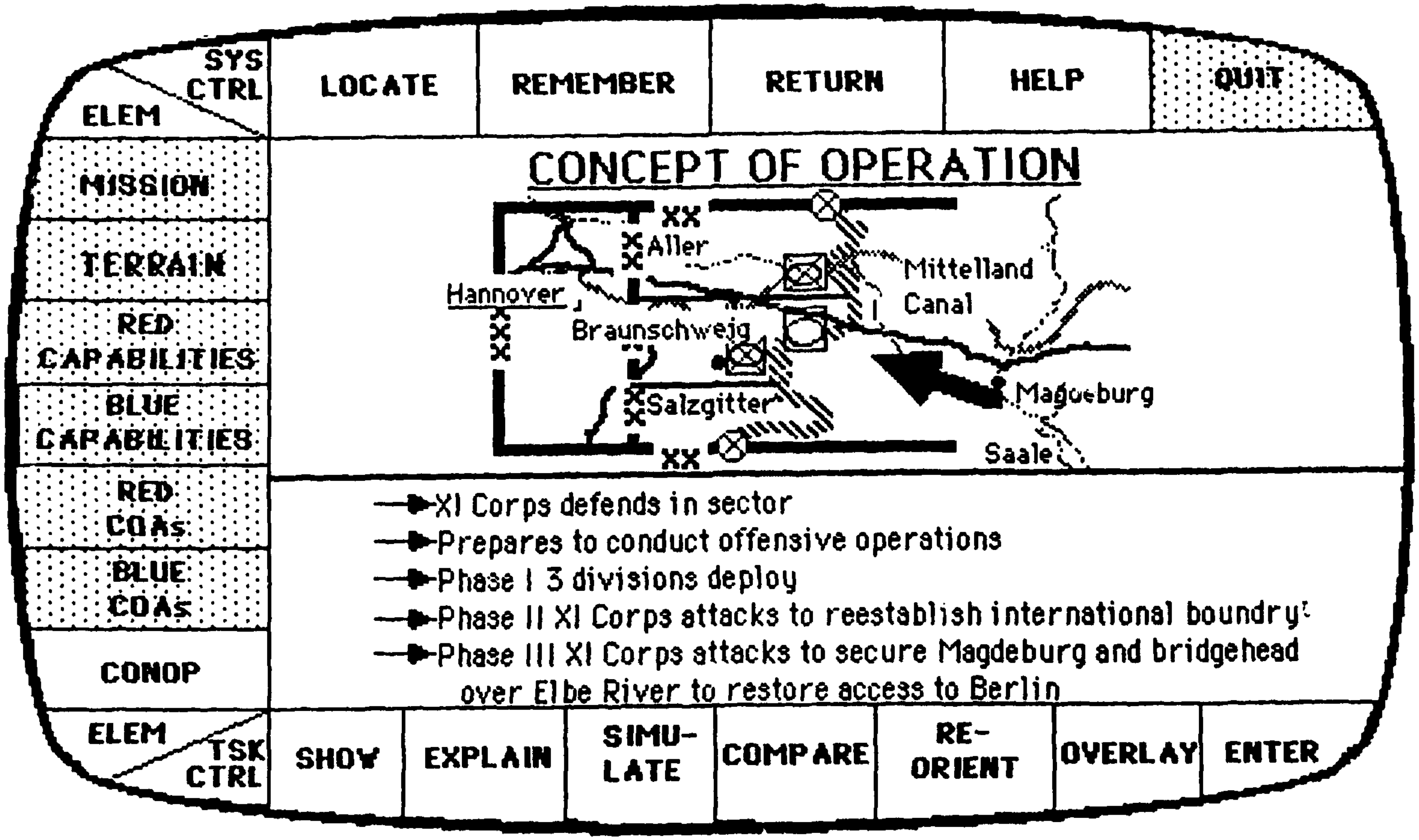
The system displays the animated battlefield . . .



The system displays the animated battlefield . . .



The user selects the **Concept of Operations (CONOP)** option . . .



The **CONOP** is displayed to the planner . . .

5.0 TESTING AND EVALUATION

5.1 Storyboard "Sizing"

There is an enormous gap between a storyboard and an actual system. Storyboards are intended to validate requirements and explore system concepts. Our mission on this project was to explore a new enhanced UCI system concept. However, as the storyboard suggests, it is impossible to conceive of an advanced user-computer interface in the abstract; it was necessary (and appropriate) for us to design the enhanced UCI in a substantive context (tactical planning).

This section of the report identifies and describes the steps necessary to move from the storyboard *prototype* to an actual (working prototype) system. By way of diversion, there are at least two kinds of prototypes, the throwaway and evolutionary. Storyboard prototypes (before Hypercard!) are historically "throwaway" prototypes; prototypes based upon working code are often described as "evolutionary" prototypes. Throwaway prototypes are usually developed when requirements are especially difficult to capture and validate; evolutionary prototypes are developed when requirements are better understood initially. The nature of this "innovative research" proposal necessarily called for the design and development of a throwaway; this section of the report deals with the issues necessary to move from a throwaway to an evolutionary prototype. This, however, is not to suggest we are starting at "square one" with the evolutionary prototype; quite to the contrary, our throwaway will permit us to hit the ground running on the working prototype and develop a system that will satisfy requirements (in a related domain) and permit us to demonstrate a credible system to Army operational personnel.

5.1.1 <u>Data and Knowledge Base Requirements</u> - As the storyboard suggests, there is an enormous amount of data, information, and knowledge assumed in the system. Intelligence data is assumed, for example, as is operationally-relevant data on terrain. But much more significantly, knowledge and "intelligence" is assumed. This is most evident during the course of action (COA) sequences where the system generates strawman hypotheses for the planner to consider. Our previous work in the area suggests that it is possible to develop knowledge bases capable of supporting course of action generation in limited (that is, "well-bounded") domains so long as access is gained to expert decision-makers and a large amount of codified doctrine exists.

G2 intelligence data is also available in tactical problem-solving and can be fed into a system designed to support operational decision-making. There are requirements for terrain data that can also be satisfied and, via some innovative programming concepts, represented interactively to decision-makers.

5.1.2 Analytical Methods Selection - It is clear that a hybrid approach is called for here. Tools and techniques from artificial intelligence and decision analysis (notably, utility theory) should be married to "drive" the system. Tactical planning elements can be treated as "objects" and processing can occur on the objects according to some heuristics about the meaningfulness of certain combinations of data (such as terrain and unit types, doctrine and enemy disposition, and the like).

The interdisciplinary approach to the design and development of the evolutionary prototype is clearly required -- and likely to yield the most cost-effective methodological solution.

Classic knowledge engineering will have to be performed to permit the system to generate strawman courses of action. Data about terrain, adversary disposition, and the like will have to be collected for display purposes, and it will be necessary to integrate various knowledge and data bases to make the system function. All of these tasks are well within the state-of-the-art of intelligent decision support systems design and development (Hopple, 1988).

5.1.3 Software Engineering Implications - Can we do all that is represented and demonstrated in the storyboard? Can it all be programmed? Can it be programmed cost-effectively? The answers to all of these questions is yes, though some reservations apply. First, with today's systems capabilities, the analytical graphics can be developed. Animation can also be programmed cost-effectively (though just five short years ago, it could not). While the throwaway storyboard prototype did not use interactive color graphic routines (because the target system was a Macintosh Plus/SE), it would be desirable to upgrade to a color system, such as the Macintosh II, which supports cost-effective color graphics (and animation).

The programming itself, in spite of the knowledge base requirements, need not be exotic. The call here is thus not for a LISP-based system, but rather one in a more modular and transportable language like C or Ada. There is also the distinct possibility of developing the evolutionary prototype in a 4th generation language (4GL) such as Hypercard.

The hardware and software issue is of course difficult to solve completely here. The point of "sizing" as it pertains to systems engineering is to determine if the system can be designed and developed cost-effecitvely -- or even at all! There is nothing in the throwaway prototype to suggest that it cannot cost-effectively transition to a working prototype.

5.2 Subject Matter Expert and Human Factors Engineering Feedback

In addition to the sizing presented above we also subjected the storyboard to experts in tactical planning and human factors engineering. These experts -- currently working on the related AirLand Battle Management program for the Defense Advanced

Research Projects Agency (DARPA) and the Army -- reviewed the storyboard to determine the extent to which it satisfied substantive planning requirements and interface requirements.

The experts were given copies of the two requirements hierarchies used to design the storyboard. They evaluated the hierarchies first to determine their suitability independent of the storyboard and then to determine the extent the stroyboard satisfied the criteria in the hierarchies. Their comments are as follows.

The first set of comments are positive while the second negative. The positive comments center around the interface itself. First, the experts felt that the interface structure was supportive of cognitive problem-solving, especially planning. They felt that removal of the keyboard was key to transfer success, since most users are inexperienced with computer technology of any kind.

They also believed that a logical extension of the interface would be touch input.

They felt that the use of split screens to alter persepctive was powerful and pertinent to the tactical planning process.

They felt that the use of case-based reasoning and analogical problem-solving was extremely vital to the planning process, since doctrine and experience play such important roles in the tactical planning process.

The ability to overlay multiple concepts of operations, courses of action, and the like, was considered fundamentally important to the planning process.

Of particular interest were the displays that communicated the Red and Blue (military) organizational structures, since it was revealed to us that planners do not always optimally deploy thier forces because they don't always know what forces they have! This was a requirement that was thus satisfied, even though we had not intended to satisfy it because we did not know it existed!

They liked the ability to "challenge" system logic very much. They felt that planners, while able to operate well via a system-generated "strawman," would require the capability to ask the system questions and pose problems and challenges. The ability to compare critical factors in terms of the extent to which they contribute to the strawmen plans they felt was particularly important.

Finally, they felt that one of the unmentioned benefits of the interface (and implied system) was its ability to generate graphic explanations of recommended courses of action. It was felt that when COAs are briefed to Commanders they ask precisely the kinds of questions that can be answered by the system. In other words, they felt that the system concept served all phases of the planning process, not just the COA generation/selection phase.

They also had some negative comments. First and foremost, they felt that the echelon (Corps) absolutely requires the capability to "zoom" over and around the terrain. We had considered a zoom option and rejected it, believing that planners could create a zoom effect by requesting explanations about terrain and key characteristics, and the like. Our evaluators felt that the system should give planners the capability to not only zoom in or out, but fly horizontally as well. They recommended that an interface option be added that permits planners to zoom around, fly in or out, and traverse terrain pretty much as they see fit. While this capability will complicate our system concept from a software engineering perspective, it will certainly add functionality to the system.

The also felt that some serious thought be given to the allocation of tasks between the human user and the computer. They felt that by and large the allocation was correct, but that aspects of the planning process could be enhanced if the allocation was based upon some structured analysis of what humans and computers do best.

They also felt that the explanation capabilities as reflected in the storyboard were inadequate. They called for more extensive detail and the capability for planners to query the system for long explanations of how COAs are generated.

They also felt that the system should permit easier "what-if" analyses. While the system concept permits planners to play "what-if" games by challenging (and re-challenging) the system, they felt that an easier approach should be designed.

Overall, expert comments were favorable, though several important changes should probably be made based on the insights. We were happy to learn that the qualitative evaluation suggested that the interface was indeed consistent with substantive requirements and compatible with the cognitive requirements of planners. However, while we agree with the comments and suspect that they are "correct," more quantitative tests should be conducted. More specifically (and ideally), a formal "requirements conference" should be held with expert planners where they would exercise the storyboard (while we collected data about their usage patterns, comments, insights, and criticisms).

Their comments also hold great implications for "sizing." If we were to implement all of their suggestions the system would be substantially more difficult to implement, especially given the weight they give to deeper explanations and additional display (i.e., zoom) capabilities. Hence, the need for more tests to determine where the greatest leverage lies (again, before programming begins).

6.0 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

This project has covered a lot of ground. We began with the hypothesis that a great deal of improvement could be made in the way we interface operational users to substantive algorithms. We looked at a specific domain -- tactical planning -- and developed two requirements hierarchies, one for substantive tasks and one for user-computer interaction (UCI) requirements. We then converted the requirements into a system concept manifest in an interactive "storyboard" prototype of how an actual system might operate in the field with "real" (in this case, G3 Operations) personnel. We then evaluated the prototype internally and via outside expertise in tactical planning and human factors engineering, especially as it pertains to the cognitive issues surrounding computer-based problem-solving.

