LEVEL II

A092293

AD A106787

TECHNICAL REPORT 81-04

Final Technical Report:

# The Design, Development, Demonstration, 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


81 11 09 129

TECHNICAL REPORT 81-04

# Final Technical Report:

# The Design, Development, Demonstration 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)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER<br>CSM-81-04 | 2. GOVT ACCESSION NO.<br>AD-A166 | 3. RECIPIENT'S CATALOG NUMBER<br>787 |
| 4. TITLE (and Subtitle)<br>The Design, Development, Demonstration, ~~Documentation~~, and Transfer of Advanced $C^2$ Computer-Based Systems. | | 5. TYPE OF REPORT & PERIOD COVERED<br>Final Technical Report<br>7/1/81 - 10/31/81 |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>James F. Wittmeyer, III | Command and Control | 8. CONTRACT OR GRANT NUMBER(s)<br>MDA903-80-C-0155 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Computer Systems Management, Inc.<br>1300 Wilson Boulevard<br>Arlington, Virginia 22209 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>ARPA Order ~~Number~~-3829 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>DARPA/DSO/SSD<br>1400 Wilson Boulevard<br>Arlington, Virginia 22209 | | 12. REPORT DATE<br>10/31/81 |
| | | 13. NUMBER OF PAGES<br>29 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)<br>Defense Supply Service-Washington (DSS-W)<br>The Pentagon<br>Washington, D.C. | 11 31 Oct 81 | 15. SECURITY CLASS. (of this report)<br>Unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE<br>N/A |

16. DISTRIBUTION STATEMENT (of this Report)

Recommended for public release; distribution unlimited.

9 Final technical rept. 1 Jul-31 Oct 81

12 35

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)

Computer-based systems; video systems; design; development; demonstration; testing; documentation; transfer; evaluation; users; software; hardware; requirements; interactive dialogue; input devices; output devices; software engineering; video discs; micrographics; shared analysis systems; networks; teleconferencing

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

The design, development, demonstration, documentation, and transfer of advanced $C^2$ computer- and video-based systems is enhanced by systematic requirements, user assessments, and prudent software engineering. New opportunities lie in the video disc, networking, and micrographic, systems areas with applications in the unconventional documentation, information management, and advanced systems design, among other, areas.

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

411586

LB

# SUMMARY

This report covers the period from July 1, 1981 to October 31, 1981; it also serves as a final report for the 1979-1981 incremental contract between Computer Systems Management, Inc. and the Defense Advanced Research Projects Agency for the design, development, demonstration, documentation, and transfer of advanced command and control ($C^2$) computer- and video-based systems. The tasks/objectives and/or purposes of the overall project have been to improve the design, development, demonstration, and transfer of $C^2$ video- and computer-based systems; this report summarizes previous efforts in this general and numerous specific areas. The technical problems addressed are myriad and the general methods employed are eclectic. Technical results include countless recommendations regarding how to augment and improve the design, development, demonstration, documentation, evaluation, and transfer process.

CONTENTS

FIGURES

# 1.0 INTRODUCTION

This report looks at the past two years contractual effort to design, develop, demonstrate, document, and transfer advanced video- and computer-based command and control ($C^2$) information management, decision-making, forecasting, training and readiness systems for the Defense Advanced Research Projects Agency's Cybernetics Technology Office, Cybernetics Technology Division, and System Sciences Division during the period from November 5, 1979 to October 31, 1981. Specifically, the report draws upon and integrates material from the following technical reports produced during the course of this contract. The numbers preceeding each report will be used throughout this report for reference purposes.

