TECHNICAL REPORT 80-02

Quarterly Technical Report:

The Design and Transfer of Advanced Command and Control (C2) Computer-Based Systems

James F. Wittmeyer, III

CSM INC.

Computer Systems Management, Inc.

1300 WILSON BOULEVARD, SUITE 102 • ARLINGTON, VIRGINIA 22209

TECHNICAL REPORT 80-02

Quarterly Technical Report:

The Design and Transfer of Advanced Command and Control (C^2) Computer-Based Systems

by

James F. Wittmeyer, III

Computer Systems Management, Inc.

1300 WILSON BOULEVARD, SUITE 102
ARLINGTON, VIRGINIA 22209

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(9) Quarterly technical rept.

(14) CSM-

REPORT DOCUMENTATION PAGE

[redacted]

1. REPORT NUMBER: 80-02	2. GOVT ACCESSION NO.: AD-A083 935
3. RECIPIENT'S CATALOG NUMBER: 1 Jan - 31 Mar 80	
4. TITLE (and Subtitle): (6) The Design and Transfer of Advanced C^2 (Command and Control) Computer-Based Systems	5. TYPE OF REPORT & PERIOD COVERED: Quarterly Technical Report 1/1/80 - 3/31/80
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): (10) James F. Wittmeyer, III	8. CONTRACT OR GRANT NUMBER(s): (15) MDA903-80-C-2155
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Computer Systems Management, Inc. Suite 102 1300 Wilson Boulevard Arlington, Virginia 22209	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Order [redacted] 382
11. CONTROLLING OFFICE NAME AND ADDRESS: DARPA 1400 Wilson Boulevard Arlington, Virginia 22209	12. REPORT DATE: 3/31/80
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Defense Supply Service-Washington (DSS-W) The Pentagon Washington, D.C. (11) 31 Mar 80	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Recommended for public release; distribution unlimited

(12) 52

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Recommended for public release; distribution unlimited

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

C^2, computer-based systems design, technology transfer, systems development, and C^2 systems demonstration

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

C^2 computer-based systems design should be informed by design filters, selection of proper terminal and "dialogue" interfaces; computer-based systems transfer must be informed by logistical considerations and follow-through documentation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

411 586

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TECHNICAL REPORT 80-02

QUARTERLY TECHNICAL REPORT:

THE DESIGN AND TRANSFER OF ADVANCED COMMAND AND CONTROL (C^2) COMPUTER-BASED SYSTEMS

ARPA Order No.:	3829
Contractor:	Computer Systems Management, Inc. 1300 Wilson Boulevard Arlington, Virginia 22209
Effective Date of Contract:	11/5/79
Contract Expiration Date:	9/30/81
Contract No.:	MDA903-80-C-0155
Principal Investigator:	James F. Wittmeyer, III (703) 525-8585
Contract Period Covered:	1/1/80 - 3/31/80
Short Title of Work:	Advanced C^2 Computer-Based Systems

This research was sponsored by the Defense Advanced Research Projects Agency under ARPA Order Number 3829; Contract Number MDA903-80-C-0155; and Monitored by DSS-W. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either express or implied, of the Defense Advanced Research Projects Agency or the United States Government.

SUMMARY

This Quarterly Technical Report covers the period from January 1, 1980 to March 31, 1980. The Tasks/Objectives and/or Purposes of the overall project are connected with the design, development, demonstration and transfer of advanced command and control (C^2) computer-based systems; this report covers work in the computer-based design and transfer areas only. The Technical Problems thus addressed include the development of approaches, methods and options for C^2 computer-based systems design and transfer. The General Methods employed include the development and use of design filters and hardware/software options analysis. Technical Results include the development of a design filter and a profiling of hardware and software options. Various Hardware Configurations are suggested as optimal design systems. Future Research will explore the role of microcomputers in the design and transfer process.

THE DESIGN AND TRANSFER OF ADVANCED

COMMAND AND CONTROL (C^2) COMPUTER-BASED SYSTEMS

TABLE OF CONTENTS

FIGURES

1.0 INTRODUCTION

1.1 Problem Statement

The Defense Advanced Research Projects Agency's Cybernetics Technology Division (DARPA/CTD) has as its primary mission the development, application, and transfer of computer-based systems for improved Department of Defense (DoD) information mangement and display, forecasting, decision making, training and human performance, especially as such activities occur in Command and Control (C^2) environments. Unfortunately, however, the research and development (R&D) process connected with the development of advanced C^2 computer-based systems is fraught with problems. Specifically, such problems may be categorized as follows:

1.1.1 Computer-based Systems Design

1.1.1.1 Neglect of "front-end" analysis. This problem runs rampant throughout all of the projects that have as a final product either computer-based systems software or analytic or descriptive data sets. Moreover, front-end analysis has seldom been conducted in any of the areas of the computational needs of the hardware and software spectrum. Neglect of such analysis invites disaster and circumvents normally acceptable programming practices. An example illustrating the myriad of problems that can occur without proper design analysis is the current state of the Terrorism Research and Analysis Project (TRAP).[1] TRAP software is a TEKTRONIX 4051 resident BASIC program. Many demonstrations of its capabilities show the extreme value it has as a sophisticated Indications and Warnings (I&W) and operations research system. However, it is designed to sun on a very specific type of graphics microprocessor for which there is only one manufacturer. It can also only be demonstrated on large screen projection via a Hughes scan converter and a specially modified 4051 (of which there are only two in the Washington, D.C. area). "Front-end analysis" was neglected in this example. Had it been conducted such potentially costly oversights in hardware and software design might have been prevented. (This example is not intended to undermine the research efforts of specific individuals or organizations, but rather to illustrate a critical problem in the design of complicated computer-based systems.)[2]