We learned a number of things. First, we learned that the way we have historically computed analytically has disqualified a large number of prospective users from the revolution in "analytical computing." We have failed to design and develop interfaces that real users -- with little or no training in computer science or decision support systems design and development -- would find easy to use. We have been driven primarily by the requirements of our analytical methods and not by the substantive requirements that our systems must satisfy if they are to be accepted by operational personnel. This project attempted to break away from the "spreadsheet" mentality and inertia and design an interface concept that is anchored in substantive (planning) requirements and the UCI requirements unique to Army personnel.

We benefited from a great deal of research that we have conducted in tactical planning and UCI over the years. Because of our extensive requirements experience in Army tactical planning we were able to hit the ground running on this project and concentrate on how real planners plan and upon how a support system must help them generate and evaluate options. We developed a new UCI requirements hierarchy and then translated its elements into a system concept that (a) does not require a keyboard, (b) is intuitively obvious, is (c) inherently graphic and (d) -- most of all -- goal-oriented.

Is it possible to build such a system? Can we move from the storyboard prototype to an actual "evolutionary" prototype? Yes. Would the cost be prohibitive? No. It is possible to design and develop a system like the one demonstrated in the prototype for a similar domain -- like Air Defense -- on off-the-shelf hardware (like the Apple Macintosh II) and off-the-shelf and original software.

Appendix A describes an architecture used to design and develop two prototypes for tactical plan generation and evaluation (at the Army Corps level). The prototypes perform many of the same analytical functions that the Phase II system would have to perform, functions such as terrain analysis, capabilities assessment, and course of action generation and evaluation. Our previous work in the area has exploited the marriage of several analytical methods classes (like artificial intelligence and decision analysis); it has also been based upon extensive requirements analysis and knowledge engineering conducted in conjunction with real tactical planners. (Appendix B describes the process in more detail.) Neither of these aids, however, was conceived with anywhere near the kind of emphasis on UCI that has directed this project. We believe that based upon the progress made for the Human Engineering Laboratory in the UCI area and based upon our previous decision-aiding work for the Army we are now in a position to design and develop a complete system for an Army aviation/air defense domain.

Such a system can be designed and developed within the parameters of SBIR Phase II efforts. It is anticipated that two years would be required to build the system, though working "evolutionary" prototypes would begin to appear much earlier in the process. Our tested systems design methodology, as suggested again in Figure 6.1, would drive the project. Note that demonstration prototypes appear as early as step two in the process. It would be necessary for the domain to be specified, for a scenario to be selected and validated, and to conduct extensive requirements analysis with real tacticians. This would lead to the design of a system concept and the selection and implementation of a powerful hybrid analytical methodology to drive the Phase II system concept, and so on down the design blueprint until the system was completed, documented and "transferred" for operational evaluation.

REFERENCES

Andriole, S.J. (1986a) The Design & Development of an Intelligent Planning Aid: The TACPLAN Prototype. Marshall, VA: International Information Systems, Inc.

__________ (1986b) The INTACVAL Storyboard. Marshall, VA: International Information Systems, Inc.

__________ (1988) Storyboard Prototyping for Systems Design: A New Approach to Requirements Validation & System Sizing. Wellesley, MA: QED Information Sciences, Inc.

__________ and G. W. Hopple (1987) Inference-Making and Option Selection in Army Corps-Level Tactical Planning: Some Hybrid Models and a System Concept Storyboard. Marshall, VA: International Information Systems, Inc.

Army War College. (1984, 1985) The Letort Scenario. Carlisle, PA: Army War College.

Bain, W. (1986) "Judge: A Case-Based Reasoning System." In Machine Learning. Norwell, MA: Kluwer Publishers.

Benbasat, I. (1984) "An Analysis of Research Methodologies," pp. 47-85 in The Information Systems Research Challenge: Proceedings. Boston, MA: Harvard Business School Press.

Benbasat, I., A. S. Dexter, and P. Todd (1986) "An Experimental Program Investigating Color-Enhanced and Graphical Information Presentation: An Integration of the Findings." Communications of the ACM, 29 (November): 1094-1105.

Bennett, J. (1977) "User-Oriented Graphics, Systems for Decision Support in Unstructured Tasks," pp. 3-11 in S. Treu (ed.), User-Oriented Design of Interactive Graphics Systems. New York: Association for Computing Machinery.

Boar, B. (1984) Applications Prototyping: A Requirements Definition Strategy for the 1980s. New York, NY: Wiley-Interscience.

Carroll, J. M. and Mack, R. (1985) "Metaphor, Computing Systems and Active Learning." International Journal of Man-Machine Studies. 22.

Clarke, A. A. (1986) "A Three-Level Human-Computer Interface Model." International Journal of Man-Machine Studies, 24: 503-17.

Dent, C., Klein, G. A. and Eggleston, R. (1987) Metaphoric Displays for Dynamic Tasks. Yellow Springs, OH: Klein Associates, Inc.

Dumas, J. S. (1988) Designing User Interfaces for Software Systems. Englewood Cliffs, NJ: Prentice-Hall.

Edgell, S. and Castellan, N. (1986) "Relevant Information is Worse than Irrelevant Information." Presented at the Psychonomic Society Meeting, New Orleans, LA, October.

Edwards, W. and Newman, J. R. (1982) Multiattribute Evaluation. Beverly Hills, CA: Sage Publications.

Ellis, S. R. and McGreevy, M. W. (1987) "Perspective Traffic Display Format and Airline Pilot Traffic Avoidance." Human Factors. 29.

Foley, J. D. and A. Van Dam (1982) Fundamentals of Interactive Computer Graphics. Reading, MA: Addison-Wesley.

Gaines, B. R. and M. L. G. Shaw (1986a) "Foundations of Dialog Engineering: The Development of Human-Computer Interaction. Part II." International Journal of Man-Machine Studies, 24: 101-23.

Gaines, B. R. and M. L. G. Shaw (1986b) "From Timesharing to the Sixth Generation: The Development of Human-Computer Interaction. Part II." Internationaal Journal of Man-Machine Studies, 24: 1-27.

Galitz, W. O. (1984) Humanizing Office Automation. Wellesley, MA: QED Information Sciences, Inc.

__________ (1983) Handbook of Screen Format Design. Wellesley, MA: QED Information Sciences, Inc.

Hirst, W., Spelke, E. S., Reaves, C. C., Caharack, G. and Neisser, U. (1980) "Dividing Attention Without Alternation or Automaticity." Journal of Experimental Psychology. 1.

Hopple, G. W. (1988) The State of the Art in Decision Support Systems. Wellesley, MA: QED Information Sciences.

__________ (1986) Alternative Defensive Plan Generation and Evaluation: Object-Attribute-Value Knowledge Base. Marshall, VA: International Information Systems, Inc.

Ishikawa, H., Y. Izumida, T. Yoshino, and A. Makinouchi (1987) "KID: Designing a Knowledge-Based Natural Language Interface." IEEE Expert, 2 (Summer): 57-71.

Kolodner, J. L., Simpson, R. (1986) "Using Experience as a Guide for Problem-Solving." In Machine Learning. Norwell, MA: Kluwer Publishers.

Kolodner, J. L., Simpson, R. and Sycara-Cyranski, K. (1985) "A Process Model of Case-Based Reasoning in Problem-Solving." Proceedings of the IJCAI-85.

Kolodner, J. L. and Reisbeck, C. K., eds. (1986) Experience, Memory and Reasoning. New York, NY: Lawrence Erlbaum.

Lakoff, G. and Johnson, M. (1980) "Hypotheticals As Heuristic Device." Proceedings of the AAAI-86.

Landee-Thompson, S. (1986) User-Computer Interface Technology: An Assessment of the Current State-of-the-Art. Marshall, VA: International Information Systems.

Maass, S. (1983) "Why Systems Transparency?," pp. 19-28 in T. R. Green, S. J. Payne, and G. C. van der Veer (eds.), The Psychology of Computer Use. New York: Academic Press.

McCracken, D. L. and R. M. Akscyn (1984) "Experience with the ZOG Human-Computer Interface System." International Journal of Man-Machine Studies, 21: 293-310.

MacLean, R. F. (1986) "User-System Interface Requirements for Decision Support Systems," pp. 99-126 in S. J. Andriole (ed.), Microcomputer Decision Support Systems: Design, Implementation, and Evaluation. Wellesley, MA: QED Information Systems.

Morgan, S. L., J. J. Kindon, and A. Nauda (1986) "A Man-Machine Interface Simulator," pp. 199-202 in D. Gantz, G. Blais, and S. Solomon (eds.), Proceedings of the 1985 Winter Simulation Conference. San Diego, CA: Society for Computer Simulation.

Preece, J. (1983) "Graphs Are Not Straightforward," pp. 41-56 in T. R. G. Green, S. J. Payne, and G. C. van der Veer (eds.), The Psychology of Computer Use. New York: Academic Press.

Schank, R. and Abelson, R. (1977) Scripts, Plans and Understanding: An Inquiry into Human Knowledge Structures. New York, NY: Lawrence Erlbaum.

Senach, B. (1983) "Computer-Aided Problem Solving with Graphical Display of Information," pp. 57-68 in T. R. G. Green, S. J. Payne, and G. C. van der Veer (eds.), The Pyschology of Computer Use. New York: Academic Press.

Shneiderman, B. (1987) Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, MA: Addison-Wesley.

__________ (1980) Software Psychology: Human Factors in Computer and Information Systems. Cambridge, MA: Winthrop Publishers, Inc.

Simmons, R. F. (1986) "Man-Machine Interfaces: Can They Guess What You Want?" IEEE Expert, 1 (Spring): 86-94.

Smith, S. L. and Mosier, J. (1984) User-System Interface Design for Computer-Based Information Systems. Bedford, MA: The Mitre Corporation.

Verbrugge, R. (1980) "Transformation in Knowing: A Realist View of Metaphor." In R. Honeck and R. Hoffman, eds., Cognition and Figurative Language. Hillsdale, NJ: Lawrence Erlbaum.

________ and McCarrell, N. (1977) "Metaphoric Comprehension: Studies in Reminding and Resembling." Cognitive Psychology. 9.

APPENDIX A

"Intelligent Aids for Tactical Planning"

by

Stephen J. Andriole, Harlan H. Black, Gerald W. Hopple
and John R. Thompson

Intelligent Aids for Tactical Planning

STEPHEN J. ANDRIOLE, MEMBER, IEEE, HARLAN H. BLACK,
GERALD W. HOPPLE, MEMBER, IEEE, AND JOHN R. THOMPSON, MEMBER, IEEE

Abstract—Two interactive decision aids developed to support tactical planners at the Corps level are considered. The article presents the systems design process used to design and develop the aids as well as a description of the specific steps taken to conduct requirements analysis, develop functional "storyboards," and program the aids. Special emphasis is placed upon the use of expert planning judgment to validate requirements and "size" the aids. Both aids are configured around an interactive video disk-based display system linked to an IBM PC / AT.

Introduction

TACTICAL planning is a perennial defense process. Commanders at all echelons must plan optimally if they are to survive. The processes by which commanders plan, however, have not yet been influenced by the revolutions in analytical methodology or microcomputing. This article looks at two aids designed to support Corps Commanders in the generation and evaluation of alternative plans or *concepts of operation*.

Tactical Corps planning is characterized by a great deal of uncertainty about adversary intentions, likely "Red" courses of action, and possible responses. The challenge we faced several years ago involved the design of aids that would amplify expert judgment in a way that would reduce battlefield uncertainty. TACPLAN and INTACVAL, the two planning aids discussed here, evolved over time in response to a steadily growing body of literature that we collected and data that we extracted from subject matter experts. This data enabled us to validate requirements iteratively, and also permitted us to develop strawmen "storyboards" (screen displays of the aids before they were programmed) that our planners could work with directly.

This article describes the steps that were taken to develop TACPLAN and INTACVAL as well as the aids themselves.