[1] James F. Wittmeyer, III, The Development, Demonstration, and Documentation of Advanced Command and Control ($C^2$) Computer-Based Systems. Arlington, Virginia: Computer Systems Management, Inc., 1979;

[2] __________, The Design and Transfer of Advanced Command and Control ($C^2$) Computer-Based Systems. Arlington, Virginia: Computer Systems Management, Inc., 1980;

[3] __________, The Design and Development ($D^2$) of Generic Microcomputer-Based Command and Control ($C^2$) Decision and Forecasting Systems. Arlington, Virginia: Computer Systems Management, Inc., 1980;

[4] __________, The Design, Development, Demonstration, and Transfer of Advanced Command and Control ($C^2$) Computer-Based Systems. Arlington, Virginia: Computer Systems Management, Inc., 1980;

[5] __________, Defense Microcomputing in the 1980s: Problems and Research Priorities. Arlington, Virginia: Computer Systems Management, Inc., 1981;

[6] __________, Microcomputer Software Engineering, Documentation and Evaluation. Arlington, Virginia: Computer Systems Management, Inc., 1981; and

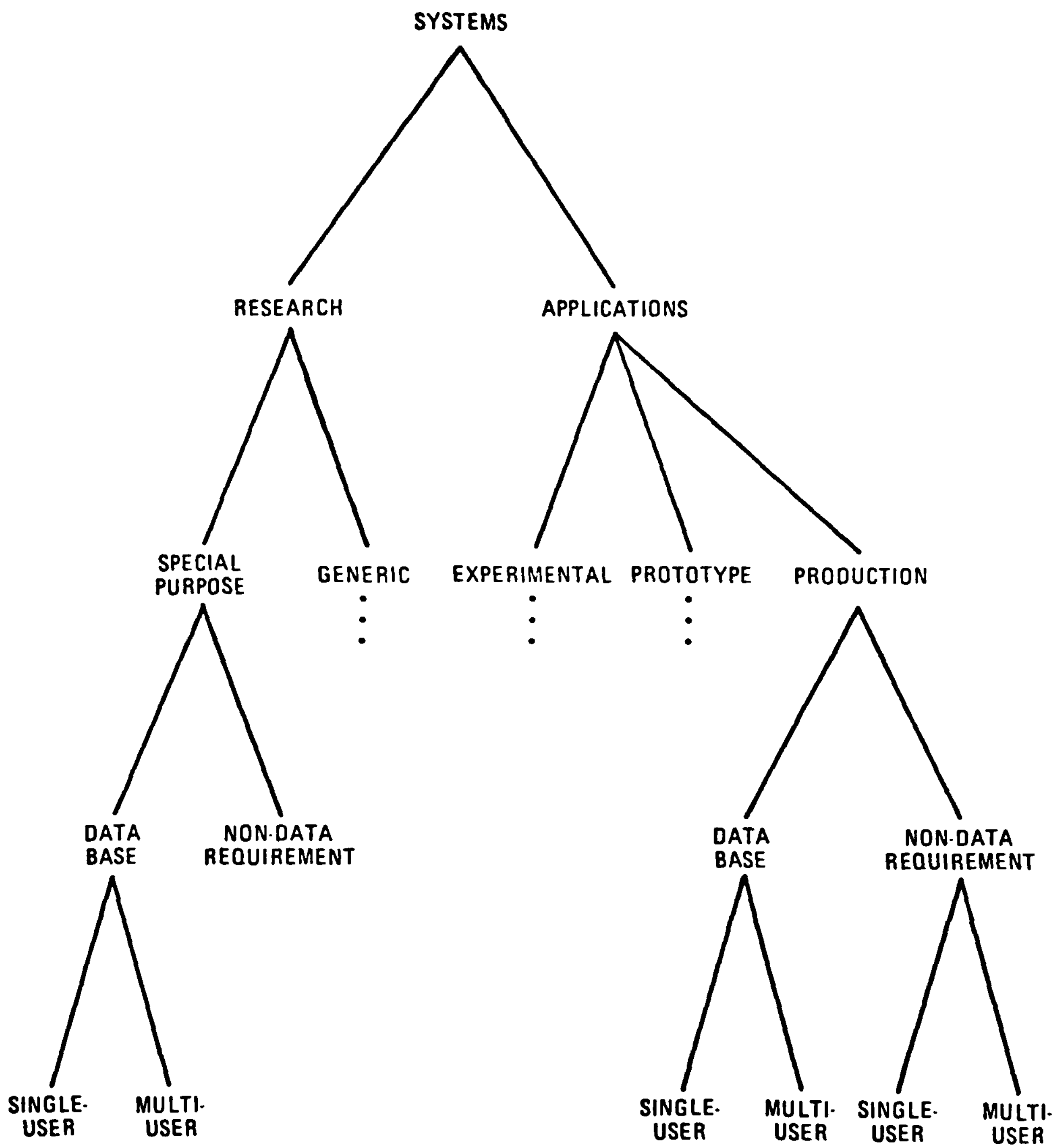
[7] __________, Video-Based Systems Research, Analysis, and Applications Opportunities. Arlington, Virginia: Computer Systems Management, Inc., 1981.

