Naval Research Laboratory

Washington, DC 20375-5320

(2)

NRL/MR/5530--93-7397

A User Task Analysis for Command and Control Systems and Its Use in Human-Computer Interaction Research

DEBORAH HIX

Human-Computer Interaction Laboratory
Information Technology Division

September 3, 1993

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE

September 3, 1993

3. REPORT TYPE AND DATES COVERED

Interim 1 Oct 2 - 30 June 1993

4. TITLE AND SUBTITLE

A User Task Analysis for Command and Control Systems and Its Use in Human-Computer Interaction Research

5. FUNDING NUMBERS

PE - 62234N

6. AUTHOR(S)

Deborah Hix

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

Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/5530--93-7397

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

Chief of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Advanced Interfaces Section of the Human-Computer Interaction (HCI) Laboratory at the Naval Research Laboratory (NRL) is engaged in creating and evaluating interactive computer systems that address the unique issues encountered in developing innovative, high performance human-computer interfaces. A goal of this project is to build a testbed based on Naval command and control (C^2) systems as a vechicle for this research. Previous work at the HCI Lab has developed new *interaction techniques*—ways of using physical input and output devices to perform tasks in a human-computer interface. We now wish to transition into more realistic Naval-related applications for new techniques, in particular, command and control systems.

This report discusses a user task analysis performed for interactive computer-based C^2 systems; this task analysis is a basis for developing and evaluating new interaction techniques. As a result of this task analysis, appropriate user tasks for incorporation into the command-and-control-like testbed will be identified. The testbed will also incorporate some of the new interaction techniques, and will be used for empirical evaluation of these techniques in human-computer interfaces.

14. SUBJECT TERMS

Human-computer interaction
Task analysis
Command and control

15. NUMBER OF PAGES

34

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT

UNCLASSIFIED

20. LIMITATION OF ABSTRACT

UL

CONTENTS

A USER TASK ANALYSIS FOR COMMAND AND CONTROL SYSTEMS AND ITS USE IN HUMAN-COMPUTER INTERACTION RESEARCH

1. INTRODUCTION

The Advanced Interfaces Section of the Human-Computer Interaction (HCI) Laboratory at the Naval Research Laboratory (NRL) is engaged in creating and evaluating interactive computer systems that address the unique issues encountered in developing innovative, high performance human-computer interfaces. Such issues include integration of multiple devices into a single interface (multi-modal interfaces); presentation of vast volumes of data to decision makers; and collection, dissemination, and analysis of data for decision making. A goal of this project is to build a *testbed based on Naval command and control systems* (hereafter called C^2 systems) as a vehicle for this research. Previous work at the HCI Lab has developed new *interaction techniques* — ways of using physical input and output devices to perform tasks in a human-computer interface. We now wish to transition into more realistic Naval-related applications for new techniques, in particular, command and control systems.

This report discusses a *user task analysis* performed for interactive computer-based C^2 systems; this task analysis is a basis for developing and evaluating new interaction techniques. As a result of this task analysis, appropriate user tasks for incorporation into the command-and-control-like testbed will be identified. In particular, the motivation for this task analysis of C^2 systems is:

- to make our research on interaction techniques more application-oriented, especially to apply them in C^2 systems
- to identify some generic C^2 tasks that are appropriate for the new interaction techniques we will develop and evaluate
- to guide design of the testbed and generation of C^2-related task scenarios for experiments using new interaction techniques

Manuscript approved July 23, 1993.

The C^2-like testbed will incorporate some of the new interaction techniques being developed and implemented in the HCI lab, and will be used for empirical evaluation of these techniques in human-computer interfaces.

This user task analysis is the first step in the Multimode Interaction task within the Human-Computer Interaction project (RS34K76) of the Decision Support Technology block (RL2C) within the ONR Exploratory Development Program.

2. Overview of Task Analysis

2.1. What is task analysis?

Simply stated, *task analysis of an interactive computcr system* is the study of what functions an operator (or team of operators) can perform using a particular system [Kirwan & Ainsworth, 1992]. Task analysis attempts to determine what people want to do, or should do, or actually do while using an interactive computer system. Task analysis is a technique often used in early stages of interactive system development, especially for development of the system's human-computer interface. A task analysis can be performed at many different levels of abstraction, ranging from highest levels (e.g., monitor the theater of operations) to lowest levels (e.g., select a specific ship).

2.2. Why task analysis is important in general

Task analysis is critical to developing effective, efficient, and usable interactive computer systems [Diaper, 1989]. It is a powerful method used predominantly for producing requirements specifications, and it is even used for evaluation of interactive computer systems. Task analysis has a long history stretching back to the early part of this century. But it still is a technique that is not always considered as part of the interactive system development life cycle. This is due at least in part to the fact that task analysis is a difficult, time-consuming process. Many developers of today's highly complex interactive computer systems do not have resources — skilled personnel and time — to adequately perform the process of task analysis. But because of the focus, over the last decade, on high-quality user interfaces, and the recognition that task analysis is a method to help improve interface quality, more development teams are now employing it, at least to some extent, as part of their life cycle activities.

The alternative to performing at least some kind of task analysis is highly undesirable. Namely, computer-centric developers rely on their own opinions of what the interactive system should do and what its users actually need. And this all too often translates into what is easiest for developers to design and implement, rather than what users really want and need. Task analysis supports a *user-centric focus*, critical if interactive computer systems are to be effectively and efficiently usable. Thus, task analysis for developing interactive systems is focused on tasks that the *user* (operator) of the system will perform, and not just on system (non-operator) functions.

2.3. Why task analysis is important for HCI work at NRL

The need for this sort of work for current and future development of Naval computer systems is motivated in *Naval Command and Control: Policy, Programs, Peoole & Issues* [DiGirolamo, 1992]: "The *form* of C^4I — the computers, the sensors, the communications systems — has supplanted the *function*.... The focus most often is on equipment — new things to buy — or information — how to *present* data rather than how to use it."