Manuscript received May 20, 1986; July, 1, 1986. This work was supported in part by the Army Research Institute and the U.S. Army's Communications-Electronics Command, and under subcontract (to International Information Systems, Inc.) from Perceptronics, Inc.

S. J. Andriole is with the Department of Information Systems and Systems Engineering, George Mason University, Fairfax, VA 22030, USA.

H. H. Black is with the Computer Research Division, CENTACS, US Army Communications Electronics Command (CECOM), Fort Monmouth, NJ 07703, USA.

G. W. Hopple is with International Informations Systems, Inc., 802 Woodward Road, Marshall, VA 22115, USA.

J. R. Thompson is with SAIC, 505 Marquette Avenue, NW, Albuquerque, NM 87102, USA.

IEEE Log Number 8610737.

Systems Design Methodology

We approached the planning support problem with the systems design methodology that appears in Fig. 1. We assumed from the outset that, since the domain was primarily cognitive, iteration would be necessary; hence the rapid prototyping methodology depicted in Fig. 1.

The methodology called for a detailed and systematic requirements analysis, requirements validation (in this case, via storyboards), and requirements/analytical methods matching—all before programming began.

The methodology permitted us to take small steps toward the development of the aids. In fact, the methodology itself led us to INTACVAL, the aid that evolved directly from TACPLAN (TACPLAN was conceived in 1983; INTACVAL in 1985).

Planning Requirements Analysis

Fig. 2 presents the tactical planning process. Note the distinctions among staff and commander actions. Note also the planning *process* depicted in the figure. We began with an assessment of the elements and steps in the planning process and, for analytical and bureaucratic reasons, focused on the Corps planning process. Corps planning is more abstract than lower-level planning though just as goal-directed and hierarchical. The systems design methodology applied to our Corps planning problem could just as well be applied to any echelon.

Several experiments were conducted at the Army War College in Carlisle, PA, to identify the processes by which Corps plans are generated and evaluated. We used several War College scenarios to identify requirements, which we gathered via video-taped exercises of actual planners solving a planning problem. As the planners formulated and evaluated alternative plans, we requested that they "think aloud" the protocols used to assess terrain, capabilities, and courses of action. We studied the tapes and, in conjunction with codified planning doctrine, developed a functional concept for TACPLAN (first) and (then) INTACVAL.

Figs. 3–6 describe the planning experiments and present the results. Two groups of planners participated. One group—students from the War College—comprised three Lieutenant Colonels (05s), while the other, three Colonels (06s who were also planning instructors at the College). They were given a scenario and asked to formulate a tactical defensive plan given a likelihood of a Warsaw Pact

0018-9472/86/1100-0854$01.00 ©1986 IEEE

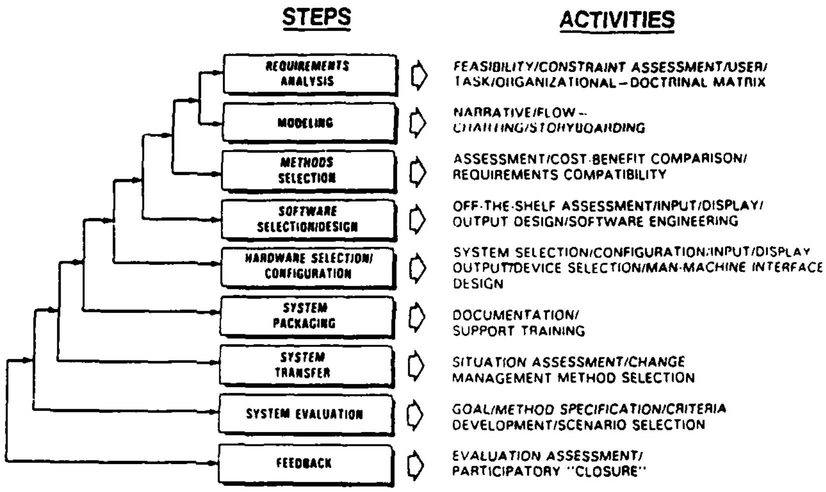


Fig. 1. Structured systems design methodology.

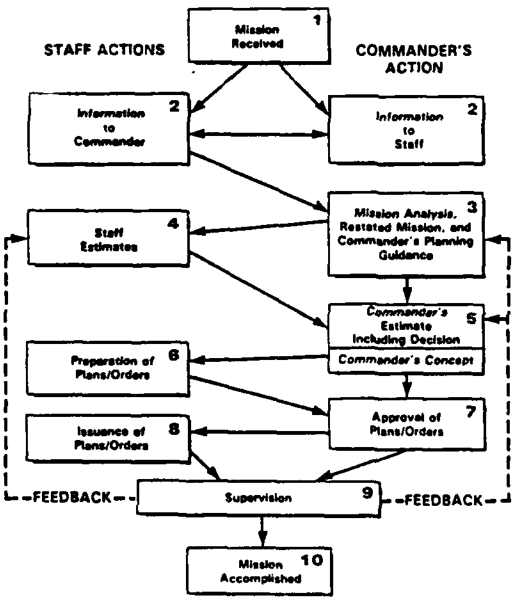


Fig. 2. Military planning process.

- **The Purpose:**
  - *To gain insight into the MILITARY planning process; to develop a model of military planning*
- **The Participants:**
  - *3 (06) expert tactical planners from Army War College's (AWC) Land Warfare group; 3 (05) planners from the AWC student bo...*
- **The Scenario:**
  - *Imminent red (Soviet) invasion of western Europe*
- **The Mission:**
  - *Develop a tactical plan for the defense of western Europe; deploy blue (Allied) forces optimally*
- **The Method:**
  - *Filming/recording of planners during the planning process; analysis of data; conversion of data into planning protocols; conversion of protocols into functional model of the military planning process*

Fig. 3. Carlisle planning experiments.

invasion of Western Europe (as the map in Fig. 4 suggests). The scenario of course was fictitious in content, though permitted us to observe the planning process in great detail.

The results of the experiments appear in Figs. 5 and 6. We compared the differences between the groups as well as their similarities. We discovered, for example, that regardless of the planner or the group, that planning was inherently graphic and nonnumeric, that without their "props"—maps, acetate overlays, and grease pencils—they simply could not plan. This finding had great implications for the subsequent design of the aids (see below).

The experiments also validated the planning process that we had discovered via other codified means, such as the one that appears in Fig. 7 [3]. TACPLAN and INTACVAL both adhere to the process represented in Fig. 7.

Functional Modeling Via Storyboarding

Following the requirements analysis, we modeled the planning process (procedures, functions, and tasks) in a "storyboard" depicting how the aids might function in operation. Storyboards are nothing more than screen displays of the functions and tasks that the aid might perform when activated by a user. They are, however, extremely powerful vehicles to requirements validation, system "sizing," and man–machine interface development. We developed a number of storyboards for TACPLAN and INTACVAL [2] and presented them to our prospective users for comment and criticism. TACPLAN went through two reviews while INTACVAL six. (TACPLAN was in concept and operation a much simpler aid than INTACVAL.)

The storyboarding exercise enabled us to validate requirements, identify some totally new ones, experiment with alternative man–machine interface (MMI) techniques, and—most importantly—select the analytical methods most likely to help drive the aid. After an analysis of the range of (computer science, decision analytic, operations research, and artificial intelligence) methods, we

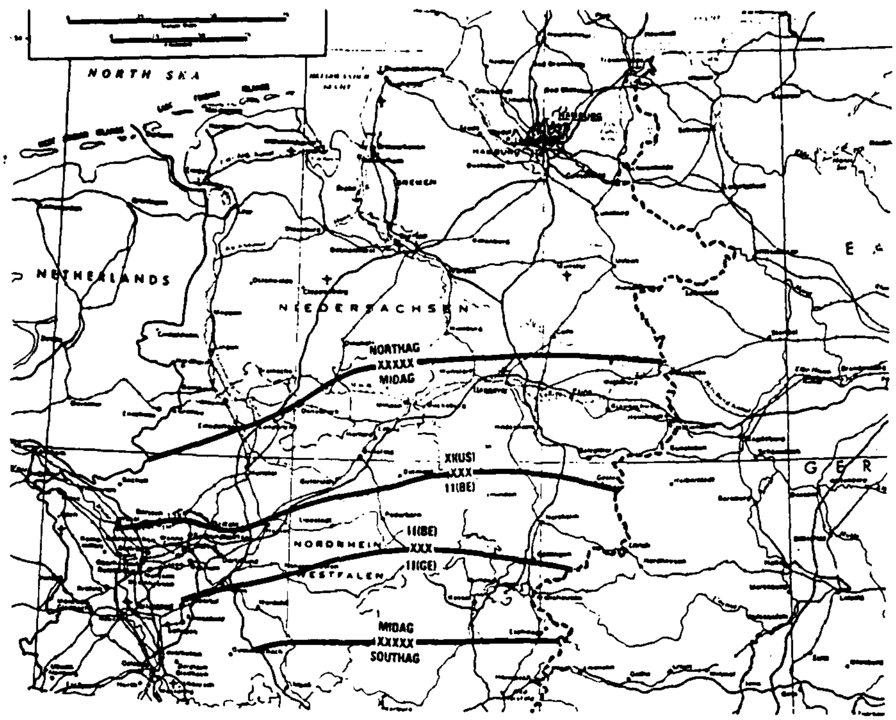


Fig. 4. Senario map. (MIDAG denotes Corps. Command.)

• THE MILITARY PLANNING PROCESS	• MILITARY VS. NON-MILITARY PLANNING (MP/NMP)
• **Highly structured** • *Sequential* • *Procedural* • *Doctrinal* • *Integrated . . .* • **Highly repetitive** • *Planning, re-planning, & contingency planning* • **Mission-oriented** • **Verbal, graphic, non-numeric process**	• **Planning guidance originates from above in MP, but from within in NMP** • **MP is usually adversarial; NMP is usually not** • **MP is accompanied by high accountability, while NMP is usually not** • **MP is explicitly goal-directed, while NMP is frequently opinion-driven** • **MP planning receives near-immediate, highly structured feedback, while NM — especially personal — planning is often stimulated from directly within the planner** • **MP is by nature & definition distributed, while NMP is frequently localized with one or two planners**

Fig. 5. Results of the Carlisle experiments: military planning process.

- **Group #1 (06 expert planners) considerably more risk averse than Group #2 (05 student planners); Group #2 expressed greater certainty re adversary intentions**
- **Group #1 more deliberate (i.e., more options, more dimensions of value) than Group #2**
- **Group #1 expressed considerably LESS confidence in their solution (plan) than Group #2**
- **Group #1 institutionalized "Devil's Advocate" role, while Group #2 avoided structured challenges to solutions to various parts of the planning problem**
- **Group #1 relatively more devoted to military (Army) doctrine than Group #2, perhaps as much a reflection of familiarity with doctrine as devotion to it**

Fig. 6. Results of the Carlisle experiments: Group 1 vs. Group 2.