# 2.0 THE DESIGN, DEVELOPMENT, DEMONSTRATION, DOCUMENTATION, AND TRANSFER OF ADVANCED $C^2$ VIDEO- AND COMPUTER-BASED SYSTEMS [1, 2, 3, 4]

## 2.1 The Design and Development of Advanced $C^2$ Computer-Based-Systems

One method for designing computer-based systems requires that the intended system pass through a set of filters as suggested below [2]. 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 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).


SYSTEMS
RESEARCH
APPLICATIONS
SPECIAL PURPOSE
GENERIC
EXPERIMENTAL
PROTOTYPE
PRODUCTION
DATA BASE
NON-DATA REQUIREMENT
DATA BASE
NON-DATA REQUIREMENT
SINGLE-USER
MULTI-USER
SINGLE-USER
MULTI-USER
SINGLE-USER
MULTI-USER

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 single-user (stand alone)? The answer to this question is as important in the design of a system as the other filters.

2.1.1 Disconnects. The application of the above filters will reduce--if not eliminate--many man-machine "disconnects" [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 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 analyses should be performed. Such analyses include, but are not limited to [2, 3, 4]:

- Users;

- Tasks;
- Requirements Analysis;
- Interactive Dialogue;
- Output Devices and Techniques;
- Input Devices and Techniques; and
- Evaluation of System Performance.

Users should be identified and classified as suggested below [2]:

- Naive Users (Inexperienced with computers);
- Managers (Including Military Commanders, etc.); and
- Scientific and Technical.

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 [2]:

- 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;

- "Delphi Technique," a survey technique in which recipient's responses are fed back, anonymously;
- "Policy Capture," one of several techniques for developing quantitative relationships between perceived system desirability and specific system features;
- 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;
- Job analysis techniques, such as task analysis, link analysis, and activity analysis, which attempt to characterize user behavior on the basis of direct observation; 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 <u>interactive dialogue technique</u> is also critical. Some properties of interactive dialogues appear below on an assumed scale [2]:

- Initiative;
- Flexibility;
- Complexity;
- Power;
- Information Load;
- System Response Time; and
- Communication Medium.

Types of interactive dialogue to be selected by the designer appear below [2, 3, 4]:

- Question-and Answer;
- Form-filling;
- Menu Selection;
- Function Keys with Command Language;
- User-initiated Command Language;
- Query Language;
- Natural Language; and
- Interactive Graphics.

The evaluation and selection of output devices is also critical. Some variations are presented below [2, 3]:

- Refreshed CRT;
- Storage Tube CRT;
- Plasma Panel Display;
- Teletypwriter;
- Line Printer;
- Tactile Displays; and
- Large-Screen Displays.

Input devices come next; options appear below [2]:

- Keyboard;

- Lightpen, Lightgun;
- Joystick;
- Graphical Input Tablet;
- Touch Panel;
- Knee Control;
- Thumbwheels, Switches, Potentiometers;
- Tactile Input Devices;
- Automated Speech Recognition;
- Mark Sensing;
- Punched Cards; and
- Touch-Tone Telephone.

The design filters are thus extremely functional; 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.

## 2.2 The Demonstration of Advanced $C^2$ Computer-Based Systems

If one is at all familiar with the difficulty of computer-based systems transfer, one realizes the importance of the role of the demonstration [4]. A new and exciting era has dawned in computer audio-visual techniques via the first production version of the spatial data base management system (SDMS) and other unique display systems which have all contributed to more effective demonstrations.