1.1.1.2 Expensive and fundamental "disconnects" among the programmer, the intended product, and the ultimate user. So often, through a very basic misunderstanding in the design phase, there occurs a strange phenomenon which seperates the intended purpose of the reseach tool from its preparation and construction. This "distance" often causes enormous problems in the final stages of software implementation and transfer. If in fact, the product delivered is neither what was intended nor what can be used, it must be re-written, re-tested, re-validated and re-documented--generally a very expensive process. The cost in man-hours alone is sometimes staggering. Further costs of late-delivery, and other projects suffering because of a re-shuffling of priorities are also not inconsequential. It is important to keep sight of who the ultimate user is, where the tool will be utilized, when it must be ready to be effective, and how it should be implemented. For example, if the product is a low cost, short lead-time one, it need not go through a rigorous design stage once the above criterion are met. Restated, if the product is to be "quick and dirty" this fact should predict to overriding developmental techniques. However, these are the only kinds of products which should be allowed to slip through an intensive design critique. Examples demonstrating this "fundamental disconnect" are plentiful; e.g. the Early Warning and Monitoring System (EWAMS) was produced with great care and planning. The goal was to create a unique monitoring and forecasting system that would be both focused and easy to use by the intelligence community, especially through the Defense Intelligence Agency's National Military Intelligence Center (DIA/NMIC).[3] The (interim) product transferred to the DIA/NMIC had no user manual; it used too many research and statistical terms; it did not reflect the needs of a daily "watch" analyst; and it was not written so that it could be transferred easily to a non-UNIX*,[4] non-timesharing, and heavily utilized DIA/NMIC computer system. Again, the purpose is not to undermine legitimate efforts, but rather to point out the critical necessity of exacting procedures that must be followed early in the design phase, and that a "fundamental disconnect" between the developer and user can increase transfer cost by orders of magnitude.

1.1.1.3 Non-standardized data sets and codebooks. In the early years of the conceptualization and creation of the Demonstration and Development Facility (DDF), it became evident that a large portion of the DDF user community would be tasked with the creation and maintenance of various data sets. This function has no less importance than the analytic software tools which often evolve from such data sets. But, here too we find design flaws. Care should have been taken, at the outset, to standardize the coding and collection of the data, particularly with a view to how they might later be analyzed and processed via a computer-based delivery system. Two examples of where improper data standardization in the design phase resulted in unnecessary man-hour efforts include the Cross National Crises Indicators (CNCI)[5] project, which collected a very specialized data set consisting of "surprise attack" event data. The data collected was intended to follow World Event Interactions Survey (WEIS) codebook techniques; however, the resultant data, differed so severely from pure WEIS data that a special version of the EWAMS program had to be adapted to utilize desired indicators and graphics. The modification of EWAMS to fit the "surprise attack" data was a relatively small task, but costly nonetheless. In the second case, data driving the CACI, Inc. U. S. Executive Aids (EXECAID) had been so well designed, it could easily fit, in its entirety, on one TEKTRONIX 4051 casette tape. But problems resulted when members of the Inter-university Consortium for Political Social Research (ICPSR) wanted copies of the data only, and found that it was uniquely tied to the software with which it is used. In both cases errors in data set standardization occurred causing delays and expensive efforts to correct.

1.1.2 Computer-based Systems Development

1.1.2.1 Hardware. Here selection and usage problems are often encountered in many of the more advanced research projects. Research products that use ineffective computer systems will fail no matter how excellent the value of the research product. Correct hardware selection is critical to the successful development and transfer of a research product. Factors such as portability, maintenance costs, backup systems, commercial availability, and life-expectancy are all important to the hardware/software marriage. For example, the earliest application of the Perceptronics, Inc., Ultra-Rapid Reader (URR) was written on the DDF using a TEKTRONIX 4025 graphics terminal.[6] The problem with this selection was the combination of phosphorous persistance and character size for projection. Even though this was a pilot application much of its success depended upon the readability of the output. The method of "hardware selection by availability" is insufficient when the hardware inhibits the applications software. It also is extremely important to address the data requirements as they apply to hardware selection. Problems in this area have plagued the DDF since its inception. For example, the size alone of the WEIS data set is so large that in many instances the data set required so much DDF disk space that there wasn't sufficient space remaining to permit further software development on either EWAMS or other projects.[7] Proper product design and development could have by-passed these problems.

1.1.2.2 Software. Here, development problems arise throughout all programming activities in every organization.[8] Poor software development procedures result in much wasted time, effort and cost. Typical problems concentrate usually around language selection and implementation. Many languages are incompatible with one another. This problem of imcompatibility exists not only between operating systems (such as UNIX vs. RSX11-M) but on computer systems such as HONEYWELL Level 6 vs. DEC 11/70 as well. For example, most UNIX trained systems programmers use a term called "vanilla UNIX". Vanilla UNIX is the basis for the next level of incompatible versions of the same Bell Laboratories software. The others, "BBN UNIX", Rand UNIX", "NSA UNIX" (closest to vanilla) and "ISC UNIX" all differ so dramatically that it becomes difficult to easily transport user or system software from one UNIX system to another.[9] There are of course literally hundreds of modifications of the UNIX version 6 operating system. This may all be corrected with the release of version 7 UNIX and PWB UNIX, but history will most likely repeat itself and lend more versions of more levels of the same complicated operating system.[10] This problem (multi-UNIX's) is further complicated by the fact that UNIX has the ability to support many compilers, interpreters and assembly languages. In fact UNIX has a language 'C' and YACC--The 'C' System Compiler and "Yet Another Compiler Compiler"! The languages of the system could include three or four BASIC's, AS, RATFOR, FOR, FC, F4P, PASCAL, LISP, and so on. The problem is real, and a pragmatic approach must be taken to proper software language development. Decisions must be made at the operating systems level as to the availability of S/W facilities such as DoD1.[11] Decisions must be made when and how DoD1 should be addressed. For example, what research now under development will still be so when DoD1 becomes universal?