Task analysis is a key to future work in the Human-Computer Interaction Lab at NRL. In particular, this task analysis of C^2 systems is an important intermediate step to creating a testbed based on real C^2 systems. This testbed, which is being modeled after tasks, functions, and scenarios found in real C^2 systems, will be used for empirical experimentation with new interaction techniques, as mentioned in Section 1. This task analysis provides a structured user-centric — rather than an ad hoc computer-centric — approach to laying the foundation for empirical work in human-computer interaction for C^2 systems. Through it we will evolve a testbed that offers an appropriate set of empirically evaluatable user tasks.

2.4. Why C^2 systems were chosen

Command and control is a highly diverse, extremely rich and demanding application area with a breadth and depth of tasks that its users perform. In essence, C^2 systems are highly complex decision support systems (e.g., [Hopple, 1986]). In this report, for simplicity's sake, we use the term C^2 to encompass all types of military "C^x" systems, recognizing that a migration (in both terminology and technology) has occurred from C^2

to C^3 (command, control, and communications) to C^3I (command, control, communication, and intelligence) to C^4I (command, control, communication, computers, and intelligence).

C^2 systems are those systems by which a commander gathers information, monitors global trouble spots, and directs and deploys forces (e.g., [DiGirolamo, 1992; Sage, 1987]). A C^2 system supports the planning, coordination, and execution of a tactical mission. Supported by one or more C^2 systems, a commander must conceive the battlefield and its ever-changing issues; integrate vast amounts of information from a broad variety of sources (e.g., charts, maps, text, messages) often in remote (away from the battle site) locations and under distracting conditions, and perform battle management based on that information. Thus the timely flow of information to and from commanders is critical. C^2 systems provide the coordinated operational control of sensors, weapons, support needs, and combat maneuvers by which modern warfare is conducted.

C^2 systems support a myriad of complicated tasks. Many of these tasks are performed by human operators, others are performed by the C^2 system itself. In the task analysis in Section 3, we concentrate on tasks performed by humans.

2.5. How to perform a task analysis

Most task analyses are performed at a fairly high level of abstraction, without going too low into the specific details of each task. Most task analyses are hierarchical in nature, starting at the highest levels of tasks, and working progressively into the subordinate tasks that comprise the tasks higher in the hierarchy. This turns out to be a very logical structuring for many interactive systems, since almost any task — with the exception of some lowest level primitives (e.g., click the mouse, press a key) — can be decomposed into subtasks. This hierarchical structure is also one that is fairly readily understood by system developers. It assists developers in structuring their work, allowing them to focus on specific parts of the task descriptions while still remaining in the context of the bigger picture of task activities to be supported by the interactive system under development. The top-down nature of a hierarchical task analysis helps analyze completeness of a design, yet can also be used in various stages of incompleteness without going into deepest details.

At higher levels of abstraction (and therefore less detail), the focus is typically on *what* the tasks are. However, at lower levels of abstraction, as tasks are decomposed into greater details of subtasks, temporal relationships and sequencing among those subtasks become important. Thus, at levels of greater detail, *when* to perform tasks, taking into account precedences among tasks, is as important as simply what those tasks are. Detailed task decomposition is important for designing the user interface, but this decomposition is often postponed until that design stage, rather than being done during (at least the initial) task analysis. The task analysis presented in this paper is a hierarchical task analysis, done at a rather high level in order to allow flexibility in decomposition and design later on.

Task analysis is a somewhat heuristic process, but there are several concrete methods that can be used in performing a task analysis for an interactive computer system. The most common methods include *elicitation techniques* for interviewing users (or potential users) and developers (or potential developers) of the particular system for which the task analysis is being done; *observing*, using, and analyzing similar existing systems; and *literature* reviews. We have employed all these methods in performing the task analysis of C^2 systems described in this report (see discussion at the beginning of Section 3 and the Appendix).

Results of a task analysis can be captured using a variety of representation techniques, including narration, charts, outlines, tables, diagrams, and other means. The representation technique should be chosen for clarity and consistency. Tasks are stated as a verb followed by an object — essentially an imperative sentence (e.g., plan route, monitor information about equipment status, analyze trends, and so on). The C^2 task analysis in the following section is presented in an outline-style form with accompanying narrative explanations, and summarized at the highest three levels (without the explanations) in a tabular format (see Table 1).

3. TASK ANALYSIS OF C^2 SYSTEMS

3.1. Approach to performing task analysis for C^2 systems

In performing this task analysis, we interviewed seven developers of C^2 systems to find out what functions they most often find in these systems. We observed or studied nearly a dozen C^2 systems to see what tasks they support, how their user interface is designed, what hardware they run on, and how easy they are to use. We will also observe other C^2 systems in the near future. We conducted an extensive literature review of C^2 systems, reading a broad variety of materials from numerous sources. Hardest to acquire access to were actual users of C^2 systems. We conducted a lengthy interview (more than four hours) with one user of these systems, and spent several hours with two users in the control room on-board a British frigate observing their use of a new type of data fusion/situation assessment system (see Appendix). We intend to interview more users in the future. Details of who we interviewed, what C^2 systems we studied, and literature sources reviewed are given in the Appendix to this report.

3.2. Primary user tasks in C^2 systems

Synthesizing all the information just described, a reasonable classification of C^2 systems tasks at the highest level indicates three primary user tasks in C^2 systems:

Sense
Plan
Act

Our task analysis shows that these activities are not sequential, as they are normally discussed. Rather, they heavily overlap, as shown in Figure 1, and a user of a C^2 system rapidly iterates among them during lower level tasks. This is due, at least in part, to the increased automation brought about by computers. The level of integratedness of these three main tasks of C^2 systems is such that it is virtually impossible to completely separate them.