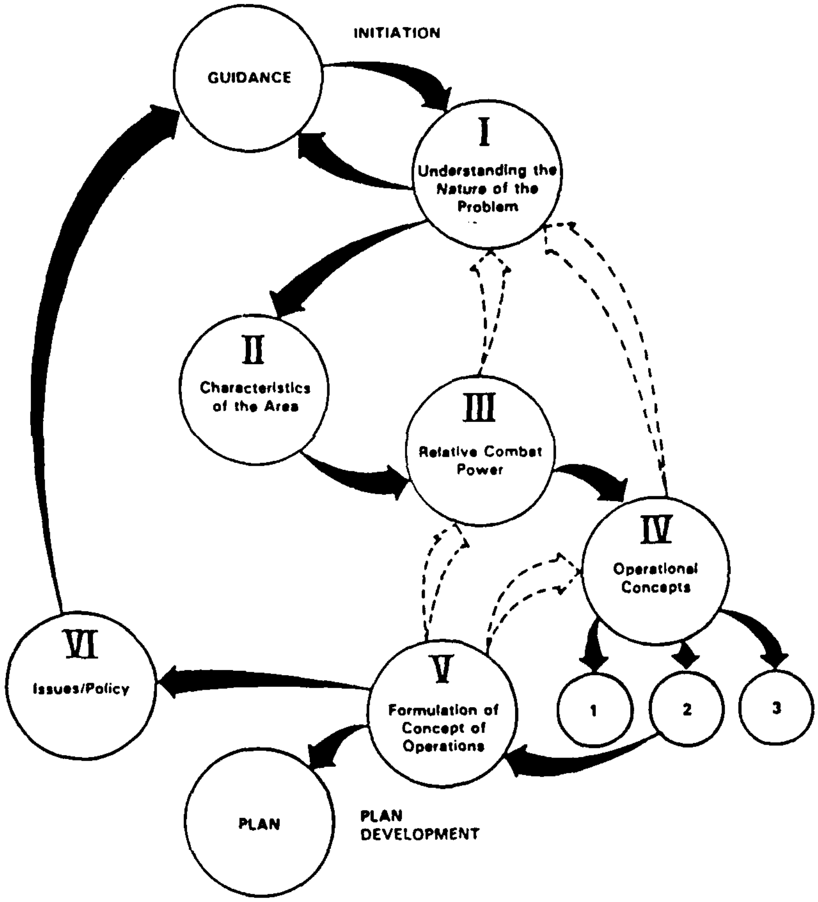


Fig. 7. Iterative planning process.

initially opted for a hybrid decision analytic/artificial intelligence methodology for TACPLAN. As the research progressed, however, we moved more toward the artificial intelligence side of the hybrid [2].

Functional and Systems Architecture: TACPLAN

TACPLAN was conceived after the first round of experiments at the Army War College and after an initial study of planning.

We relied as heavily upon planning doctrine (at least as far as we had gone with it at the time) as upon expert planner judgments (as captured in the experiments at the Army War College). The Carlisle experiments, refined analytical framework, TACPLAN functional description, and TACPLAN concept of operation all determined the TACPLAN *delivery system*. There were also a number of overarching technical and applications guidelines that determined the configuration. Compatibility across these two data bases yielded the delivery configuration described below.

Personal Planning Aid: A common technology need heard over and over again these days calls for the design and development of low-cost, portable, and distributed personal decision aids. All of these terms are of course relative. Low-cost generally means less expensive than a minicomputer. Portable usually means *transportable*, and distributed usually means *accessible* and sometimes *networked*. However, *personal* has some interesting connotations beyond the literal derivatives. Personal should mean controllable and manageable. It should mean nonthreatening and supportive; and it should mean flexible and adaptive. The planning process is certainly compatible with these definitions.

Our observation of the planning process suggests the appropriateness of personal aiding. There are also no computational or display requirements that cannot be satisfied with a personal aid. Low-cost, portable, distributed, and personal thus suggested to us the use of a microcomputer with interactive graphic display capabilities. We selected the IBM Personal Computer (PC) with ample storage and memory capabilities (IBM/PC/XT). The TACPLAN prototype is easily implemented on this microcomputer (though INTACVAL did indeed require the capabilities of an IBM/PC/AT, which has larger memory and storage capabilities). This system gives us the option of color or black and white (monochrome) displays as well as several input options. The prototype permits keyboard and joystick/mouse input; it has also been configured to interact naturally and synergistically with the interactive graphic interface.

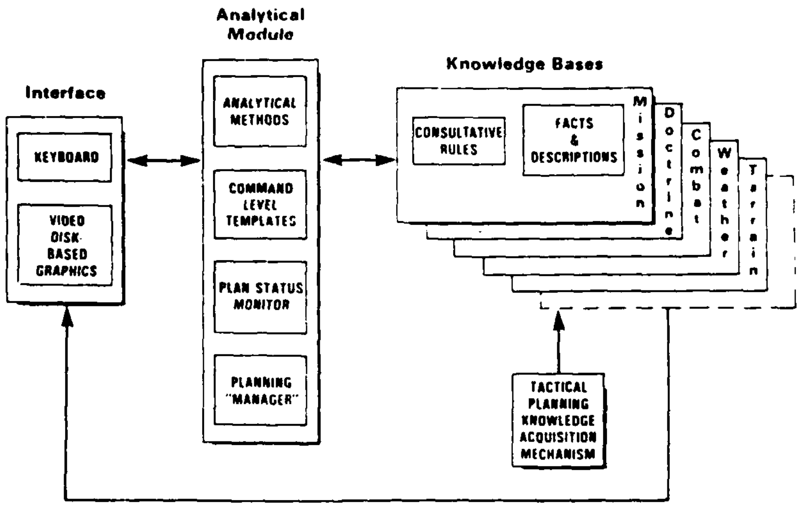


Fig. 8. TACPLAN.

The Graphics Interface: As Fig. 8 suggests, TACPLAN has a graphics interface comprised of an interactive video disk, video disk player, and video disk/mapping control panel. Video disk-based maps are made by first filming the area that is to become the disk-based map. The team has already filmed the areas necessary to implement the TACPLAN prototype. The NATO/Pact areas of Western and Eastern Europe are already on disk at multiple scales.

The video disk displays actual maps of the Corps area of responsibility at near perfect resolution. In fact, the video disk-based map is clearer and much more flexible than any conventional paper map. It permits planners to annotate what appears on the display (which is located right along side the IBM/PC/XT display) with nodes and symbols for later reference. It also enables planners to create their own personal symbols for annotation purposes. It supports decluttering, a simple operation that removes (stores or erases) annotations.

The video disk-based mapping system also permits planners to "fly" around a map, which, if laid out, might cover a warehouse floor. Under joystick control, planners can fly across great spans of maps, switch off to other kinds of data (photographs, films, weapons descriptions), and even zoom in on an image or location of particular interest.

The most important capability of the video disk-based mapping system, which has been linked to the (IBM/PC resident) TACPLAN analytical system, is its ability to communicate symbolically with both the planner and the analytical side of TACPLAN. When a planner illustrates an enemy course of action by drawing it on the video display, TACPLAN immediately "knows" something about this course of action. It may know, for example, about the terrain that might constrain or propel the course of action, it may know about the relationship between the course of action and force structure, and it may know about the relationships among doctrine, orders of battle, and trafficability. Coordinates on the video disk-based map are linked to analytical routines and knowledge bases stored in TACPLAN. When planners illustrate courses of action, annotate on the disk image, or erase an idea, TACPLAN knows what has happened and reacts accordingly. The nature, timeliness, and depth of the reaction is determined by the analytical routines and knowledge bases resident in TACPLAN not by the capabilities of the interactive video disk.

In time it should be possible for a planner to draw a complete route across a map, or deploy several divisions along a border, and expect TACPLAN to react immediately, consistently, and substantively about the implications of the route or plan (see the sections below on the capabilities of INTACVAL). Once the technology that links locations on the video disk to analytical routines and knowledge bases is fully developed, then the possibilities are unlimited. Procedural and substantive rules can be invoked spatially. Alternative plans can be generated once a planner has entered his first candidate, and narrative criticism and advice can be offered—all from what the planner does via TACPLAN's graphic interface.

TACPLAN Knowledge Bases: TACPLAN's knowledge bases are primarily oriented to the relationship between terrain type, maneuverability, and unit type. If a certain kind of unit tries to cross a specific kind of terrain TACPLAN "knows" whether or not the route makes sense, that is, whether a planner has driven an armored unit into a forest. When a relationship between unit and terrain type is identified, TACPLAN either remains quiet or warns the planner about a rule violation.

The idea is to use the rules as constraints not prescriptive planning tools. Plan formulation remains in the hands of the expert planner, not in the system's software. The concept behind the TACPLAN aid was to marry elements in decision analysis with powerful AI representation techniques. In operation TACPLAN permits planners to make wholesale judgments about area characteristics, mission, and doctrine and then subjects these judgments to what is contained in the knowledge bases.

The knowledge bases themselves were developed using a scaling technique for maneuverability and terrain. If certain classes of units are expected to move along certain terrain lines then certain maneuverability constraints will restrict such movement. Units may be lightly, moderately, or highly constrained by terrain types. A window appears on the screen of TACPLAN when a rule is violated and the outcome calculated. In operation, then, the planner would see a list of constrained units as well as their (light/medium/heavy) score. He would then be permitted to change his judgments or assumptions about the planning situation.

The important aspect of this capability is the direct link between the graphic interface and the knowledge bases, which can be triggered by drawing a line (with the joystick/cursor) directly onto the video-disk image as a designated hypothetical course of action. TACPLAN then calculates the location of the units designated as part of the course of action and then determines how constrained they are by the terrain. All of this is done instantly without the planner's intervention. In other words, all the planner has to do to implement the knowledge bases is interact with the video disk-based display.

Another important aspect of the knowledge base is the way its output is displayed to the planner. Instead of complicated formal rule structures (such as IF–THEN), TACPLAN displays a simple explanation of the problem, such as "you have just moved armored units into a forest and they are highly constrained; your mission is in jeopardy." This subtle change in the way rules and rule violations are displayed to users (from the way conventional "expert systems" display rules) is critical to the success of TACPLAN-like systems. Note that our intended users are not experienced systems users nor are they methodologically sophisticated. They therefore need (and have a right to expect) systems to support the way they actually do business. A simple change in the way rules and rule violations are displayed to users will improve performance immeasurably.

TACPLAN Analytical Modules: TACPLAN permits the evaluation of alternative plans (concepts of operation) via several embedded multi-attribute utility routines that are (like the knowledge bases) linked directly to the graphic display system. Planners are simply asked to make judgments about the elements of tactical planning: the mission, area characteristics, blue and red capabilities, and red and blue courses of action. These judgments are not quantitative-empirical, that is, the planner does not have to come up with scores for each element of tactical planning. Rather gross judgments are required to a) feed the embedded MAU routines and b) fire the constraint knowledge bases. The aid is decidedly interactive and works with the planner to fine tune judgments and hypotheses. The aid very much assumes the value of an intelligent assistant and behaves accordingly.

System Configuration: Fig. 8 suggests how TACPLAN is configured and also suggests the components of its architecture. One of the ways we were able to satisfy the low-cost portable constraint was to first develop the knowledge bases symbolically and then convert them to numeric form so that we could implement them on the PC/XT. We then developed conversion files for rule and rule violation explanation purposes. This approach not only supported our goals for smooth man-machine interaction, but it also permitted us to use conventional programming techniques instead of alternative recursive/search techniques. Our primary motivation for this approach was not to avoid symbolic processing, but to make certain that the system would run on the least expensive configuration possible.

We were pleased with the mileage we were able to get from off-the-shelf software and software utilities. The Lattice C windowing system, for example, served our purposes very well. We were also able to exploit existing Perceptronics video disk control software. The TACPLAN project demonstrated that it is indeed possible to productively utilize off-the-shelf software and thereby cut design, development, and especially programming costs dramatically.

A TACPLAN Session: TACPLAN walks through the planning process by asking the planner a series of questions about his planning problem. It first asks him to describe his mission. It then asks him how much time he has to solve the planning problem as well as the problem's geographic location.

It then asks the planner to describe the area characteristics in terms of advantages and disadvantages to the planner. He is required to decide if friendly or adversary forces have advantages in aspects of terrain by simply checking blocks that ask: "favor NATO?/favor PACT?" The embedded MAU routines test these judgments against what is stored in the knowledge bases and if no violation occurs remains silent. If, however, a violation does occur, then a window appears to the planner that describes the violation.

The planner has to assess relative positions for area characteristics, combat capabilities, and courses of action. As the planner makes these judgments the knowledge bases are checked for violations. More importantly, the course of action assessment is done completely on the video disk display, where the planner actually draws hypothetical courses of action and then waits for TACPLAN to respond. These COA's can be labeled and stored for later comparison. They can be overlayed onto one another for highlighting and comparative purposes. The ability to write directly (via a digital overlay onto the video disk-based image) onto the graphic display permits the planner to recreate many of the same capabilities s/he is so familiar with via acetate overlays and grease pencils.

After the course of action is selected TACPLAN provides a summary of what the COA and overall concept of operation assumes about the situation. As is true throughout the TACPLAN-aided planning process, the planner is free to disagree with the system's judgments.