Pragmatic decisions must be made not only as to the user availability of such languages but the fundamental choice of a language for an application. For example, if a short term project, with a specific transfer application requires APL, then it should be used. But, on the other hand, if the need for the resultant research tool is more universal, then it would be wrong to allow programming to begin in APL. For example, Decisions and Designs, Inc. and the Computer Corporation of America, Inc. constructed a very valuable set of decision making and evaluation aiding tools for the U. S. Marine Corps.[12] The results were spectacular but the software was written in APL, and the Marines could only accept transfer of software written in COBOL.

Also under the jurisdiction of software development comes the very important topic of documentation. Throughout the development phase of either software tools or data sets, time _must_ be spent on building effective documentation. Documentation takes the forms of internal documentation, systems specifications, users manuals, and codebooks. Improperly documented code, or code without documentation lives only as long as its author. It loses its efficiency; it can no longer be maintained; it becomes forgotten. The cost of rewriting software, because the documentation no longer exists, is prohibitive. For example, the first release of the BBN "steamer" program to the DDF had no users manual and no system specifications. The program could only be run by guess work, and modifications to it were difficult and time-consuming.

1.1.2.3 _Coordination._ Far too often redundant code, or redundancy of effort exists in program development. The problem comes about partially through the "not invented here" syndrome. Competitive researchers do not wish to admit that there may be others that have attacked a similar problem with success. Therefore, they close their minds to the existance of a solution to the problem at hand. In other cases the "re-invent the wheel" syndrome applies. By sheer fact that competition exists between contractors, inventions, new ideas and methods are not shared among the user community. Result: the wheel is re-invented over and over and over again. An example of lack of coordination can be seen in data collection efforts by both the Brookings Institution and CACI, Inc.[13] Both research projects collected similar crisis data sets. At no time during these projects was there any interfacing of data or ideas. At the end of these projects the DDF received two tapes, one from each contractor, both similar, and neither acknowledging the existance of the other. Other examples of poor coordination exist. Software written at the University of Southern California (USC) to allow efficient terminal input for data collection and management was installed on the DDF for use with the WEIS data set.[14] However, many of these utilities are not used, causing wasted efforts, on the part of others who could use the techniques for WEIS and other data collection.

Falling also under the heading of poor dissemination and coordination, is the management of research tools for internal development. Part of the DARPA/CTO mission is the development of computer-based systems for improved information management, display, forecasting, decision making, training and human performance research products. These new tools could be used internally by the researchers to help achieve a "bubbling-up" or "percolator" effect. For example, Decisions and Designs, Inc. constructed many useful tools in decision making. Many of these tools such as the Multi-Attribute Utility (MAU) program could be used to make more efficient research decisions.[15]

Another problem in coordination is the "closed community" syndrome. Too often researchers within the CTO program refuse to look beyond work outside of their own community. For example, Artifical Intelligence (AI) techniques developed by the Information Processing Techniques Office (IPTO) may be useful to CTO researchers. It is cost effective to glean all relevant work regardless of the sponsoring office, agency or department.

1.1.3 Demonstrations

1.1.3.1 Opposing "cultures". Contractors typically by the nature of their roles as researchers are disconnected from the the nature of organizational decision making. As a result, contractors are seldom able to provide the necessary context for an effective demonstation. This is not meant that they cannot (or are unwilling to) bridge the gap between product development and application. Instead, it is to highlight the issue of "comparative advantages". Researchers are good at what they do, but seldom can make an effective leap from development to application. They are often at a comparative disadvantage when they try. Without this leap however, work may be rendered useless and ineffective. Here, the real value of a Demonstration and Development Facility (DDF)--cognizant of the critical distance between the researcher and the "user"--comes into play. Since contractors are generally unskilled in the art of interacting effectively with government, military and civilian personnel at all levels, they harbor latent misunderstandings and confusions about the nature of the operational world. All of this predicts to the failure on the part of the researcher to incorporate into his or her computer-based system the necessary user-oriented features so critical to generating interest, appeal and, ultimately, acceptance. An anecdotal example recalls the first use of the Joint Chiefs of Staff (JCS) regional capability of the EWAMS for use by a military analyst, who promptly asked why he must relearn all the nations of the world in three letter WEIS representation rather than the SOP military two-letter designation.

1.1. . Demonstration staging. The success or failure of a developed software product often completely depends upon the nature of the environment in which the product is first introduced to a potential user. Nothing is so disconnected from (or detrimental to) a presentation of significant work, than to present it in a bad environment.

It most surely will jeopardize the acceptance of that product. For example, imagine the effect of a demonstration of EWAMS in an environment cluttered by bearded professors and disorganized papers (a not untypical research environment) versus one that takes place in a controlled and secure command-center-like environment. The results are obvious.

1.1.3.3 Premature demonstrations. Another problem all too often encountered is the premature release and demonstration of a product. Contractors by their nature have very little feeling for the proper timing, and in this regard, often demonstrate too early with disastrous possibly even fatal results. A brief discussion of an advanced computer-based training system failure will drive this point home. Recently, during a set of training system demonstrations to the Director of DARPA, one system implemented on an APPLE II micro-computer failed to meaningfully communicate the value of the training aid. This resulted in an unwarranted skepticism of the value of micro-processor-based training aids. Quite simply, this was a problem which could have been averted by more careful planning and evaluation of the system's "readiness".