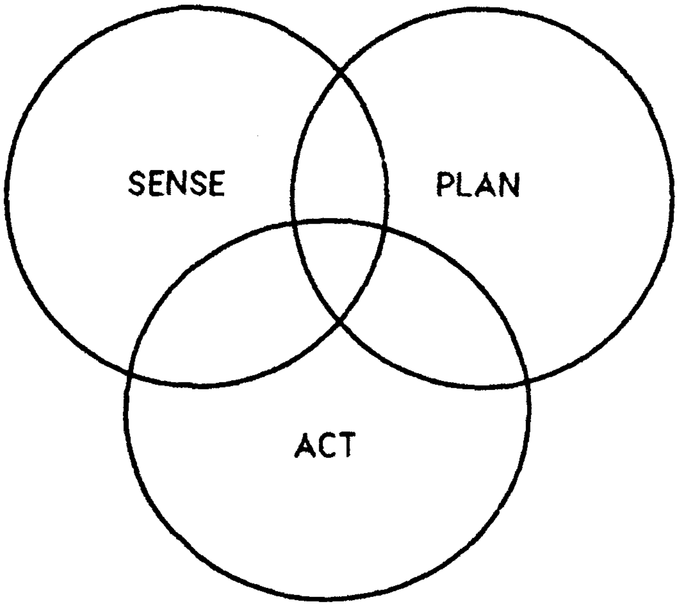

Figure 1. Primary user tasks in C^2 systems.

In order to establish a common terminology and attempt to avoid different interpretations of terms, following is a brief explanation of what the three primary user tasks — sense, plan, act — each mean as they are used in this task analysis. We have attempted not to alter the conventional Naval and military meaning of these terms, but in case some readers have different interpretations, we offer these explanations for clarification of our use herein.

Sense — To gather information about friendly and enemy forces and the environment over a period of time, in order to conduct a military operation. This involves monitoring a situation, and constantly and critically analyzing and challenging received data. Such information can come from a variety of sensors, including shipborne, airborne, subsurface-based (including submarines and ocean-floor undersea arrays), land-based, and space-based radars and electronic intelligence sources.

Plan — To prepare for possible alternative situations, with alternative approaches to managing these situations. This involves evaluating the credibility and significance of the sensed information and its correlation into a situational picture. It also involves making a decision about which course of action is best to take in order to accomplish the assigned mission.

Act – To issue orders to engage troops and equipment. This is followed by subsequent monitoring of the ever-changing real-time situation; essentially a return to the sense and plan tasks (as discussed above).

There is a fourth major task that is critical in C^2 systems, namely, *training*. However there is much agreement that, with improved C^2 systems (especially their improved human-computer interfaces) and systems integration, training is becoming less a separate activity. Rather it is becoming more "built in", so that the distinction between training and actual use of many C^2 systems becomes blurred. We will not consider training to be one of the primary tasks of C^2 systems, but rather assume that it is subsumed in learning to perform the three primary tasks of sense, plan, and act.

It is important also to note that *data fusion* underlies virtually all tasks and subtasks in C^2 systems, and it is especially critical during the sensing task. However, the efficacy of the correlated and fused data is critical to the other primary tasks of planning and acting. Real-time coordination of sensors at the unit and force level is needed to collect, process, and display tactical information received from all sensors. Integration of data from a variety of sensor types can be dispersed over large geographical areas, and can involve numerous personnel with a variety of capabilities. Data fusion is heavily supported by the computer, with some input from the user. Because of this, in this task analysis we will consider data fusion to be primarily a computer task, rather than a user task.

Further, we do not, in this task analysis, consider whether tasks are performed ashore or afloat or elsewhere, nor will we address the unique issues involved in distributed use by multiple users of C^2 systems.

3.3. Hierarchical task analysis of C^2 systems

Following are numerous examples of the kinds of *tasks* that are commonly *performed by humans using C^2 systems*. It is important to note that the classification of each task is based on which primary user task — sense, plan, or act — each one seems to best fit. There is inevitably overlap, and even some tasks that seem to logically fit parts of two or even all three of the primary tasks. Further, it is surely possible to decompose these tasks in many ways different from what is given here; we have attempted to associate tasks and subtasks in the way that appears most often used in C^2 systems. These examples are in *no way intended to be an exhaustive set of mutually exclusive tasks* performed using C^2 systems. Rather, they are intended to *represent the most interesting and critical numerous concerns and needs of a commander or battle manager*. Thus, because C^2 is such an extremely complex domain, this task analysis is not complete, nor is it the only possible decomposition, but rather is representative of the kinds of user tasks supported by C^2 systems. In particular, it emphasizes tasks that we are considering for use with new interaction techniques.

SENSE

There are at least two main tasks associated with the primary task of sensing:

- Gather information
- Detect threat

The types of tasks that relate to ***information gathering*** include the following:

* *Monitor current situation to determine force posture* — A commander requires accurate and timely information for situational assessment; this necessitates subtasks such as:
  -Acquire knowledge of friendly and enemy forces.
  -Determine potential battle areas.
  -Determine weather.
  -Determine logistical support available to sustain combat.

* *Monitor surveillance data* — Statusing of sea, subsurface, air, land, and space-based sensors; this decomposes into such subtasks as:
  -Determine target location.
  -Receive alert messages from sensor(s).

[Note: this subtask is worded user-centrically; the user *receives* messages. But "present alert messages" or "display alert messages" would be computer-centric wording of the task. Again, as emphasized previously, it is important to be careful to work in the user's world when performing a task analysis for user interface development.]

* *Monitor intelligence data* — This includes such subtasks as:

  -Monitor areas of interest or influence.

  -Determine ocean, terrain, weather, and other environmental conditions.

* *Monitor track information* — This can include current, historica:, and projected future information; subtasks include:

  -Hook (select) a track.

  -Request detailed information about a track.

* *Monitor information about equipment status* — This includes all types and classes of equipment. Subtasks, for example, for monitoring various kinds of equipment include:

  -Determine aircraft bias (i.e., friendly or enemy), course bearing, speed, altitude, range, fuel, weapons on board, and so on.

  -Determine ship bias, course bearing, speed, latitude and longitude, range, fuel, weapons on board, and so on.

  -Determine submarine bias, course bearing, speed, depth, range, fuel, weapons on board, and so on.

  -Determine radar bias, emitter type, location, speed (if appropriate), mode (e.g., normal, long range, mixed), and so on.

* *Monitor direct·· es* — Communications (e.g., for information or for battle orders) among commanders, subordinates, and other personnel; subtasks include:

  -Exchange information and orders within own ship and force.

  -Determine time from message origination to message receipt.

  -Determine time from message receipt to associated confirmation transmission.

  -Determine time from confirmation transmission to receipt.