In order to assure that applied computer-based systems are tailored during the design and development stages, CSM has worked (at the direction of DARPA/CTD/SSD) to evaluate systems development against perceived user needs and requirements. At the general level this has resulted in the application of known and standardized user-oriented programming techniques, including ease of input, menu selection, and effective display, among other features. At the more specific level, attention has been devoted to the particular intended use of the system to be demonstrated.

In an effort to avoid at all cost a demonstration failure resulting from the premature release of a computer-based system, CSM has developed and applied a set of demonstration "readiness criteria." Specifically, the criteria are relevant to the "state-of-the-system" in the following ways: "Has the system been thoroughly checked for all software errors and bugs?" "Has a coherent demonstration sequence been pre-tested?" "Can the user operate the system easily by himself?" "Is the system flexible enough (and, where appropriate, are the data sets extensive enough) to permit a flexible demonstration?" "Have back-up technical and non-technical materials been adequately prepared?" "Have likely questions been anticipated?" "Has the back-up system been thoroughly tested?" "Has supporting (carryaway) documentation (sample-output and user's manuals) been prepared?"

## 2.3 The Documentation of Advanced $C^2$ Computer-Based Systems

Without effective documentation software dies a slow and painful death. Along the way software research progress is encumbered, demonstrations are complicated, and technology transfer is undermined. Interestingly, while the disasterous effects of non-existent or poor documentation are widely verifiable, few are willing to allocate resources aimed at improving documentation techniques. The reason is simple: documentation and documentation research are relatively boring analytical subtasks connected with the potentially exciting design and development of microcomputer-based systems.

At the same time, some effort has been made to define and improve documentation, and given the progress recently made in voice input/output system development, video technology, interactive graphics technology, and computer-controlled microfiche systems development, it is now possible to experiment with the development of several variations of unconventional documentation not possible just five years ago. For example, systems should be programmed to introduce and explain themselves in a manner not unlike that which is used by manufacturing vendors. Such demonstrations could be of invaluable help to those who must convince others that what they have developed may be of real use. Documentation should also be transformed from the inanimate to the animate. Computer-generated system specifications and functional descriptions can be of immense transfer use, as can on-line users manuals. Similarly, films of documentation can also help to bridge the gap between the developer and the user. Here computer-controlled fiche could be used to minimize cost, time delay, and obsolescence. Similarly large screen display systems could be used to present complicated documentation "blueprints" to large audiences and program conversion teams and groups. Self-documentation and automatic flowcharting systems should also be developed. Indeed, the approach now taken by MIT regarding the development of videodisc-based training systems could be used to develop videodisc-based documentation systems [1, 6].

## 2.4 The Transfer of Advanced $C^2$ Computer-Based Systems

If adequate requirements analyses are performed, the transfer process can be greatly improved [2, 4]. Candidly, it is infrequently the case that requirements analyses are performed; instead, most computer-based systems are designed as a function of <u>perceived</u> requirements. Consequently, all too often systems are <u>retrofitted</u> 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.

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.

Following a successful demonstration it is necessary to assess prospects for and difficulties with actual transfer. This generally involves assessments regarding the transferee's hardware and software capabilities. Such assessments enable personnel to tailor and/or modify the system(s) to be transferred in an expeditious manner.

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, initial documentation as well as updated (systems and data) documentation must be provided.

## 2.5 Microcomputer Software Engineering [6]

Ideally before one attempts to build a microcomputer program an effort is made to identify and define the driving functional requirements which together comprise the reason(s) why one attempts to build a problem-solving software system (instead of some other kind of problem-solving system).

At the most basic level are several requirements which are specific context and applications independent; that is, they are relevant to all instances of microcomputer programming regardless of for whom and/or what the software is to be developed.

2.5.1 Response Time. The first is response time. Note that the issue here is not how fast or slowly the system responds to a particular user via-a-via a particular task, but how fast or slowly it responds generally. This kind of speed (or slowness) is a function of the software language used and the microcomputer system I/O device times.