1.1.4 Transfer

1.1.4.1 Targeting. Too often we are not cognizant of the overall requirement that we are trying to address via the development of a computer-based system. That is to say, we must remain mindful of the target, need, use and problem to be solved, and avoid becomming overly impressed with techniques or inventive solutions, which although may be major breakthroughs in and of themselves, must address a targeted need.

1.1.4.2 Transfer site requirements. There exists a significant problem in the failure to adequately survey the hardware and software at a proposed user's site, often resulting in a serious mismatching. Transfer efforts must be done professionally and expertly. Those individuals responsible for the evaluation of the users hardware configuration and facilities have little room for error. Beginning with the design phase, all work should be aimed at effective solutions to specific development problems. The final phase in the solution of that problem must take into consideration every possible condition which could prevent the integration of the new work with the existing technology. Problems encountered in the transfer process are highly visible to the ultimate user and may overshadow the impression of competence as well as confidence in the delivered work. No matter how valuable the analytic tool, if its final installation appears to be a comedy of errors, then a value of similar worth will be placed upon the product being installed. As yet this situation has been only narrowly avoided.

1.1.4.3 Documentation. More commonly, there is a failure to provide necessary instructional documentation along with the software and data sets being transferred. This problem has the potential of being as dangerous as a faulty transfer. Without the proper documentation, interest in the product will wane. Because it is too difficult to understand, it will not be used. An example which fits the above problem description occurred at the very first DDF transfer to the Naval Postgraduate School (NPS) in Monterey, California. The software and data transferred was the complete crisis management system. The software products included the URR, U.S. Executive Aids (EXECAID), EWAMS, and various utility programs. The data sets transferred included the URR and supporting data, EXECAID and supporting data, and a complete set of WEIS analytic and descriptive data. The installation of the software onto the NPS PDP 11 went reasonably well. All programs tested well and transfer was complete. But, when asked for more information about how to use all of the capabilities of the software--adequate documentation was unavailable. When asked about some of the statistical techniques used by the creators of the software, again there were problems. Although overall the transfer was quite successful, without sufficient documentation including users manuals, sample output and succinct research papers the initial impact of the products was minimal.

1.1.4.4 Marketing. Another problem that can occur is the failure to adequately assess the bureaucratic back-drop necessary for thorough, formal transfer. While this is not to say that informal transfer is not valuable, it is to say that often without top-down approval from the "chain-of-command", transfer could be at best delayed, or at worst, prevented. A clear example of this failure to "market" a transfer at the proper decision-making levels exists in the "continuing" transfer of the EWAMS software and data to DIA/NMIC. Just prior to final transfer of the software to the NMIC assurances were made as to the availability of a "stand-alone" PDP 11/45 computer system, complete with assistance and guidance from the pentagon staff. This never occurred. The "stand-alone" computer system never became available, and the DIA internal and contractor support never materialized. Even though the DDF did in effect manage to demonstrate that the software and data were ready to be transferred, and that it all could be done to the original DIA specifications, it was still delayed indefinitely. Why? Simply because of the failure to realize an everyday axiom of marketing, that is, to gain approvals from those in key decision-making positions.

1.2 Proposed Solution

It has become clear that the research and development process especially as it applies to the development of advanced C^2 computer-based systems has many problematic areas. The cataloguing of these problems is the first step toward the framing of a solution. Step two requires the establishment of a set of general and specific mission-oriented objectives by which the solutions can be determined and applied. The general solutions (and overarching project goals) appear below; some specific solutions appear in section 2.0.

Objective 1 - The establishment of a new and vital design phase for all candidate DARPA/CTD contractor software. A rigid set of design standards will be applied to all new and intended software products. Through the application of these standards the software and data requirements can be categorized into groups which will serve to indicate the needs of the project which are to be provided by DDF personnel. Further analysis can be made at this time to determine the pre-estistence of data sets that may fulfill the requirements of the effort. As a function of this design phase clear cut alternatives can be examined, weighed and selected thereby insuring that the proposed effort is both well conceived and within the limits of acceptable programming practices.

Objective 2 - Provide a superstructure of effective hardware and software development capabilities. Every effort will be made to accommodate the hardware and software requirements of the DDF user community. This effort will be centered around the operation of DEC PDP 11/70 dual mainframe computer systems. The full resources of this service will be extended in an attempt to adequately support the on-going development, design and transfer efforts. However, the provision of this service is not the only goal. Also included in the development phase are: hardware selection assistance, programming assistance, training and advice, creation of documentation standards and a concerted effort to keep all researchers informed of technical advances made both inside and outside the DARPA/CTD community.

Objective 3 - Take a leadership role in the organization and presentation of professional and effective demonstrations. The advanced computer-based systems and data developed for DARPA/CTD must endure intensive critique by a viewing audience. Therefore, it becomes increasingly important that a concerted effort be made to establish and conduct policies for and training about effective demonstrations. Here too resides the need for physical as well as consultantive services. It is very important to provide a well organized program of demonstrations in an environment conducive to generating interest, appeal and hopefully acceptance of the research products being developed by those representatives of the operational world who may wish to adopt new computer-based systems.

Objective 4 - Provide expertise ready to address the problem of transferring selected software and data from research status to operational service. As part of the overall mission of DARPA/CTD successful research products must, after design, development and demonstration, be transferred to a proposed user's site. Care must be taken to adequately analyze the systems and facilities available at the transfer site so as to assure that the least amount of difficulty is encountered during the transfer. During the transfer phase all supporting documentation should be assembled to accompany the software and data, comprising a complete package which is useable and understandable. Also, care should be taken to coordinate with the transferee to insure that the products being delivered are what is expected as well as needed.

Objective 5 - Create a new management approach to the organization of the DDF. Most, if not all, of the technical problems currently facing the DDF user-community cannot be resolved by technical adjustments alone. These problems require a new management model, a re-organization to accomodate technical advancement.