[Note: At this rather low, detailed level of subtasks, we have an example of temporal constraints (see Section 2.5) on the order in which the subtasks are performed. One obviously cannot confirm receipt of a message until after the message has been transmitted.]

The types of tasks that relate to ***threat detection*** include the following:

* *Detect crisis warnings* — This can cover a broad range of types of crises, varying from major to minor, and including such subtasks as:

  -Determine when fuel reserves for a particular vehicle fall below prespecified levels.

  -Determine buildup in enemy at designated site.

  -Determine the designated number and class of ships rated in some particular readiness category.

SENSE/(RE)PLAN

There are at least two main tasks associated with the primary tasks of sensing and planning in which activities are so closely related that it is difficult to separate them into either the sense or the plan task:

- Monitor execution of course of action
- Assess battle damage (own and enemy)

The types of tasks that relate to ***course of action monitoring*** include the following:

* *Analyze trends* — Follow patterns in theater of operations; subtasks include:

  -Monitor ship, submarine, and aircraft movement.

  -Monitor communications conditions and stability.

  -Monitor status of fluctuations in weapons and fuel supplies.

  -Monitor ever-changing capabilities of an individual ship (e.g., new weapons in readiness, ready weapons fired, damage).

  -Monitor weather, ocean, and other environmental conditions.

* *Assess track information* — This can include current, historical, and projected future information; subtasks include:

  -Ascertain enemy's force composition and its cruising or battle formation.

  -Deduce as soon as possible enemy force's intentions.

  -Provide targeting data for rapid weapon employment.

* *Monitor engagement data* — Statusing of sea, subsurface, air, land, and space-based weapons; this includes such subtasks as:

-Receive launch detection messages from sensor(s).

The types of tasks that relate to ***battle damage assessment*** include the following:

* *Assess tactical performance* — Determining the effectiveness of an executing or executed course of action; this includes such subtasks as:

-Determine projected threat strength by counting objects of different types of enemy units under surveillance and their positioning.

-Compare projected threat size to number of active tracks.

-Receive hit assessment messages from sensor(s).

-Count number of hits.

-Evaluate weapons performance (e.g., by comparing number of expended interceptors to number of hits by weapon type and threat type).

-Evaluate sensor performance by comparing number of objects in track to number of expected objects in surveillance area.

-Determine expected time to reach target using weapons identification and weapons deployment characteristics.

PLAN

There are at least three main tasks associated with the primary task of planning:

- Generate and select alternative strategies
- Plan fire support
- Coordinate all assets

Since the planning task is one that will be exploited in our C^2 testbed (see Section 4.3 for an explanation of this decision), it will be discussed in a bit more detail here. The planning task is used to develop operations plans (OPLANs), which identify what forces will be used and how those forces will be employed to fight a battle or battles.

Obviously, a commander developing an OPLAN must have information about major combat forces and other resources that are available.

Planning can be characterized in more than one way, namely as deliberate or crisis (urgent), and as strategic or tactical [Sage, 1987]. Both deliberate and crisis planning involve predominately the same tasks, but deliberate planning can take days, weeks, or even months. Conversely, crisis planning is usually performed in a greatly compressed time frame, sometimes a few hours (or even for very low-level planning/replanning, a few minutes). Strategic (high level) planning decisions are made infrequently, while tactical (operational) decisions — for both crisis and deliberate planning — are generally made quite often, and must therefore be easy to modify.

The types of tasks that relate to ***alternative strategies generation and selection*** include the following:

* *Evaluate and allocate available resources* — This planning task is based on the mission commander's specification of the mission goal; subtasks include:

-Determine available resources data based on most recent messages.
-Hook (select) the track for an element and investigate its current state.
-Calculate resources (weapons, personnel, support needs) essential for executing mission.
-Compare resources available to pre-defined criteria.
-Determine actions taken in previous similar anticipated crisis situations.
-Predict vulnerability of own and enemy forces.
-Forecast probability of mission success, accounting for resource vulnerabilities and strengths.
-Control own ship's maneuvers.

* *Plan route(s)* — This includes subtasks such as:

-Determine (in order to avoid) current threat envelope of enemy units and weapon and radar ranges.
-Determine needed information for equipment and other elements involved in each plan (see, for example, types of information available for equipment, given previously).
-Determine environmental conditions.

-Determine point of intended movement.
-Take way points and build flight path with viewpoints along the way.
-Prepare and disseminate daily air tasking order (mission and support data required by flying units).

* *Evaluate surveillance and engagement data* — Performance of sea, subsurface, air, land, and space-based sensors and weapons; subtasks include:

-Count objects of a given type (e.g., a specific weapon, a ship type).
-Count number of weapons assigned to various targets by weapon type.
-Determine length of time until threat attacks.

The types of tasks that relate to ***fire support planning*** include the following:

* *Plan weapons allocation to targets (weaponeering)* — Application of resources (e.g., ship-to-ship missiles, ship-to-air missiles, ship-to-submarine missiles, close air support aircraft) against enemy targets to support force objectives; subtasks include:
-Access rules of engagement
-Determine projected time to achieve the desired readiness level for each element.
-Determine projected time that available resources can maintain the desired readiness level.

The types of tasks that relate to ***asset coordination*** include the following:

* *Direct own ship's and forces weapons* — This includes subtasks such as:
-Determine the number, percentage, and distribution of assets currently in a readiness posture.
-Determine any shortage in the number, percentage, and distribution of assets required to support engagement.

ACT

There are at least three main tasks associated with the primary task of acting:

- Select course of action
- Engage forces
- Terminate hostilities and active operations

The task of acting, compared to the other two primary tasks and as we consider it here, is rather straightforward and typically heavily automated. So it does not necessarily decompose into many lower levels of detailed subtasks. Following this task, the operatoı immediately iterates back to the sense and (re)plan tasks, continuing to move rapidly among them during a real-life situation.

Selection of a course of action is based on information and activities from the planning task. ***Engagement of forces***, as we consider it here, is simply the order to proceed with the chosen battle plan. We consider control (real-time after the engagement task is performed) of ships, subsurface units, aircraft, and other remote units to be tasks that are covered by sensing and (re)planning.