TACPLAN is very much a simple assistant in the planning process. Via some rules about the relationships among unit type, terrain, and maneuverability, it advises and constrains planners about what they can and perhaps should not do. It places a great deal of input burden on the

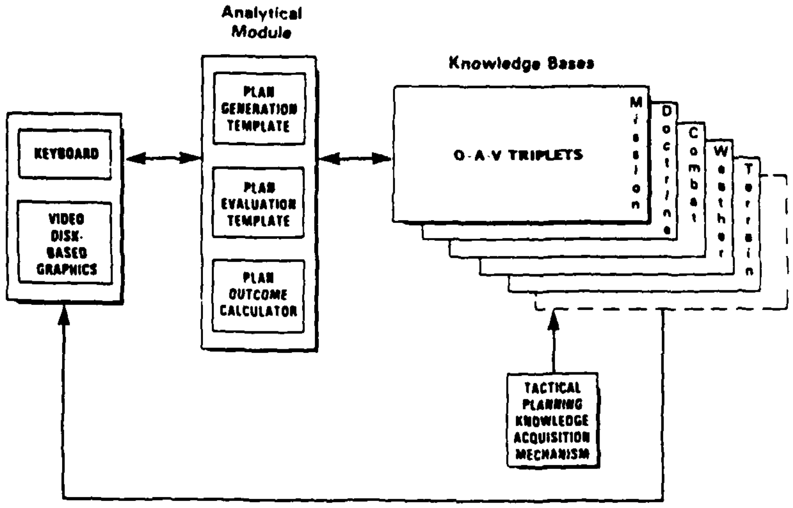


Fig. 9. INTACVAL.

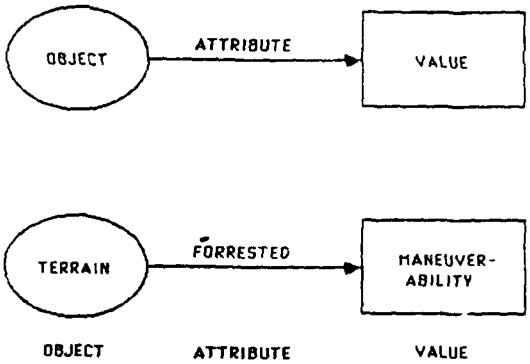


Fig. 10. Basic object–attribute–value triplet.

- *OBJECTS* ARE PHYSICAL ENTITIES OR CONCEPTUAL ENTITIES (e.g., "UNIT" OR "PLAN")
- *ATTRIBUTES* ARE GENERAL CHARACTERISTICS OR PROPERTIES OF OBJECTS (e.g., "ARMORED" OR "READY")
- *VALUES* ARE NATURES OF ATTRIBUTES (e.g., "MANEUVERABILITY")
- CAN BE USED TO DIRECT INFERENCES IN RULE/NETWORK STRUCTURES.

	OBJECT	ATTRIBUTE	VALUE
F	Terrain	Forrested	Low-Maneuver
	———	———	———
	———	———	———
	———	———	———
THE	UNIT	ARMORED	CONSTRAINED

Fig. 11. O-A-V's explained.

planner, requiring him to make many (nonnumeric) assessments about opportunities and constraints.

At the same time, TACPLAN successfully demonstrates the utility of "graphic equivalence," and the benefit to be gained from a flexible graphic representation of the planning environment. It also demonstrates the utility of hybrid analytical methods. INTACVAL, on the other hand, expects less from the planner and tries to make the leap from "assistant" to "associate."

Functional and Systems Architecture: INTACVAL

INTACVAL was conceived after TACPLAN was tested in prototype form with actual planners. We discovered that the "aiding paradigm"—or the role that the aid should play—had not been well enough specified; hence INTACVAL. The aid itself was built upon TACPLAN, but is fundamentally different in operation and function.

Functional Components: INTACVAL, as Fig. 9 suggests, comprises a planning interface, knowledge bases, plan formulation and evaluation templates, and a plan outcome, or battle, calculator. Note that INTACVAL is an interactive aid, not an automated expert system. The planner is intended to be an integral part of the aiding system.

Knowledge Bases: INTACVAL's knowledge bases are not comprised essentially of rules (like TACPLAN's). In fact, the content and depth of its knowledge bases are determined by the depth identified within the overall object–attribute–value structure that we have adopted [4].

Objects–attributes–values, or O-A-V's, enable a great deal of information about planning to be encapsulated in a flexible, yet formal, structure. As Figs. 10 and 11 suggest, O-A-V's represent a technique for knowledge representation that is a derivative of larger frame- and network-based approaches.

In the planning domain, O-A-V's comprise planning objects that correspond to the familiar elements of tactical planning. These objects have attributes which in turn have values (see the figures). It is thus possible to identify a set of objects, attributes, and values that pertain to all of the elements of tactical planning and then look for patterns among the values to determine which, in effect, "go" together. Various combinations of values relate to specific concepts of operations that in turn can be used to guide

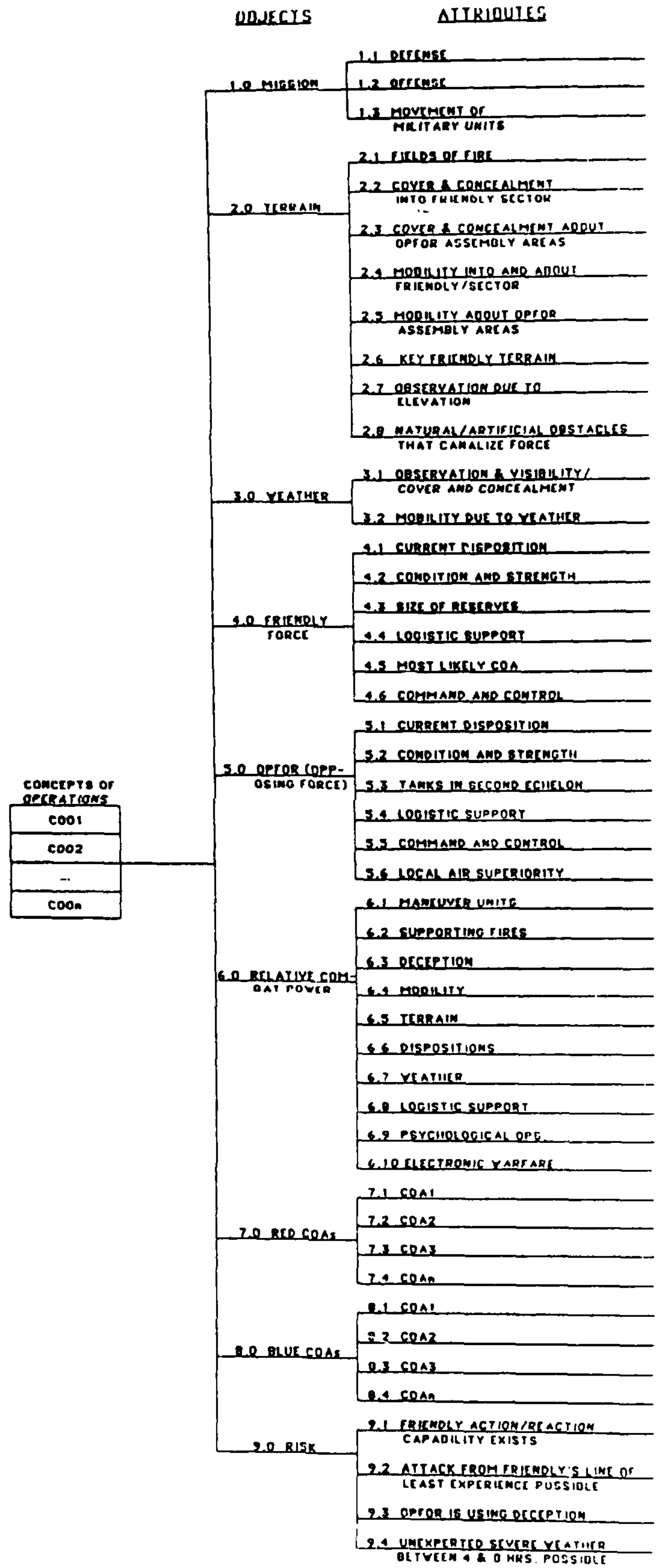


Fig. 12. Objects and attributes from INTACVAL knowledge base

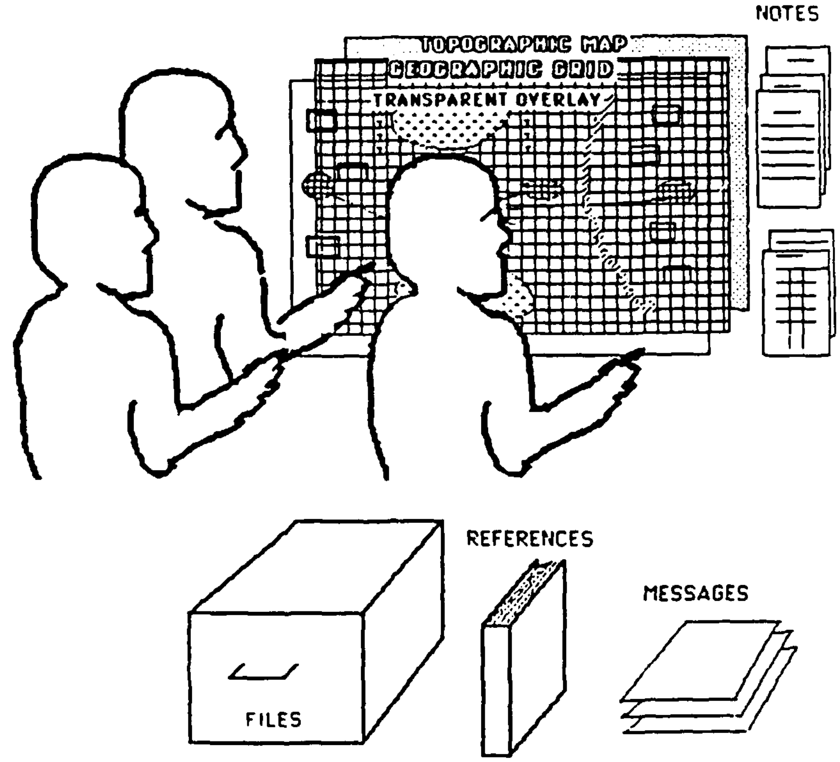


Fig. 13. Manual planning process.

the planning process, check planner judgments, and even generate strawman concepts of operation.

O-A-V's lie at the heart of the INTACVAL knowledge base. They are used to write rules about value combinations, to display planning logic to planners, and update what is learned about tactical planning over time. A large O-A-V triplet structure has been built by International Information Systems, Inc. (with help from expert planners and doctrine). A list of the objects and attributes in the structure appear in Fig. 12. The values, along with definitions, all appear in [4].

The Tactical Battle Calculator: INTACVAL's knowledge bases will support the development of alternative plans and their evaluation. The aid permits a planner to formulate a concept of operations and then activate a tactical battle calculator that will play out the plan vis-a-vis the terrain, the force structure, the mission, logistics, and, of course, the adversary. This battle calculator is model- *and* knowledge-based. It is model-based to the extent that there is a set of battle patterns resident in the battle model; it is knowledge-based to the extent that the model accesses INTACVAL's knowledge bases in order to determine the plan's most likely outcome.

INTACVAL displays the processes by which alternative plans are executed both graphically and alphanumerically. The graphic display consists of digital overlays onto a video disk-based map image, while the alphanumeric display presents INTACVAL's reasoning. The planner is thus able to "see" how the plan does and then receive guidance about why it is likely to succeed or fail (see the section below on how INTACVAL actually behaves in operation).

The battle calculator does not "decide" which plan is best and which plan is worst; instead, it "checks" the plans to see if they violate any of the constraints and prescriptions in the knowledge bases. It does not numerically rank-order the alternative plans, though it will suggest how the plans compare with one another along consistent logic/reasoning paths. For example, when a plan is "run" through the battle calculator it will be assessed according to what INTACVAL knows about tactical planning. When several plans are run through the calculator INTACVAL will permit the comparison of the plans according to how well or badly they each did in all of the appropriate categories, e.g., terrain, unit types, relative, and absolute combat capabilities, and doctrine. It is thus possible for the planner to make meaningful comparisons across the alternatives. Some techniques for facilitating comparison include listings of "strong" and "weak" plan components, side-by-side comparisons of plan features, and the capability to respond to queries about individual or several plans.