In the coming months--indeed throughout Fiscal Years 1980 and 1981--Computer Systems Management, Inc. will endeavor to solve these problems through the implementation of a specific technical/management/administrative approach. This report covers our early efforts and covers the period from January 1, 1980 to March 31, 1980.

2.0 THE DESIGN & TRANSFER OF ADVANCED C^2 COMPUTER-BASED SYSTEMS

2.1 The Design, Development, Demonstration, Transfer and Documentation Tasks

During the course of Fiscal Years 1980 and 1981 all of the tasks associated with computer-based systems research will be addressed. After briefly addressing the context for our work--the DARPA/CTD FY80 research program--we will turn in section 2.1.2, to computer-based systems design and, in section 2.1.3, to computer-based systems transfer.

2.1.1 FY80 DARPA/CTD Research Program

It must be noted that the DARPA/CTD research program is constantly evolvong. CSM is now working from one blueprint; at the same time we continually monitor changes in the nature and direction of the program in order to assess possible impact upon the operation of the DDF.

2.1.2 Computer-Based System Design

One method for designing computer-based systems requires that the intended system pass through a set of filters as suggested below in Figure 1. Filter one asks whether the system is to be a research system or an application system. Research systems are generally aimed at developing techniques and ideas, so that they may be proved worthy. On the other hand, applications systems draw upon previous research in such

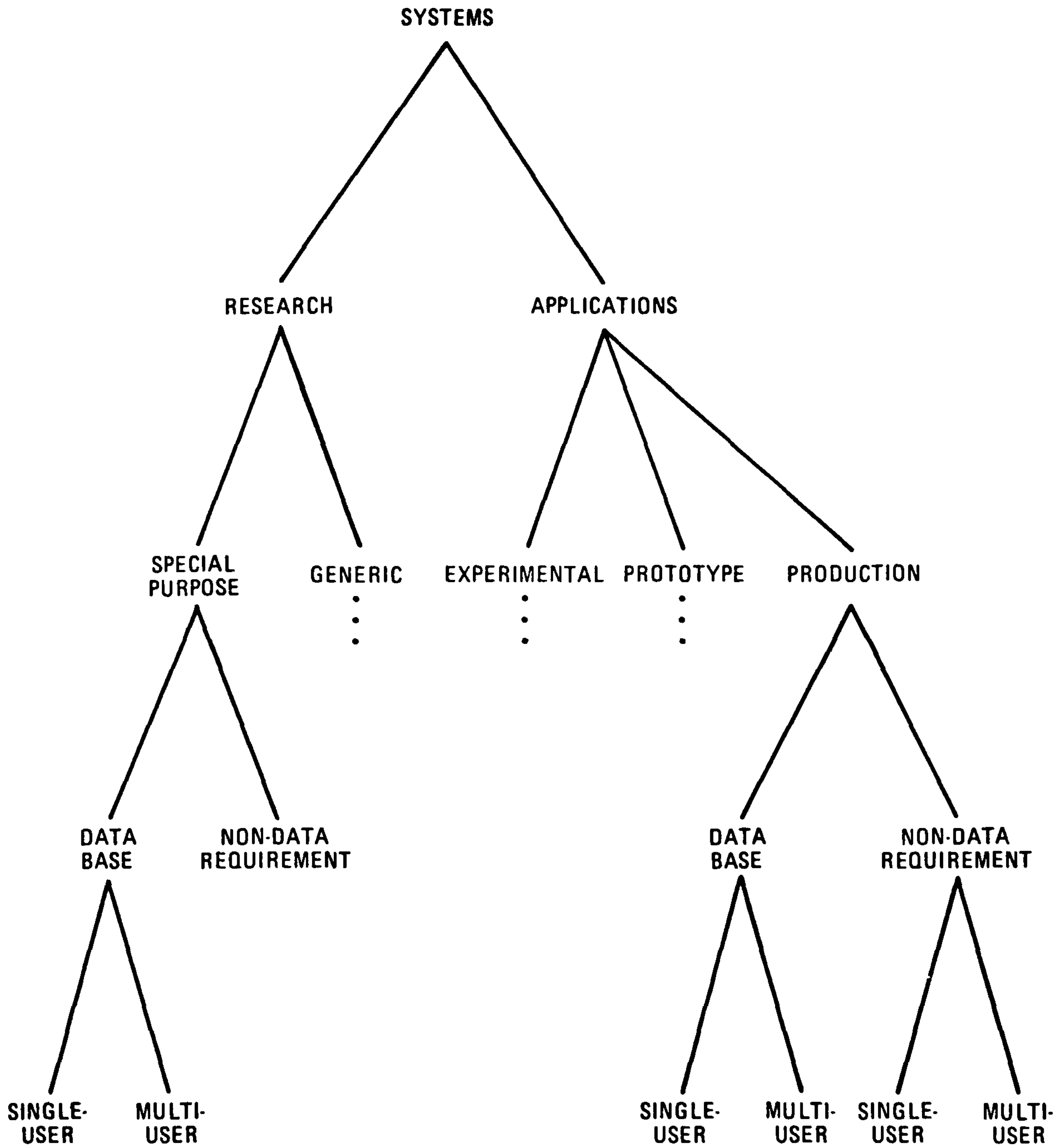

Figure 1
"DESIGN FILTERS" FOR DARPA/CTO COMPUTER-BASED SYSTEMS

a way as to expand previous ideas into complete experimental or working models.

The second set of filters further expands research systems into two categories: special purpose or generic. Special purpose differs from generic in the sense that it has an intended single purpose. Applications systems are subdivided into three types: (1) experimental, (2) prototype, and (3) production. A sequence in the maturization of an applications software system is first that of an experimental model, which thus leads to prototype development and finally full production model(s).