When the commander determines (largely via iterations back through the sense and (re)plan tasks) that further engagement will not contribute significantly to attainment of the battle objectives, the commander issues an order to ***terminate hostilities***. This allows combat capabilities to be conserved for future needs.

SENSE/PLAN/ACT

There are at least three tasks that are common substrata across all three primary tasks; these have already been alluded to, at least indirectly, in the previous task discussions:

- *Monitor situation display* — This is the composite picture of the theater of operations, constantly displayed, with multiple views, and controllable by the user.

- *Communicate with command authorities* — Each commander, potentially during performance of any task or subtask, must have communication with all associated personnel at all times.

- *Produce summary reports* — The operator can request various kinds of reports (e.g., resource use over specific time periods) during any of the primary tasks.

In addition to the high level primary user tasks, there are several very low level, primitive actions that are used extensively in the performance of virtually all tasks in C^2 systems. These include tasks/actions such as select, pan, zoom, scroll, move, and edit/change. Because these are so pervasive, we have not decomposed the previous tasks to this low level of detail.

3.4. Summary of hierarchical task analysis

A summary of the top three levels of this hierarchical task analysis for C^2 systems is given in Table 1. From this analysis, we will explore those user interactions that are most common and critical within C^2 systems, and use those as the basis for our experimental testbed.

SENSE

- Gather information
    * *Monitor current situation to determine force posture*
    * *Monitor surveillance data*
    * *Monitor intelligence data*
    * *Monitor track information*
    * *Monitor information about equipment status*
    * *Monitor directives*

- Deiect threat
    * *Detect crisis warnings*

SENSE/(RE)PLAN

- Monitor execution of course of action
    * *Analyze trends*
    * *Assess track information*
    * *Monitor engagement data*

- Assess battle damage (own and enemy)
    * *Assess tactical performance*

PLAN

- Generate and select alternative strategies
    * *Evaluate and allocate available resources*
    * *Plan route(s)*
    * *Evaluate surveillance and engagement data*

- Plan fire support
    * *Plan weapons allocation to targets (weaponeering)*

- Coordinate all assets
    * *Direct own ship's and forces weapons*

ACT

- Select course of action
- Engage forces
- Terminate hostilities and active operations

SENSE/PLAN/ACT

- Monitor situation display
- Communicate with command authorities
- Produce summary reports

Table 1. Summary of Hierarchical Task Analysis for C^2 Systems

4. C2 TESTBED FOR INTERACTION TECHNIQUES

4.1. Motivation for C^2 testbed

Based on results of this task analysis of C^2 systems, we are developing a generic testbed that will allow us to empirically evaluate new interaction techniques — both singly (one technique used in isolation) and multi-modally (multiple interaction techniques used in combination) — and other unusual aspects of human-computer interfaces. Earlier research in the HCI Lab on new interaction techniques has used rather simplistic, non-military, low fidelity domains and tasks [Jacob & Sibert, 1992; Jacob, 1993]. This research has yielded valuable information about alternative interaction techniques such as eye gaze and three-dimensional trackers. This is a clear indication that continued work in this area will have even greater value and long-term consequences for Naval C^2 systems if it is set in the context of a more realistic, command-and-control-like testbed to be used for experimentation.

Many of the techniques being considered for inclusion in the testbed are unusual, non-routine techniques (such as eye gaze, head mounted trackers, pen-based and other gestural input, voice, and so forth). Obviously, these techniques are, in some cases, drastically different in look, feel, and behavior from the standard mouse-and-keyboard interfaces that users most commonly use. As a result, people often mistakenly assume that the very uniqueness and novelty of such techniques make them naturally "better" — or at least more fun — for users. This is, of course, not necessarily the case. Thus, rather than assuming that neat new technology improves user task performance and satisfaction, we want to determine empirically whether it does or not, and in which cases and for what tasks. So a goal of using the C^2 testbed for experimentation is to evaluate user performance. Testbed development will also allow us to extend existing work at the HCI Lab on architectures for user interface software/management [Jacob, 1986].

4.2. Approach to developing C^2 testbed

We will use an incremental approach to producing the C^2-like testbed, incorporating scenarios that will allow users to perform the kinds of C^2 tasks described in Sections 3.3 and 5. The testbed will focus on generic, commonly performed tasks. The testbed architecture will be such that it can support the addition of new interaction techniques and devices to the testbed as "snap-ons", so that new techniques can be incorporated as quickly and easily as possible. This *generic, extensible testbed framework* will provide us with a suite of user tasks for a specific application area, namely C^2 . The suite of common tasks is representative of *what* tasks C^2 systems support; the interaction techniques relate to *how* the tasks are performed.

Some of the interesting issues involved in developing the testbed include how to design the human-computer interface incorporating the new interaction techniques so that they can effectively be compared to current techniques (e.g., mouse, keyboard). It is overly simplistic and optimistic to assume that an eye gaze device for a selection task substitutes directly for a mouse for that same task. Similarly, it would be possible to develop a testbed scenario using the eye gaze technique that is inherently better (but elusively provably so) than a mouse-driven version of the same scenario, and vice versa. In this case, comparative performance measures are obviously fallacious.

The basis on which we will choose interaction techniques for particular tasks, another interesting issue in testbed development, is not yet formalized. Obviously some interaction techniques work better for some types of tasks than others. A simple example is the awkwardness of entering alphanumeric characters one at a time by clicking on a keyboard display on the screen. A mouse simply does not work well for entering discrete alphanumeric characters; the traditional keyboard is much better. So in determining which techniques best fit which tasks, we will hypothesize, based on any existing work on similar interaction techniques, naturalness, our own expertise, and some pretesting of interaction techniques on simple tasks.

Another issue is the criteria for choosing the initial tasks for the testbed. The criteria are fairly obvious; namely, the tasks that are most often performed and the tasks that are most critical to accomplishing a mission. The tasks must also be designable and implementable in a reasonable time, so they cannot be tremendously complex. They must also be learnable by users in a reasonable time, since we will not be able to obtain users as subjects in experiments for large amounts of time. (This latter criterion, is, of course, counter to real C^2 systems, which can have 26 weeks of training.)