Man–Machine Interface: The primary interface device is the video disk-based map display. Through this display the planner is able to build tactical plans (by actually moving unit symbols across a map), label them, overlay them upon each other, make notations about their features, and "see" the plans executed in real-time animation.

Use of the video disk-based map display may be justified by several experiments conducted over the years regarding the use of spatial/graphic versus alphanumeric information processing and our experience with the TACPLAN prototype [1]. Some tasks are inherently spatial, graphic, and nonnumeric, while others are more natu-

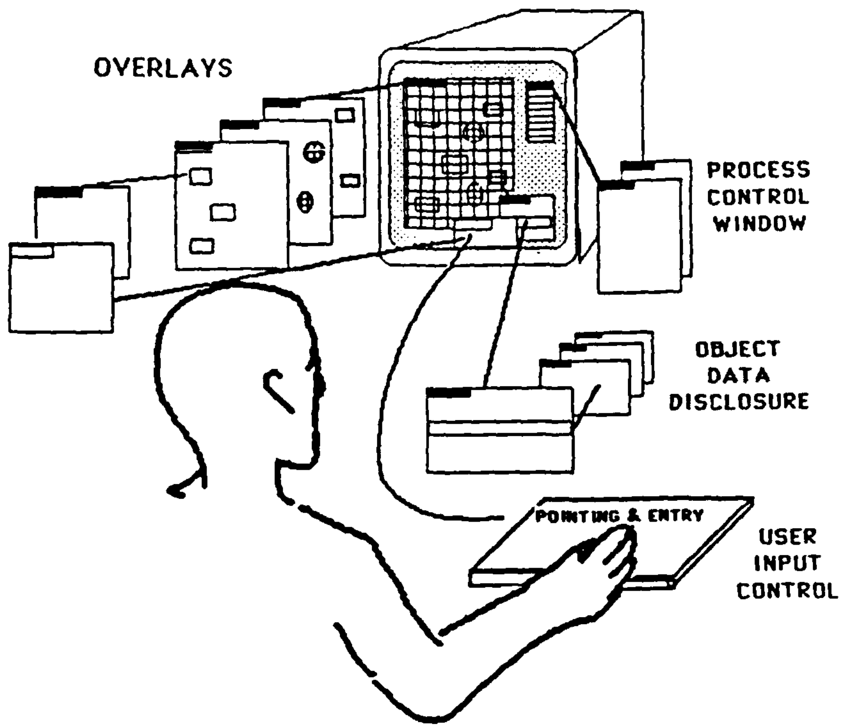


Fig. 14. TACPLAN- and INTACVAL-aided planning process.

rally solved via mathematical or quantitative means. Our research suggests that tactical planning is inherently graphic and non-numeric. When planners plan, they move icons, refer to illustrations, draw courses of action, and argue via references to pictures, graphs, and lines. They do not perform mathematical calculations or convert their judgments into numeric form; hence the use of an interface device capable of supporting nearly all aspects of the non-numeric problem-solving process.

The issue of video disk-based versus computer-generated imagery (CGI) displays is also worth noting. In many applications, CGI displays are appropriate, especially when one-to-one realism is unnecessary. However, in those applications where it is important to present near-perfectly real information, then the video disk alternative makes sense. What about tactical planning? Without question, tactical planners prefer the actual map image to a computer-generated one. Real planners are trained on real maps, understand real maps, and solve problems efficiently with real maps. Perhaps more importantly, INTACVAL's video disk-based interface actually improves upon conventional (paper) map displays by permitting the planner to scroll around the map, zoom in or out for different scale views of the area, and separate out the (terrain, rivers, cities) features of the map for "decluttered" viewing and problem-solving. Since INTACVAL's interface permits digital annotation of images, lines, and text directly onto the map image, planners are able to annotate alternative courses of action and store them for comparison.

There is also the issue of cost-benefit. Unless the application specifically calls for—or can tolerate—CGI-based maps, there is every reason to opt for the video disk-based map simply because of the enormous cost of CGI maps relative to their video disk counterpart. CGI systems are also expensive to maintain and field.

The arguments about flexibility are also worth noting. In the not too distant past, video disks were criticized for their lack of flexibility, but since the establishment of production centers throughout the United States the criticism is no longer valid. Video disks can now be produced in a matter of days instead of the months that were previously required.

Finally, the age of read/write video disks is all but upon us. Within a few years it will be possible to add or delete images from the same disks. When this technology matures, the last functional shortcoming of video disk-based technology will disappear.

INTACVAL—like TACPLAN—has dual screens. One display presents the map(s), while the other presents the alphanumeric aspect of the planning process (e.g., displays of reasoning); actual development of alternative courses of action will, however, all be done via the interactive video disk.

Actual manipulation of the map and related images is via a trackball integrated into INTACVAL's keyboard. Nonvideo disk input is via a conventional keyboard with graphic and nongraphic menus organized around the planning process.

System Configuration: INTACVAL is configured much like TACPLAN, except it resides on an IBM PC/AT with enhanced graphic display capabilities. It also has a different menu structure from TACPLAN's, relying on a pull-down/windowing type.

INTACVAL was programmed in C (like TACPLAN) and makes heavy use of interactive graphics to display the components of the knowledge base, avenues of approach, and the like. INTACVAL also has a modified animation capability.

An INTACVAL Session: INTACVAL was designed to support Corps Commanders and their G-3 (Operations) staff. G-2 (Intelligence) inputs to the process are simulated in INTACVAL. The commander and his staff thus receive information about area/terrain, adversary disposition and strength, and the like and the planner is expected to process it into a concept of operation, or plan.

When all of the planning "data" has been received, the aid generates a series of strawmen options for the planner to consider. Note that this represents a radical departure from the aiding paradigm in TACPLAN. The planner is then free to query INTACVAL about why it recommended the strawmen that it did. The O-A-V triplets become very important in this iterative process. Recall that the triplets are used to determine the relationships among specific courses of action and values in the structure. After receiving all of the planning data, INTACVAL searches for pattern matches among the values until it finds one. It then recommends the course of action that its rules (firing on the O-A-V's) tell it is valid. The planner then can ask INTACVAL to display the values that led to the plan's generation and acceptance. If he does not like the values in the pattern, he can change them immediately to what new rule might be triggered by the change. This on-line knowledge base interaction capability links the planner directly with the essence of the aid, and supports knowledge-based iteration on strawman candidate plans.

As the candidates are displayed, the planner can also ask INTACVAL to play out its implementation quickly. This modified war-gaming capability gives the planner another look at the plan; INTACVAL also calculates a kind of balance sheet for how well (or badly) the plan did vis-a-vis the prescribed mission.

The planner's primary function in the INTACVAL aiding process is to iterate on strawmen generated by INTACVAL, to query INTACVAL about how and why it recommended what it did, and to play out candidates in real-time for further inspection. In spite of the system-generated solutions, INTACVAL is not an expert system. It is intended to amplify planning expertise, not to replace it. It is a constraint checker and an "associate"; thus it performs like TACPLAN but is far more intelligent. Final judgments about the acceptability of plans lie with the planner.

Conclusion

INTACVAL and TACPLAN represent attempts to aid a complicated analytical process. They are both prototypes, far from operational application. At the same time, they successfully demonstrate the integration of advanced methodology and microcomputing for support purposes. They also suggest that the systems design methodology used—especially with its emphasis on requirements definition and functional modeling via storyboarding—can be expected to yield continued dividends over time.

The objective of our work can be seen graphically in the final two figures. We are clearly trying to move from "manual" planning to computer-aided planning—without disturbing or distorting the process itself. While we may have taken some small steps in that direction via TACPLAN and INTACVAL, the real work lies ahead.

References

[1] S. J. Andriole, *The Design and Development of an Intelligent Planning Aid: The TACPLAN Prototype.* Marshall, VA: International Information Systems, Inc., 1986.

[2] ——, *INTACVAL Storyboard (Version IV).* Marshall, VA: International Information Systems, Inc., 1986.

[3] Army War College, *Sound Military Decision-Making.* Carlisle, PA: Army War College, 1984.

[4] G. W. Hopple, *Alternative Defensive Plan Generation and Evaluation: Object-Attribute-Value Knowledge Base.* Marshall, VA: International Information Systems, Inc., 1986.

Stephen J. Andriole (M'81), for a photograph and biography please see p. 764 of this issue.

Harlan H. Black was born in 1948. He received the B.A. degree in Mathematics from Loyola College, Baltimore, MD, the M.S. degree in computer science from Atlanta University, Atlanta, GA, and the M.S. degree in education from the Johns Hopkins University, Baltimore, MD.

He is with the US Army Communications Electronics Command (CECOM), COM/ADP Center where he conducts research on man-machine interfaces. As a member of the CECOM/COM/ADP's Computer Research Division he has conducted research in tactical symbology, adaptive interfaces, and C3 graphics. As a former member of the CECOM Ada Division he conducted research on Ada development and maintenance and developed the first audio-visual demonstration of the Ada Language System.

Gerald W. Hopple (M'86), for a photograph and biography, please see p. 964 of this issue.

John R. Thompson (S'61–M'67) received the B.S. degree in electrical engineering from the University of New Mexico, Albuquerque in 1964 and the M.S. degree in engineering from the University of California, Los Angeles, in 1971.

He was formerly a Group Director in the Artificial Intelligence and Man-Machine Systems Division of Perceptronics, Inc. He is now with SAIC. His technical research interests focus on prototyping activities, innovative user interface design, decision aiding, and the application of AI to both general and specific military problems. He has been an invited speaker on the topic of man–machine interface at numerous conferences and workshops, including AFCEA's "Artificial Intelligence and National Defense: Applications to Command and Control Beyond" and "Decision Aids for Command and Control".

Mr. Thompson is a member of the Air Force Association, American Defense Preparedness Association, Association of the US Army, and the Armed Forces Communications and Electronics Association.

APPENDIX B

"TACPLAN -- An Intelligent Aid for Tactical Planning"

by

Stephen J. Andriole

TACPLAN

-An Intelligent Aid for Tactical Planning

by Stephen J. Andriole

Not long ago, Gen. Edward C. Meyer, former Army chief of staff, stated that, "The most demanding challenge confronting the U.S. military, in the decade of the 1980s, is to develop and demonstrate the capability to successfully meet threats to vital U.S. interests." Meyer's marching orders required the Army to reaffirm its commitment to operational and force planning and to formalize the processes by which plans are developed, assessed and implemented.

Around the same time, organizations such as the Defense Advanced Research Projects Agency (DARPA), the Army Research Institute (ARI) and the Army War College (AWC) began to respond to some technology challenges issued by the office of the Under Secretary for Defense Research and Engineering (USDRE). Without much fanfare, Dr. William Perry challenged his R&D organization to computerize—where appropriate—as many defense processes as possible. Among the specific technology targets were cruise missiles, command and control (C2) and strategic and tactical planning (TACPLAN).

TACPLAN is an offspring of the marriage arranged by Meyer and Perry. It is a microcomputer-based, "intelligent" planning aid capable of supporting planning at the corps level.

Corps planning, like the planning that occurs at all command levels, is mission-directed, top-down and structured. Corps commanders and their staffs must take a number of steps before developing a concept of operations. They must first convert mission guidance into sets of goals and subgoals that are clear and realistic. They must "prepare" the battlefield by integrating intelligence about weather, terrain and enemy objectives and capabilities. They must identify likely enemy courses of action and then determine the best way to deploy, attack or defend (figure 1).

There are reams of "how to" planning documents available to the corps commander.[1] But the essence of successful planning can be traced to the quality of available intelligence, the capabilities of opposing forces and the judgement of the commander.

Judgement and experience drive the tactical planning process. In many important respects, tactical planning is an art. Aspects of the planning process cannot always be taught unless the planner has had actual field experience. At the same time, this is not to suggest that parts of the process cannot be computerized via the implementation of some "scientific" procedures.