The third level tests the system for its data requirements; does it require a prestored data set or is the data to be generated on-line during execution of the system? An example of prestored data exists in the large WEIS data set required for execution of several modules of the EWAMS. Whereas ADT systems elicit user probabilistic assessments which are then saved for further systems analysis. The last set of filters to test the user requirement asks the question: will the system be on-line to multi-users, or will it be a single-user (stand alone)? The answer to this question is as important in the design of a system as the other filters.

For example, had the EWAMS been passed through these filters the first level would have distinguished it as an applications system. It would have been distinguished as prototype in nature, requiring a prestored data set, and multi-user-oriented. Unfortunately, these filters were not employed and numerous difficulties plagued the development, demonstration and transfer phases of EWAMS. Retrospectively, EWAMS might best have been designed as two systems employing the production versus prototype filters.

These filters, then, are illustrative of the kind that CSM attempts to develop and apply for DARPA/CTD at their direction as their computer-based systems development requirements arise.

2.1.2.1 Disconnects. The application of the above filters will reduce--if not eliminate--many of the "disconnects" discussed in section 1.1.1.2. It is our view that prudent passage through the above discussed design filters will minimize the "distance" among the research, the intended applications or research area, and the ultimate user.

2.1.2.2 User Emphases. Throughout the design process, the intended user or users should be studied carefully. At the lowest level of the design filtering process, then, particular attention must be paid and specific analyses should be performed. Such analyses include, but are not limited to:[16]

- Users

  - Their behavior in general; how to determine the properties of a particular user population; the implications of those properties for the interactive system;

- Tasks

  - What tasks users perform; how to determine tasks involved in an application;

- Requirements Analysis

  - How to analyze information requirements; how to select appropriate types of problem-solving, clerical and user support aids; allocation of basic tasks to user or computer; modeling of user-system interactions; evaluation of basic design;

- Interactive Dialogue

  - Properties of different dialogue types; selection of appropriate dialogue type(s); detailed design of command language, system access structures, tutorial aids, etc.;

- Output Devices and Techniques

  - Properties of display devices; implications of dialogue method for display device selection; selection or design of display device(s); detailed display design, formatting, coding techniques, etc.;

- Input Devices and Techniques

  - Properties of input devices; implications of dialogue methods for input device selection; selection or design of input device(s); and

- Evaluation of System Performance
  - Use of subjective evaluations, objective performance measures.

<u>Users</u> should thus be identified and classified as suggested below:

- Naive Users (Inexperienced with computers)
  - Computer-naive users are actually a very heterogeneous group, but have many common properties. Naive users benefit greatly from computer-initiated dialogue, usually require more tutorial features. Correct implicit "mental model" of computer systems and interactive dialogue cannot be assumed, must be explicitly conveyed by system. Naive user population has many detailed implications for dialogue design. Smooth transition from naive to experienced user is often difficult in current systems.

- Managers (Including Military Commanders, etc.)
  - Managers tend to have highly variable information needs; current systems are often too rigidly constraining to satisfy those needs. Managers tend to place high negative value on own effort, have considerable discretion with respect to mode of system use or nonuse. Thus, very low "impedance" is required to capture manager as direct user. If dissatisfied, manager tends to resort to "distant use" (interposing operator between manager and system) or partial use.

- Scientific and Technical

  - High proportion report dissatisfaction with available automated tools. These users often respond to such dissatisfaction by becoming personally involved in design or implementation of software tools, or by altering task to match available tools.

Tasks as well should be specified ideally in taxonomy form.

User requirements analyses should preceed any and all implementation. Some requirements analysis techniques appear below:

- Use of questionnaires to obtain ratings of the relative importance of various categories of information and system features;

- Use of questionnaires to obtain estimates of time spent on each task associated with recipient's job;

- "Repertory Grid Technique", a questionnaire-based technique for determining user's "cognitive frame of reference";

- "Delphi Technique", a survey technique in which recipient's responses are fed back, anonymously. Recipient responds again, while aware of previous responses of entire group;

- "Policy Capture", one of several techniques for developing quantitative relationships between perceived system desirability and specific system features. In this case, relationship takes the form of a multiple-regression equation;

- Interviews with users to determine information requirements, decision points, organizational constraints, etc;

- "Ad Hoc Working Group", in which subject-matter experts devise system requirements by analysis and negotiation;

- "Critical Incident Technique", in which users are asked, via interview or survey, for information about incidents of particular success or failure in the process of which the computer system will be a part;

- Job analysis techniques, such as task analysis, link analysis, and activity analysis, which attempt to characterize user behavior on the basis of direct observation;

- "Paper" simulation, in which the possible function of a computer system is simulated by human observers, in order to obtain information about the user's problem-solving and information-seeking behavior;

- "Protocol analysis", in which the user comments extensively on his activities during simulated problem solving, and formal content analysis of the resulting commentary ("protocol") is used to make inferences about user behavior and problem-solving processes; and

- Interactive simulation or gaming, in which the actual system, or an interactive computer simulation of the system, is used with a contrived scenario to observe user behavior and system performance.

The selection of interactive dialogue technique is also critical. Some properties of interactive dialogues appear below on an assumed scale:

- Initiative
  - Initiative is concerned with whether the user of the computer initiates the individual information transactions within the dialogue. If the computer asks questions, presents alternatives, etc., and the user responds, the dialogue is "computer-initiated". If the user inputs commands without such computer "prompting", the dialogue is "user-initiated". "Mixed initiative" and "variable-initiative" dialogues are also possible;

- Flexibility
  - Flexibility is a measure of the number of ways in which a user can accomplish a given function. High flexibility can be achieved by providing a large number of commands, by allowing the user to define or redefine commands, etc.;

- Complexity
  - Complexity is related to flexibility. Complexity is a measure of the number of options available to the user at a given point in the dialogue. Low complexity can be achieved by using few commands, or by partitioning the commands so that the user selects from a small set at any given time;

- Power
  - Power is the amount of work accomplished by the system in response to a single user command. In a dialogue with powerful commands, the user may

accomplish, with a single command, an operation which would require several commands in a system with less powerful commands. Power is related to flexibility and complexity;

- Information Load

  - Information load is a measure of the degree to which the interaction absorbs the memory and/or processing resources of the user.