Some other technical issues in testbed development involve simulation in the testbed; it must be able to allow the user to dynamically adapt a plan if unforeseen events occur, and dynamically develop new plans to capitalize on occurring events. However, supporting this kind of environment is a difficult technical challenge. Also, a database of Naval information (e.g., about ships, weapons, personnel, and so on) must underlie the testbed. Determining the extent of the database and acquiring the appropriate information is another challenge yet unresolved.

4.3. Choice of tasks for C^2 testbed

Interestingly, as global threats diminish, C^2 sys‹:ms are used more for monitoring (sensing) and planning than for actual engagement and fighting, which is heavily automated in many C^2 systems. This implies that the primary user task of acting is not a particularly rich one for our testbed. Sensing is largely an output-oriented task, with presentation of data and information being of key importance. Planning is largely non-numeric and temporal, with graphical and visual needs. In addition to its output requirements, planning occurs through a great deal of interaction of the user with the C^2 system. In the early stages of our testbed development, we are most interested in interaction techniques for input and their effects on the output (presentation). Thus, planning is a logical high level task for us to develop, at least initially, in our C^2-based testbed. Strike mission planning is of particular interest.

The following section describes some of the kinds of scenarios and associated tasks we may develop and use to evaluate human performance in the C^2 testbed. For the reasons just explained, most of them are based on the types of tasks that are performed by the user during the primary task of planning. Details of exactly how these scenarios will be designed for the testbed (e.g., specific screen sketches or interaction techniques) are not given here, as they are not yet available. These scenarios are presented to show the kinds of generic command-and-control-like tasks we are considering for inclusion in the testbed, to be used for experimentation with users.

5. SOME POSSIBLE SCENARIOS FOR EXPERIMENTS

5.1. Outer air battle plan

An outer air battle (OAB) is one that is conducted at the extreme ranges of the combat air patrol capability in order to engage threat aircraft before they can launch antiship cruise (or other) missiles. Each OAB is typically of short duration (often measured in minutes) and of high level intensity. Changes to the tactical plan are limited once the battle starts, so the plan must be as optimal and accurate as possible. Tactical planning for an OAB is a candidate for users to perform with the testbed.

Planning of defensive tactics for an OAB involves such tasks as determining the required size of barriers set up by the combat air patrol that threat aircraft must cross to gain targeting information for weapons launches, establishing the necessary surveillance to detect and track the threat, positioning the air patrol to provide the required combat power, and providing tanker support to sustain the air patrol.

Planning of offensive tactics for an OAB involves such tasks as determining the threat's ingress routes and the threat location information necessary to support long-range combat air patrol engagement (e.g., threat location accuracy, timeliness of location information, and threat identity). Countertargeting is an approach used to increase the overall effectiveness of OAB operations tactics. The goal of countertargeting is to deny or confuse the threat's targeting ability so that their missiles are launched on false targets or they are forced to move closer to the target area to improve targeting quality. This deterrent increases the time and battle space available to engage the threat.

A display that would support this tactical planning task for an OAB would include at least the current location, formation, and other information about threat aircraft, combat air patrol aircraft, airborne early warning aircraft, and friendly ships. It could, for example, use overlays that the user requests to show the outer edge of the air battle area, and these overlays could change dynamically as locations of the threats change. Changes in icon appearance could serve as alerts to warn the operator that vehicles are moving into or out of the threat envelope. Defensive barriers and protected areas could also be displayed as user-requestable overlays. Later versions of the testbed might include real film footage of a particular area of operations, the ability to "fly over" a particular feature, and extensive use of simulated three-dimensional information displays.

5.2. Generation of course of action

A somewhat different way of performing the kinds of tasks just described for an OAB is: Given a mission analysis, a terrain analysis, a situation analysis, and a friendly force analysis, generate a course of action.

*Mission analysis — This is the clarification and understanding of the mission given to a battlegroup. The plan to accomplish the mission must be consistent with the higher level commander's concept of the operation and doctrine.

*Terrain analysis — This is the description and evaluation of any militarily significant aspects of the terrain over which a battlegroup will operate.

*Situation analysis — This is the identification and evaluation of enemy capabilities (courses of actions we believe the enemy can conduct), and inference of enemy intentions (courses of actions we believe the enemy will conduct).

*Friendly force analysis - This is characterized by vast amounts of reliable data, describing available units with both objective (location, type, strength, equipment status, ammunition and fuel status, and so on) and subjective (readiness, morale, training, and so on) attributes.

Any of these planning tasks could be set in the OAB scenario described previously.

5.3. Two-phased battle plan

Many battle plans are formulated in two phases:

1. An analysis of the effectiveness of each weapon-to-target allocation is performed. The effectiveness of each weapon against its assigned target is determined as the expected proportion of the target destroyed if a particular weapon is fired at it. This is calculated using numerous factors related to the current weapons, targets, and battle field situation (e.g., range, position, personnel readiness, counterfire ability, resupply, ammunition states, maintenance state, weather)
2. The effectiveness is then used to calculate and plan the overall allocation of weapons to targets, including the expected total destruction for each plan. From this information, an optimal (or near-optimal) plan can be chosen.

The OAB discussed in Section 5.1 could again be an effective scenario for performing this planning task.

5.4. Defense of overlapping field of view

Following is a quite complicated task. Consider two ships defending an area, each sharing an overlapping field of view of the whole area. Each ship can independently select targets as it identifies them. Both have some information about the other (e.g., location, bearing, speed, weapons on board and their status). Determine which ship fires when a target appears in the intersection of their ranges such that a maximum amount of targets are hit with a minimum number of shells and minimum communication messages between the two ships over the duration of the mission. Because the ships can move, the intersecting area changes. This is a quite difficult problem; in the testbed we will, at least initially, keep tasks simple, changing situations fairly slowly.

5.5. Some other scenarios and tasks

Following is a list of some additional ideas for tasks that could be used or adapted for use in our C^2 testbed:

Determine whether a particular Iranian ship, alleged to have laid mines in the Persian Gulf, did, in fact, traverse the Gulf at a time when it could have laid the mines.

Find the shortest/safest route between two points.