Good corps planning is iterative. Successful corps commanders are simultaneously creative and pragmatic. They are also good choreographers, required to balance the priorities of command against the realities of limited resources, imperfect intelligence and a formidable opponent.

The planning process could not be implemented without certain "props," such as clear maps, clear acetate, tactical symbols and grease pencils. In spite of frequent criticism about the use of "low technology," historically, it has been very difficult to improve upon—even with "high technology" fixes.

Attempts to develop computerized planning aids began about two decades ago, but significant progress has only been made during the past five to 10 years.[2] But even the recent aids suffer from a common problem—the forcefitting of specific analytical methods onto the planning process regardless of whether they were appropriate. Some planning aids, for example, use a theorem of conditional probabilities to calculate likely enemy courses of action. While the theorem is powerful, it forces the planner to consider likelihoods in ways that require him to ignore what he has been taught about tactical planning.

Other problems can be traced to the kind of interaction that many aids impose upon the planning process. One aid, for example, requires corps commanders to input hundreds of numeric scores to calculate the "best" concept of operations.[3]

Still others failed because they were simply too large to support anything but tactical training. Many were also too expensive to enjoy widespread distribution.

The challenge we faced a few years ago required us to determine if corps planning *should* be computerized. Could computers help, or would they add yet another layer of complexity upon a process already well understood and executed? Next, we faced the challenge of identifying the points in the planning process where an aid could make the best contribution. Were there enough to justify the investment our sponsor, ARI, planned to make? What about methods? Was the inventory large enough to perform critical tactical planning tasks? What about cost and portability?

The ultimate challenge, of course, was simple. Could we move significantly beyond acetate and grease?

TACPLAN was developed by International Information Systems, Inc., and Perceptronics, Inc., with support from ARI. The first step involved the conduct of a series of requirement analyses designed to determine exactly how corps commanders formulate tactical plans. In January 1984, we videotaped six expert planners as they formulated a concept of operations for a defensive Western European scenario (see box on the birth of an analytical aid). The planners, from the Army War

College's Center for Land Warfare, were divided into two groups and asked to develop a solution to the same tactical planning problem.

This data was supplemented with Field Manuals, officers' handbooks and general literature on planning. After we studied the data, we developed a taxonomy of planning tasks and sub-tasks and then identified the analytical methods likely to satisfy them.[4]

The first issue we faced required us to profile the role that we wanted the aid to play. Should the goal be to develop an aiding system or an automated planner?

Our requirements clearly suggested that the development of an automated planner was impossible, given the state-of-the-art of our technology—and undesirable, given what we were able to discover about tactical planning. We determined that the best way to proceed was to conceive of TACPLAN as an aid only and to require its behavior to be as unobtrusive as possible.

TACPLAN uses several conventional and unconventional methods to help corps commanders develop a tactical plan. First, it assumes that tactical planning is amenable to a divide-and-conquer strategy, where planning problems are broken down into sub-problems. It also assumes that the sub-problems should be solved in a specific order. It assumes, for example, that course-of-action assessments should not be made until area characteristics and relative combat capabilities have been thoroughly analyzed. It also suggests that planners define their primary and secondary goals before any analysis takes place. TACPLAN supports these assessments by a method known as multi-attribute, utility assessment (MAUA). MAUA is a technique that supports analysis by identifying, weighting and scoring criteria vis-a-vis courses of action, terrain features and other analytical considerations. The technique yields lists of the most likely courses, the most inhibiting terrain features, and the concept of operations most likely to satisfy the commander's goals.

TACPLAN also has "intelligence." It "knows" about terrain constraints, preferred doctrinal options, force structures and their interrelationships. Its knowledge is stored in planning rules that are consulted whenever a planner makes a judgment, designates a likely course of action, or makes an assumption about combat maneuver capabilities.

Remember that TACPLAN is an aid, not an automated expert system. It is intended to help planners, not replace them. Its *intelligence* is passive and unobtrusive. TACPLAN does not ask the planner what he wants to do and then do it for him; instead, the aid watches the planning process and only alerts the planner when something it knows about the process or about the problem at hand has been ignored or contradicted. It then alerts the planner to the problem. If the planner chooses to ignore TACPLAN's advice, then the process proceeds.

Figure 2 suggests how computer-aided tactical planning might occur. The planner manipulates data and information, tests alternative hypotheses about the best way to deploy his forces, and calls up different displays (*terrain, order of battle, etc.*) which appear to him as overlays on the computer screen. He can add or subtract overlays, while a set of rules about the tactical planning process observe his planning and alert him when a violation occurs.

The entire aid resides on an IBM PC/XT. The aiding interface is through the monochrome display and an interactive videodisc-based, graphic display system. This dual screen display system permits planners to interact alphanumerically and graphically within the context of the same problem within

The Military Planning Process

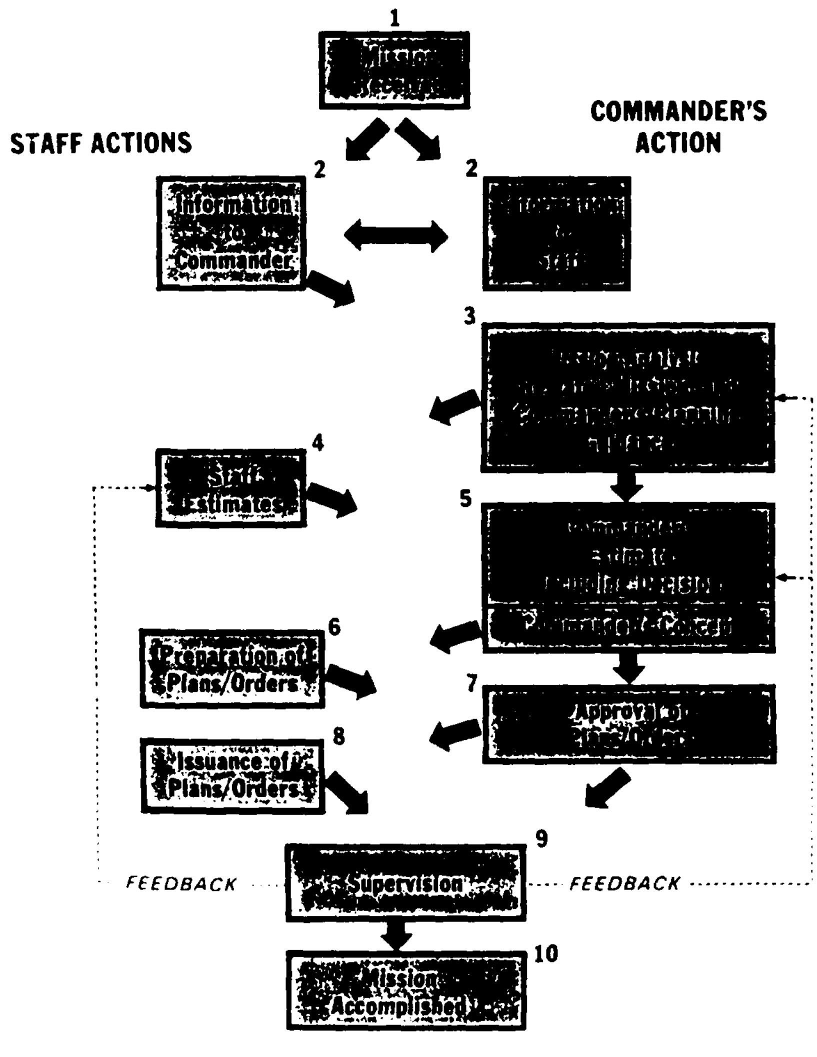


Figure 1

TACPLAN

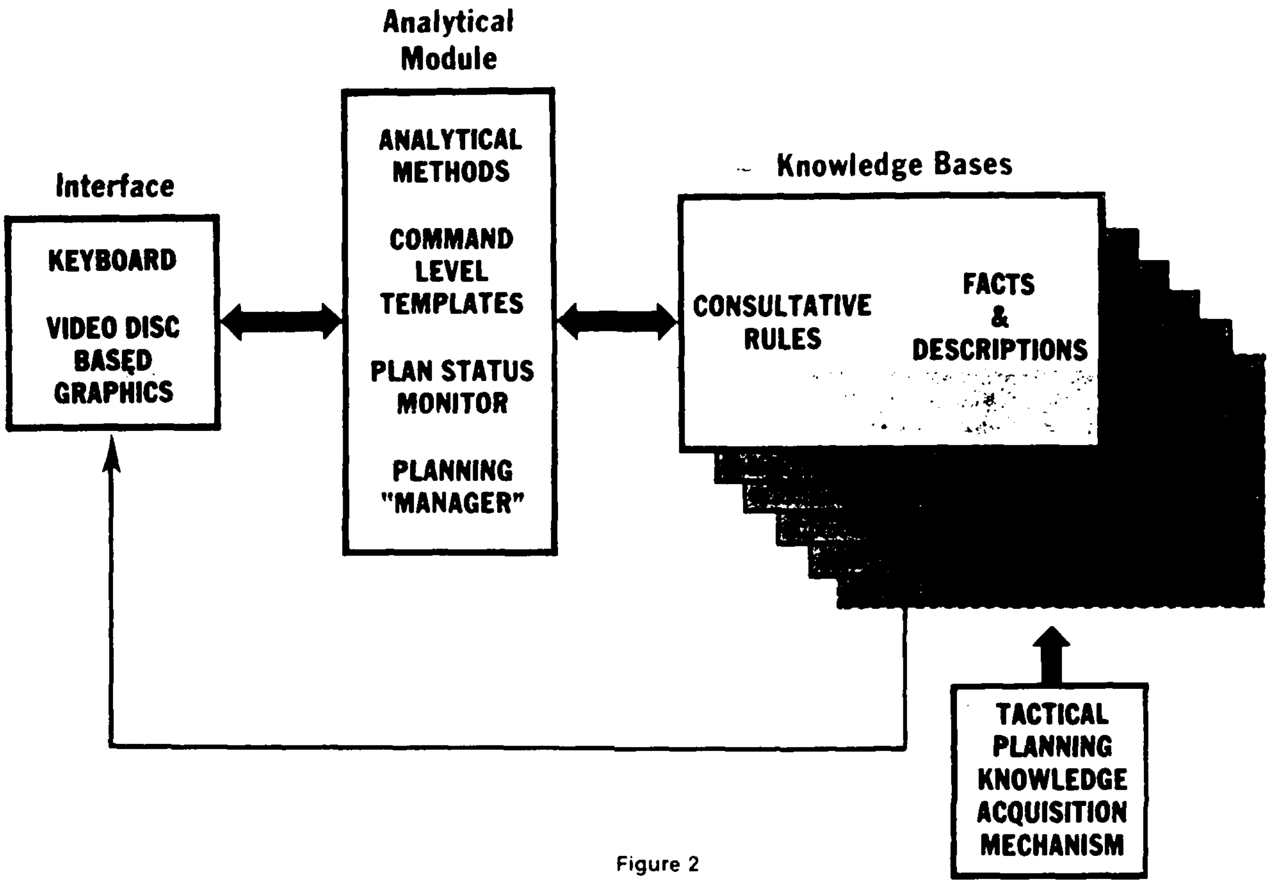


Figure 2

the same aid.

The monochrome display guides planners through the plan building process, asking a series of questions about the planning problem. Planners are asked to assess area characteristics, relative combat capabilities, enemy and friendly courses of action, and potential concepts of operations. While this is occurring, the video display is building the plan graphically. Planners can interact directly with the video display by adding, deleting or moving units around the map, by drawing courses of action directly onto the map, or by recalling and overlaying old plans onto a current problem.

The strength of the video display lies in its realism and the extent to which it improves upon the use of paper maps, clear acetate and grease pencils. TACPLAN's video display presents actual maps of corps areas of interest. Tactical symbols and courses of action are stored as overlays on the video map.