- System Response Time

- Communication Medium

Types of interactive dialogue to be selected by the designer appear below:

- Question-and Answer

  - Computer asks a series of questions, to which user responds;

- Form-filling

  - Computer presents form with blanks. User fills in blanks;

- Menu Selection

  - Computer presents list of alternatives, and user selects one or more;

- Function Keys with Command Language

  - User indicates desired action by depressing keys, each of which represents a command, command modifier, or parameter value;

- User-initiated Command Language
  - User types commands, perhaps using mnemonic abbreviations;
- Query Language
  - User inputs questions or data-base access procedures to a data base system. System produces response or report;
- Natural Language
  - Dialogue is conducted in user's natural language (e.g., English); and
- Interactive Graphics
  - Generation of pictorial displays, ability of user to select displayed entities and spatial locations by pointing or similar nonverbal means.

The evaluation and selection of output devices is also critical. Some variations are presented below:

- Refreshed CRT
  - The ordinary, refreshed CRT is currently the "basic" computer display. A good deal of data exists concerning appropriate visual properties of CRT displays. Studies which have compared user performance using CRTs with performance on other display devices do not provide a satisfactory basis for selection decisions;
- Storage Tube CRT

- For some graphical applications, direct-view storage tubes may be preferable to refreshed displays. The storage tube allows very high-density, flicker-free displays, but imposes significant constraints on interactive dialogue. Although information exists concerning the basic functional advantages and disadvantages of such displays, no empirical data pertaining to human factors concerns were found;

- Plasma Panel Display
  - Plasma panel displays are inherently "dot", or punctate, displays, and studies of symbol generation method are relevant. Little empirical information exists on human performance aspects of plasma displays per se;

- Teletypewriter
  - Reasonable guidelines exist with respect to the design of teletypewriter terminals, including both physical and functional properties;

- Line Printer
  - Research on typography is voluminous and directly applicable. Research dealing directly with the line printer used in computer output is scanty, but consistent with findings of typographic research (e.g., mixed upper-lower case is best for reading comprehension). Guidelines are not known to exist, but could be constructed with additional survey of typographic research literature. Use of line printers for "pseudographic" displays is common, little discussed in the literature. Pseudographics is an inexpensive way to convey simple graphical information, and should probably be used more widely in batch applications;

- Laser Displays

  - Reasonable human factors guidelines with respect to visual properties have been proposed, but these displays are not widely used;

- Tactile Displays

  - Although some tactile displays have been proposed or even developed, little human factors research has been done other than that concerned with prosthetics;

- Psychophysiological Displays

  - Psychophysiological input is technically feasible now, but psychophysiological displays are still only a topic for research; and

- Large-Screen Displays

  - There is conflicting evidence with respect to the performance effects of large-group vs individual displays. The main advantages of large-screen displays are a larger display area and the existence of a single display which is clearly the same for all viewers. Unfortunately, higher display content is not achievable due to the resolution limits of existing technology (e.g., light valve displays), and may be unachievable in principle, since the large-screen display usually subtends a <u>smaller</u> visual angle than an individual display located close to the user.

Input devices come next; options appear below:

- Keyboard;

- Lightpen, Lightgun;
- Joystick;
- Trackball;
- Mouse;
- Graphical Input Tablet;
- Touch Panel
- Knee Control
- Thumbwheels, Switches, Potentiometers;
- Tactile Input Devices;
- Psychophysiological Input Devices;
- Automated Speech Recognition;
- Hand Printing for Optical Character Recognition (or for Subsequent Entry by Typist);
- Mark Sensing;
- Punched Cards; and
- Touch-Tone Telephone.

Our design filters are thus extremely functional and higher order; the selections of detailed computer-based system components

--oriented to the user and actual use--are far more complex and critical to the successful development and use of advanced C^2 computer-based systems.

2.1.3 Computer-Based System Transfer

If adequate requirements analyses are performed, the transfer process can be greatly improved. Candidly, it is infrequently the case that requirements analyses are performed; instead, most computer-based systems are designed as a function of perceived requirements. Consequently, all too often systems are retrofitted to the intended user. Proposed here is thus the conduct of requirements analyses using some or all of the techniques described above in order to properly select the "right" dialogue technique, and input and output devices.

2.1.3.1 User Manuals. So often a system is either partially completed and then transferred or completely developed without requirements analyses and then transferred. Almost always the User's Manual is an afterthought conceived and constructed in a vacuum relative to the intended user. First and foremost, User's Manuals should never be system capabilities driven: they should always be requirements (of the intended user) driven. Secondly, they should be animated, that is, heavily steeped in graphical/visual explanations and illustrations--again in the context of user requirements. Third, technical detail, while pleasing to the scientist/developer, should be in the Appendix or non-existent. Fourth, they should be iterative and flexible. Fifth, they should be short. Finally, they should be modular.