Determine all courses of action that Red (the enemy) might be expected to consider.

Determine the length of time expected until the threat attacks.

Detect and track one or more airliner-sized aircraft rumored to be in the eastern Mediterranean.

5.6. Commander's catechism as ideas for experimental tasks

An interesting set of questions that can serve as an informal source list for basic steps in virtually any high level C^2 task is captured in the "Commander's catechism". This list may be useful as we develop our generic testbed with its common C^2-like tasks. It is taken from Lt. Gen. John H. Cushman, as quoted in [Wohl, 1981, p. 120]):

Where am I? (situation)
Where is the enemy?
What is the enemy doing?
How can the enemy hurt me most?
What have I got to thwart the enemy?
How can I do the enemy in?

Am I in balance? Movements in order? Reserves? Logistics?
How long will it take me to .... ?
How long will it take the enemy to ... ?
How will it look in an hour? Six hours?
What is the most important thing to do right now?
How do I get it done?

Prior to World War II, the effectiveness of a commander heavily depended on that commander's ability to act upon this set of queries. Until World War II, the pace of battle was such that a commander and support staff could gather the necessary information, study that information, generate and evaluate alternative hypotheses, and choose the most appropriate course of action. World War II, however, brought increased speed of ground and air movements, coupled with increased radio communications. This began stretching the ability of commanders to react fast enough to a more quickly changing theater of operations. Organizational, procedural, and doctrinal changes all occurred to address this new need. A look at the catechism shows its age most notably in the question "How will it look in an hour? Six hours?" Given the state of today's technology and the speed of our vehicles, weapons, and communications, six hours may be, for many real-time battle situations, too far to plan ahead. However, most of the queries in the catechism still apply, at a general level, today, and are therefore useful as the basis for possible scenarios for use in a C^2 testbed.

5.7. Approaches to experimentation

In our experiments with the testbed, we hope to show quantifiably (and qualitatively) improved user performance on various tasks or classes of tasks using particular interaction techniques (as opposed to conventional interaction techniques). There are essentially two approaches to such evaluations: isolated evaluation and comparative evaluation.

In *isolated evaluation*, we build the testbed with its new interaction technique(s), such as eye gaze for selection or a head mounted 3-D tracker for panning and zooming. Then we let people perform tasks using the new techniques and collect data (quantitative data on objective performance measures such as time and number of errors and on subjective performance measures such as questionnaire rankings, as well as qualitative data such as user opinions and comments). There is no real comparison to other approaches (either via real C^2 systems or via the testbed). In isolated evaluation, we cannot make claims about improved user performance of one interaction technique over another; we can only claim that user performance with a particular technique is satisfactory, where satisfactory has been defined before the experiments.

In *comparative evaluation*, we build the testbed with its new interaction technique(s). We then also build into the testbed more conventional interaction technique(s) (e.g., mouse or trackball) for performing the same intended tasks. For some tasks, there may be a real C^2 system with conventional interaction techniques that is appropriate to use as a comparison base for the new interaction techniques. In either case, data are collected from users using both kinds of interaction techniques (new and conventional) to perform the same or similar tasks. These data are then compared for significant differences, in order to determine which gives better user performance.

An example of the kind of absolute evaluation we might perform involves a new interaction technique called pre-screen egocentric projection [Templeman, 1993]. Using this technique, the user wears a light-weight helmet with a three-dimensional Polhemus tracker mounted on the front. As the user's head moves, the display on the screen changes in response. As the user moves from side to side, the display pans. As the user moves closer to or further from the screen, the display zooms in and out, respectively. This technique allows the user to selectively reveal different portions of a screen within limited screen real estate. Because this is a completely new interaction technique, we need to determine some fundamental information and basic parameters about its use and human-computer interface design that might incorporate it. For example, we need to know the comfortable range of movement a person can make in each direction, particularly while wearing a helmet. Obviously, this depends on factors such as the person's height, weight, and visual acuity, and also on the person's age and flexibility. Developing the parameters for this "envelope of comfort" could be an enormous task, so it would be necessary for us to keep it simple and determine some informal basic limits as quickly as is feasible. Then we can move to more complex tasks using egocentric projection, addressing, for example, such issues as how users temporally use egocentric projection (e.g., for how long do they zoom in), the best way to do and undo a freeze of some or all of the screen (when the user wants the display to stop changing in response to head movement), and whether egocentric projection can be used to help reduce unnecessary detail and clutter on the display. A next logical step would be to study use of egocentric projection in combination with other interaction techniques (i.e., a multi-modal interface).

An example of a comparative evaluation we might perform could also involve egocentric projection. Because this technique is so suitable for panning and zooming, it is logical to compare users' performance with the egocentric projection device to, say, the conventional mouse-driven approach to panning and zooming. This ability to mix and match interaction techniques with tasks is one of the main motivations for our C^2 testbed.

6. SUMMARY

A goal of current work in the Advanced Interfaces Section of the HCI Lab at NRL is to focus on function, rather than form, for interactive system development. In particular, we want to focus on realistic user tasks and study which interaction techniques provide improved user performance of tasks. Results from this work will, in the longer term, be a force multiplier for amplifying the effectiveness and efficiency of Naval C^2 systems.

Source after source indicates that the Navy is poised to refocus its goal for command and control systems, away from the dazzling technology and toward the "operational technologist" (so-called by Vice Admiral Jerry O. Tuttle) — in today's and tomorrow's world, the operator of the most sophisticated command and control systems ever developed and deployed.

ACKNOWLEDGMENTS

The author wishes to acknowledge interesting and spirited discussions with colleagues at the Naval Research Lab, in particular Dr. Jim Templeman, Dr. Rob Jacob, and Linda Sibert. Many of their ideas are captured in this report. This work was sponsored by the Decision Support Technology block (RL2C) within the ONR Exploratory Development Program, and is managed by Elizabeth Wald.

REFERENCES

Diaper, D. *Task Analysis for Human-Computer Interaction.* Ellis Horwood Ltd. Chichester, England. (1989).

DiGirolamo, V. *Naval Command and Control: Policy, Programs, People & Issues.* AFCEA International Press. (1992).