The maps are stored on videodiscs

The Birth of an

Before designing a computer-based aid of any kind it is necessary to conduct a series of requirements analyses to determine if the aid should be built and what tasks the aid can and should perform. TACPLAN R&D began with the identification of some expert planners willing to share their expertise. Six planners at the Army War College agreed in 1984, with the support of Gen. Healy, then commandant of the college, to participate in a two-day experiment. We used a War College scenario of an impending Warsaw Pact attack and asked the planners to form two groups and independently solve the defensive planning problem. Both sessions were prepared by setting up a four foot by eight foot tactical map of the situation, overlayed with acetate. Grease pencils were provided, and friendly and enemy forces were represented by magnetic symbols that could be moved as the simulation evolved.

The problem itself saw massive deployment of "Red" forces all along and to the east of the NATO/Pact boundary. "Blue" forces were relatively scattered and outnumbered.

The planners were videotaped as they formulated their concepts of operations. The suggestion to videotape was made by Dr. Robert M. Sasmor of ARI, who hypothesized that a great deal of information about planning could be captured not only from the conventional audiotaping of planning utterances but also from the observation of

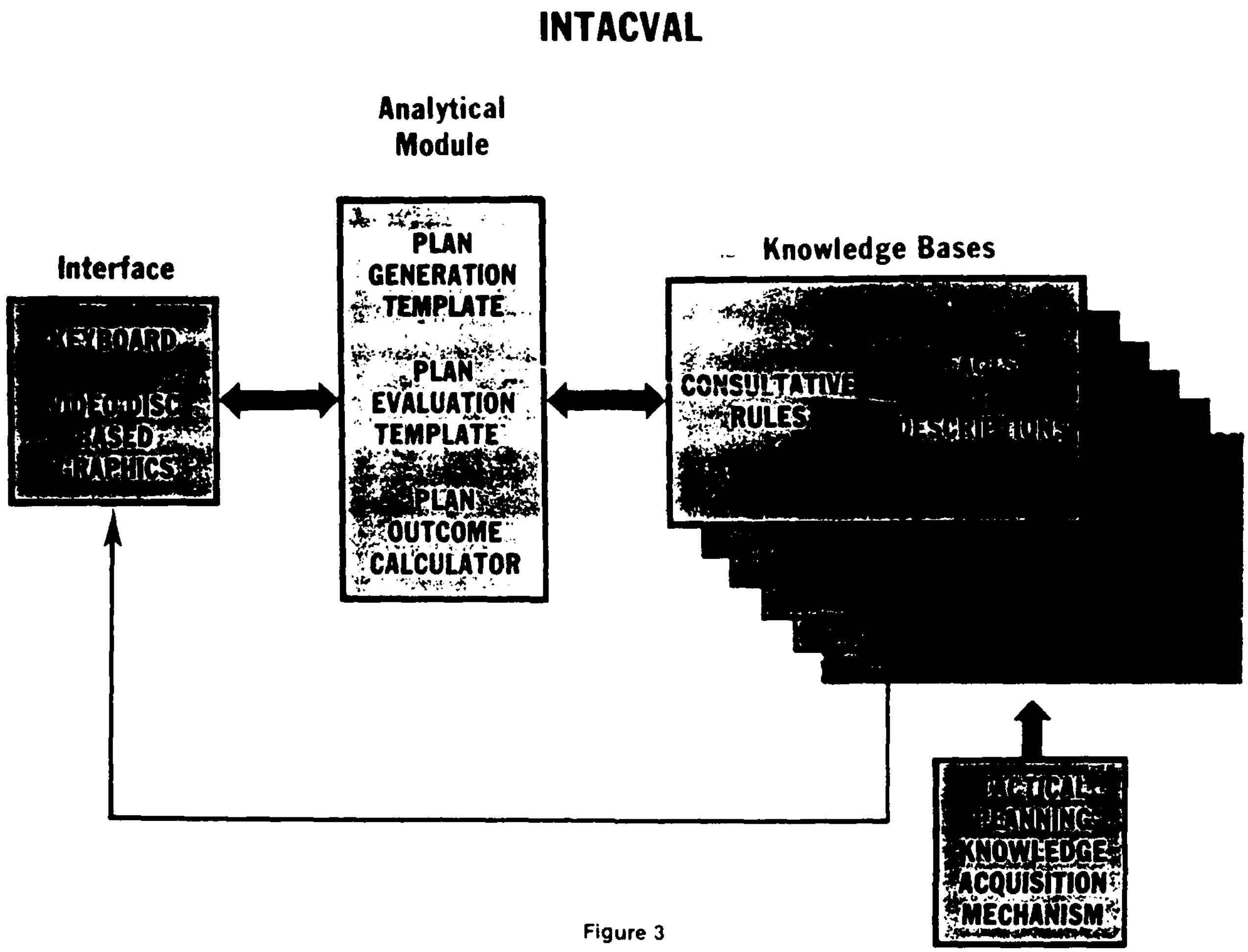


Figure 3

Analytical Aid

[illegible] suggestion [illegible] dividends. The video- [illegible] planning process [illegible] tactical planning [illegible] experiments, we were able [illegible] tactical planning, as [illegible] sequential, doctrinal [illegible]. We noted that tactical [illegible] repetitive, mission- [illegible] and perhaps most importantly, [illegible] verbal, graphic and non-nu- [illegible]

These and other observations enabled us to design an aid that was consistent with the actual planning process. Our original design did not call for the use of an interactive graphic display interface. After observing the actual planning process, it was clear that if the aid were to satisfy any real operational requirements, it would need a video window of the planning situation. The results also suggested that analytical methods would have to be embedded into the aid unobtrusively, and that any method that required planners to cognitively "break" from the substance of the planning process would undermine the aiding process. Finally, observation of the videotape enabled us to construct a taxonomy of planning tasks and sub-tasks that was subsequently used to design all the interaction sequences that now operate in TACPLAN.

while the annotations are stored digitally. Since maps have been placed on the videodiscs at different scales and fields of view, planners have the capability to zoom in or out on the display. Since the digital overlays are computer controlled, they expand or contract depending on the planners' field of view.

TACPLAN permits planners to develop tactical plans in much the same way that they now develop plans, but with some important differences. First, the aid structures the process. Second, it integrates elements and sub-elements of the process. Third, it monitors and instructs the process by checking planning judgments against its own knowledge base. Fourth, it permits planners to work with the same medium—actual maps—that they use when developing tactical plans without the aid of a computer. Fifth, it permits them to annotate onto the map image (not unlike the way they annotate on acetate with grease pencils), store their annotations and recall them at will (capabilities that acetate and grease pencils cannot match). Sixth, TACPLAN records the planning process for future study or application. In fact, it is possible to develop an inventory of corps plans that can be recalled and displayed whenever a similar problem is faced. Seventh, TACPLAN is flexible and adaptive. Over time, the rules that govern its behavior will be modified and the knowledge base will be expanded. If, for example, a particular piece of advice is consistently ignored by corps commanders, then the rule responsible for the advice can be changed. Since TACPLAN records the planning process it also records the disagreements, which can be used to make the aid more intelligent.

Finally, TACPLAN is inexpensive and transportable. Nearly all of its components can be purchased off-the-shelf. Special emphasis was placed on the design of an aid that would not consume huge resources in development or use. In fact, TACPLAN is in many respects a "rapid prototype" that can be modified easily, quickly and inexpensively.

TACPLAN is an aid that supports the development of tactical plans. But after *a battle begins or after events or conditions* change dramatically, commanders must revise their plans to some extent. We are currently in the process of developing an aid that will support plan evaluation and revision for the U.S. Army's Communications and Electronics Command's Center for Tactical Computer Systems' Computer Research Division (CECOM/CENTACS/CR). This aid—known as INTACVAL (Intelligent Tactical Plan Evaluation)—will be more intelligent than TACPLAN since it will respond to a variety of contingencies, help the planner decide whether to revise or completely rewrite his plan, and support the revision/replanning process. The aid must also be capable of displaying alternative plan outcomes on the video display, a capability which will require the aid to conduct "fast time simulations" in response to alternative repairs or new plans[5] (figures 3 and 4).

The INTACVAL architecture is flexible and supports the integration of the three aiding functions (plan formulation, plan evaluation and replanning) into a single system with the same dual screen display interface. The goal is to develop a computer-based "plan processor" that will support corps plan formulation, evaluation, execution, monitoring, revision and replanning. ★

THE TACPLAN & INTACVAL-Aided Planning Process

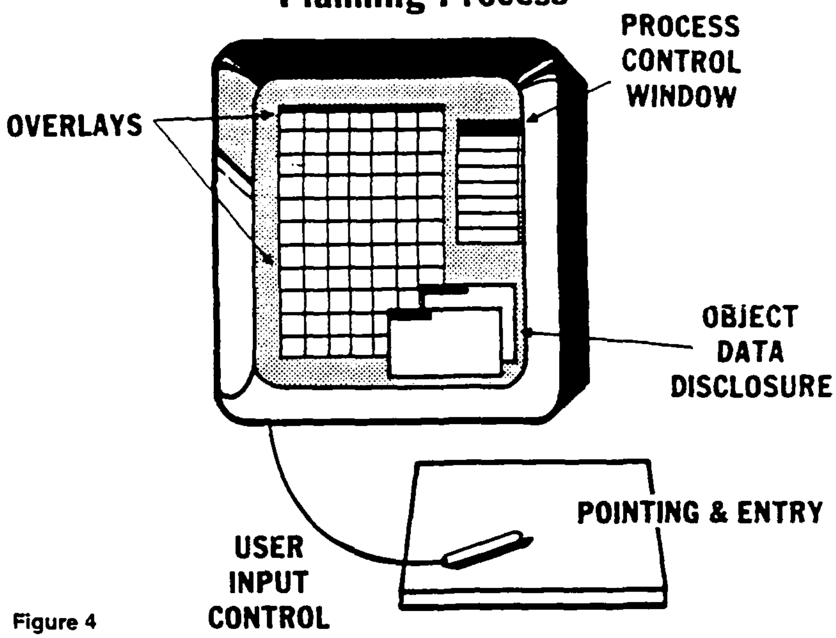


Figure 4

Footnotes

1. **Sound Military Decision Making**, Carlisle Barracks, Carlisle, Penn.: Army War College, 1983.
2. Stephen J. Andriole, ed., **Coursebook on Decision Aids for Command and Control**, Marshall, Va.: International Information Systems, Inc., 1984. Also see Andriole, ed., **Microcomputer Decision Support Systems**, Wellesley, Mass.: QED Information Sciences, Inc., 1986 and Andriole, **Interactive Computer-Based Systems Design and Development**, Princeton, N.J.: Petrocelli Books, Inc., 1983.
3. Ann Martin, et. al., **Conscreen**, McLean, Va.: Decisions and Designs, Inc., 1983.
4. Andriole, **The Design and Development of an Intelligent Planning Aid**, Woodland Hills, Calif. and Marshall, Va.: Perceptronics and International Informations Systems, Inc., 1984.
5. Andriole and Thompson, **Tactical Plan Repair and Revision: Concepts & Architecture for a Corps Level Interactive Aid**, Marshall, Va. and Albuquerque, N.M.: International Informations Systems, Inc. and Perceptronics, Inc., 1985. Also see Andriole and Thompson, **The Design and Development of an Intelligent Aid for Tactical Plan Generation and Evaluation: The INTACVAL Prototype**, Marshall, Va. and Albuquerque, NM: International Informations Systems, Inc. and Perceptronics, Inc., 1986. Finally, see Andriole, **INTACVAL Storyboard Workbook**, Marshall, Va.: International Informations Systems, Inc., 1986, for a close study of INTACVAL's functional intent.

Stephen J. Andriole is the president of International Informations Systems, Inc. He is formerly the director of DARPA's Cybernetics Technology Office. He was also a research analyst at Decisions and Designs, Inc. He has taught applied and quantitative national security analysis, interactive computer-based systems design, and decision support system development at the Johns Hopkins Univ., the Univ. of Maryland, and George Washington Univ. Andriole teaches a course for the Armed Forces Communications and Electronics Association entitled, "Decision Aids for Command and Control." Andriole received his bachelor's degree from LaSalle College, and his master's and doctorate degrees from the Univ. of Maryland. He is a frequent contributor to professional journals and has authored numerous books. Andriole is an active member of many national and international organizations and associations, including the Armed Forces Communications and Electronics Association and the Institute of Electronic and Electrical Engineers (Systems, Man and Cybernetics Society).