2.1.3.2 Transfer Logistics. Following a successful demonstration it is necessary to assess prospects for and difficulties associated with actual transfer. This generally involves assessments regarding the transferee's hardware and software capabilities. Such assessments enable CSM/DDF personnel to tailor and/or modify the system(s) to be transferred in a expeditious manner. CSM thus prefers to, when feasible and at the direction of DARPA/CTD, conduct on-site analysis of the transferee's capabilities and affect transfer accordingly. Every effort is then made to accommodate his capabilities and other requirements necessary to affect the transfer, and special attention can be devoted to maintaining the professionalism connected with the particular transfer in question and the transfer process in general.

In an effort to avoid a transfer short fall, and the loss of valuable feedback, CSM prefers to initiate a follow-through transfer procedure whereby a user will not find himself abandoned after an on-site visit and an initial tutorial session. Instead, CSM maintains a detailed transfer record and interacts with the transferee on a scheduled and ad hoc basis.

Finally, obviously the success of any transfer is dependent upon the quality and quantity of system(s) documentation made available to the transferee. As part of its follow-through strategy, CSM provides initial documentation as well as updated (systems and data) documentation.

3.0 CONCLUSION

This second Quarterly Technical Report has examined the issues of computer-based systems design and transfer connected with the overall design, development, demonstration, and transfer of advanced command and control (C^2) computer-based information, decision, forecasting, training and readiness systems.

4.0 FOOTNOTES

[1] The TRAP system was developed by CACI, Inc. for DARPA/CTD.

[2] Similar problems occurred in connection with the development of the DARPA/CTD Early Warning and Monitoring System (EWAMS) and the Executive Aids for Crisis Management.

[3] See S. J. Andriole, Progress Report on the Development of an Integrated Crisis Early Warning System, Decisions and Designs, Inc., McLean, Virginia, December, 1976; and Judith Ayres Daly and Thomas R. Davies, The Early Warning and Monitoring System: A Progress Report, Decisions and Designs, Inc., McLean, Virginia, July, 1978.

[4] UNIX is a trademark of the Western Electric Company. It was developed at Bell Laboratories by Kenneth Thompson and Dennis Ritchie.

[5] See Gerald W. Hopple, Final Report of the Cross-National Crisis Indicators Project, University of Maryland, College Park, Maryland, 1978.

[6] The ultra-rapid reader (URR) was developed by Perceptronics, Inc. for DARPA/CTD. Succinctly, it is a system for rapid reading which presents text in short bursts, one word at a time in the center of a CRT. The technique enables a user to focus his or her eyes in one position and not move them as the words appear one at a time on the screen. See Steven Levin, The Ultra-Rapid Reader, Perceptronics, Inc., Woodland Hills, California, February, 1979.

[7] The WEIS data set contains approximately 24M bytes.

[8] The software problem is so visible in the DoD that it attracks a disproportionate amount of attention (in terms of dollar investment) each year during Congressional budget testimonies.

[9] Bolt, Berenek, and Neuman, Inc. (BBN); National Security Agency (NSA); Information Science Center (ISC).

[10] Prudence dictates that version 7 of UNIX be scrutinized carefully before implementation in order to avoid unnecessary problems.

11 DoD1 is the intended (higher order) language to be adopted by DoD as DoD standard at some point in the future (the buzzword is now ADA).

12 See James J. Allen, MCCRESSA Users Guide, Computer Corporation of America, Arlington, Virginia, August, 1978.

13 See Leo Hazlewood, et.al., Planning for Problems in Crisis Management, CACI, Inc., Arlington, Virginia, June, 1977 and Barry M. Blechman and Stephen S. Kaplan, Force Without War, The Brookings Institution, 1978.

14 See Gary M. Guilbert, The GEN System for Entering, Validating, Updating and Reporting of Events Data, International Relations Research Institute, University of Southern California, August, 1978.

15 See Ward Edwards, How to Use Multi-Attribute Utility Measurement for Social Decision-Making, Social Science Research Institute, University of Southern California, August, 1976 and Dennis M. Buede and Janice E. Ragland, Cost-Benefit Analysis Applied to the Program Objectives Memorandum (POM), Decisions and Designs, Inc., McLean, Virginia, November, 1978.

[16] This section relies heavily upon H. Rudy Ramsey and Michael E. Atwood's excellent Human Factors In Computer Systems: A Review of the Literature, Science Applications, Inc., Englewood, Colorado, September, 1979, pp. 8-133.

5.0 REFERENCES

Allen, James J. MCCRESSA Users Guide. Arlington, Virginia: Computer Corporation of America, August, 1978.

Andriole, S. J. Progress Report on the Development of an Integrated Crisis Early Warning System. McLean, Virginia: Decisions and Designs, Inc., December, 1978.

Blechman, Barry M. and Stephen S. Kaplan. Force Without War. Washington, D. C.: The Brookings Institution, 1978.

Buede, Dennis M. and Janice E. Ragland. Cost-Benefit Analysis Applied to the Program Objectives Memorandum (POM). McLean, Virginia: Decisions and Designs, Inc., November, 1978.

Daly, Judith Ayres and Thomas R. Davies. The Early Warning and Monitoring System: A Progress Report. McLean, Virginia: Decisions and Designs, Inc., July, 1978.

Edwards, Ward. How to Use Multi-Attribute Utility Measurement for Social Decision-Making. Los Angeles, California: Social Science Research Institute, August, 1976.

Guilbert, Gary M. The GEN System for Entering, Validating, Updating, and Reporting of Events Data. Los Angeles, California: International Relations Research Institute, University of Southern California, August, 1978.

Hopple, Gerald W. Final Report of the Cross-National Crisis Indicators Project. College Park, Maryland: University of Maryland, 1978.

Levin, Steven. The Ultra-Rapid Reader. Woodland Hills, California: Perceptronics, Inc., February, 1979.

Ramsey, H. Rudy and Michael E. Atwood. Human Factors In Computer Systems: A Review of the Literature. Englewood, Colorado: Science Applications, Inc., September, 1979.