Hopple, G.W. Decision Aiding Dangers: The Law of the Hammer and Other Maxims. *IEEE Transactions of Systems, Man, and Cybernetics.* (November - December 1986).

Jacob, R.J.K. A Specification Language for Direct Manipulation Interfaces. *ACM Transactions on Graphics 5,* 4 (1986).

Jacob, R.J.K., "Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces," In *Advances in Human-Computer Interaction, Vol. 4,* H.R. Hartson and D. Hix(eds.), Ablex Publishing Co., Norwood, N.J. (1993).

Jacob, R.J.K. and L. Sibert. The Perceptual Structure of Multidimensional Input Device Selection. In *Proceedings of CHI'92 Conference.* ACM Press. New York. (1992).

Kirwan, B. and L.K. Ainsworth. *A Guide to Task Analysis.* Taylor & Francis Ltd. London, England. (1992).

Sage, A.P. Human Information Processing Principles for Command and Control. *Principles of Command and Control.* J.L. Boyes and S.J. Andriole (eds.). AFCEA International Press. (1987).

Templeman, J. The Egocentric Projection of Scenes in Front of the Screen. In *Proceedings of the Navy Scientific Visualization and Virtual Reality Seminar.* Washington, DC. (1993).

APPENDIX

Following is a list of the most important resources used in performing the user task analysis for C^2 systems.

Some of the systems studied or observed:

Data Fusion/Situation Assessment Demonstrator — Traveled to Norfolk in late April, spending a day on-board, including the control room, of the HMS Marlborough, the British Royal Navy's newest class of frigate. The system performs picture compilation to produce a tactical picture of the ship's area of interest, and includes automated knowledge-based decision support. This is a new type of system that is being tested on the Marlborough.

SLQ-32 — Electronic support measure system to detect, identify, and track courses of potential threats, active countermeasures capability that enables the system to jam tracking radars of detected threats. Receive and transmit functions (listening and jamming) can be performed simultaneously.

JTIDS (Joint Tactical Information Distribution System) — Communications systems for secure, high-speed data communications with long-range surveillance aircraft and carrier-based fighters, for long-range interception of enemy naval bombers; enhances interoperability of separate communication systems.

NTDS (Navy Tactical Data System) — For automated organization and display of information for C^2 teams for threat detection and assessment and weapon-to-target allocation. Primarily for anti-air warfare, but limited anti-surface vessels and anti-submarine warfare capabilities.

OSIS (Ocean Surveillance Information System) — Ashore sites directly involved in tactical threat analysis. Primarily for submarine and air attacks. Provides near real-time, all-source messages and warnings, threat assessment, positional and movement information, and over-the-horizon targeting support to national, theater, and fleet users.

JOTS (Joint Operational Tactical System) — An advanced decision support system used by tactical action officers, battlegroup commanders, and other personnel largely for planning.

KOALAS — A system for automating some of the decision-making process in an aircraft, being developed at NRL in the HCI Lab and the AI Center.

PowerScene — A new system being developed at Cambridge Research Associates in McLean, Virginia, and funded by the Naval Air Systems Command. It renders raw digital image data into high fidelity perspective scenes for a variety of tactical simulations for situational awareness, orientation, planning, and training.

Advanced Airborne Situation Awareness System — Another system being developed at Cambridge Research Associates. Its purpose is mission rehearsal, preview, and training. It currently uses a helmet for some of the user interactions with the system, but expects to use goggles within a year or so.

There are numerous other C^2 systems that will be studied, including several simulations on NRL (e.g., strike planning system), at NSWC Dahlgren, and at the Johns Hopkins Applied Physics Laboratory in Baltimore.

Some of the people interviewed:

Two operators on-board the HMS Marlborough (see previous list of systems)
Lt. Cmdr. Steve Harris, AI Center, Bolling AFB, Washington DC
Dr. James Ballas, Information Technology Division, NRL
Mr. David Tate, Information Technology Division, NRL
Dr. L. Scott Randall, Cambridge Research Associates, McLean VA
Dr. Jerry Owens, JJM Systems, Crystal City VA
Dr. Ann Harrison, JJM Systems, Crystal City VA
Mr. Dan Averill, JJM Systems, Crystal City VA

Some of the literature reviewed:
In addition to the literature in the reference list, the following sources provided useful information:

IEEE Transactions on Systems, Man, & Cybernetics
Signal
AFCEA publications

Specific references of particular interest include:

Boyes, J.L., and S.J. Andriole. *Principles of Command and Control.* AFCEA International Press. (1987).

Copernicus Architecture. Office of Chief of Naval Operations. Washington DC.

Hyde, J.P., J.B. Warren, and C.E. Kesson. C3 Planning in Crisis Response. *Signal.* (March 1986).

Information Technology for Command and Control: Methods and Tools for Systems Development and Evaluation. Special Issue of *IEEE Transactions on Systems, Man, and Cybernetics*. S.J. Andriole and S.M. Halpin (eds.). (November - December 1986).

Johnson, S.E.,, and A.H. Levis. *Science of Command and Control: Coping with Uncertainty*. AFCEA International Press. Washington DC. (1989).

Loescher, M.S. Origins of Modern Naval Command and Control. *The Copernicus Architecture*, OP-094. (December 1990).

Metersky, M.L., J.M. Ryder, and M.A. Leonardo. A Change in System Design Emphasis: From Machine to Human. In *Advanced Technology for Command and Control Systems Engineering*, S.J. Andriole (ed.). AFCEA International Press, Washington DC. (1991).

Shore, D. Trends in Command and Control. *Signal.* (October 1974).

Ward, R.E. and W.J. Brennan. Navy Battle Force Command and Control — A Tactical Coordination and Tactical Communications Management Perspective. *Signal.* (November 1985).

Wohl, J.G. Force Management Decision Requirements for Air Force Tactical Command and Control. *IEEE Transactions on Systems, Man, and Cybernetics.* (September 1981).

Other activities:

Attended two two-day conferences in June 1993 on C^2 Systems, both held at Ft. McNair in Washington DC:

10th Annual Conference on C^2 Decision Aids
1993 BRG C^2 Research Symposium