But many operations are non-I/O-oriented, depending instead upon the skill of the programmer and efficiency of the program, which, in turn, depends upon the characteristics of the language used and whether or not the (higher-level) language is compiled or interpreted in operation.

2.5.2 Operating Time. Operating time equals I/O time and processing time. But the processing time is always dependent upon the software languages used, the form of the language, and, of course, the efficiency of the programmer.

2.5.3 Program Status. Another requirement has to do with the status of the program to be developed. Programs which are fundamentally prototypical or experimental usually bear no resemblance to production (systems or applications) programs. Similarly most programs developed as an initial outgrowth of research and development are iterative in their evolution and should therefore be developed differently from programs intended for wide distribution and use.

2.5.4 Support Requirements. Not unrelated to all of the above are support requirements. Is the program to be transferred for on-line use? Or is it to be used off-line by research and development counterparts? Such questions determine to what extent the software must be self-contained, among other considerations.

2.5.5 Microcomputer Software Languages. Response and operating time requirements, the status of the program, and support requirements, among many other conceivable requirements, should determine the selection of a software language. Indeed, a set of guidelines regarding the use of one or more languages should be developed and updated frequently in order to ensure the most prudent and practical use of one or another language. In any case the first task is to understand the relationships among the different types of software.

In addition to such relationships are those which surround the requirements, available capabilities, and optimal language selection. (Note that for the purposes of this exemplar exercise substantive requirements are not suggested since they differ from case to case.) For example,

- _If_ response time and operating time is important _then_ one should, assuming programming competence, use compiler rather than interpretive languages for production systems;
- _If_ a system is by definition iterative _then_ interpretive languages should be utilized; and
- _If_ the talent (capability) exists, _then_ machine and assembly languages should be used to maximize the speed of production systems, and so forth.

The point here is that based upon existing empirical studies it is possible to develop sets of guidelines about the selection of software programming languages against explicit requirements. Such guidelines might even be computerized in a developmental reference system which could be used by research and development managers, programmers, and higher-level decision-makers who must make major software investment decisions. Such a production rule system would make systematic a selection process that is now dominated by preference.

2.5.6 Planning. Structured microcomputer programming is very similar to decision analysis-based problem-solving because it rests upon the principle of problem decomposition. The functions that the program is to perform should inform the decomposition process, and, much like a multi-attribute assessment structure, represent functions decomposed to their smallest component units. In this way programmers can adhere to a simple rule of thumb: software solutions should never be more complex than the problems they are intended to solve.

2.5.7 Software Economy. Programmers should write and collect "speedcode modules" that incorporate all of the basic algorithms which the programmer has previously used. Then the modules should be refined onto different microcomputers in different languages.

In a previous report [3] a design for the development of generic microcomputer-based command and control ($C^2$) decision and forecasting systems was presented which was based in part upon the use of pre-programmed software modules. It was even suggested that the routine $C^2$ decision and forecasting systems functions probably numbered less than twenty-five. If this is true then a series of modules (for retrieving and displaying empirical data, for calculating value, and making inferences, and so forth) could be developed and used over and over again. Similarly, it would be possible to identify and develop modules for information management, training, and generic information display.

2.5.8 Software Psychology. All programming methodology must be applied within a particular personnel context; indeed all of the above presumes the existence of highly talented, dedicated programmers who are as knowledgeable about hardware as they are about software. Unfortunately, virtually every projection available today indicates that throughout the 1980s a critical shortage of programmers will exist. We must therefore maximize the output of those programmers which we do employ. Learning, designing, composition, comprehension, testing, debugging, documentation, and modification capabilities must all be evaluated and improved. Perhaps for the first time, serious programming managers must pay very special attention to the overall programming environment, the components of which include the physical, social and managerial environments.

2.5.9 Software Engineering Guidelines and Recommendations. Requirements analyses should precede programming. Requirements should be matched to software characteristics, and then recommendations regarding how to write the software should be generated. In fact, there is no reason why a production rule system such as RITA could not be used for this purpose. Such a software requirements/software characteristics/programming structure system might be of invaluable use to DARPA researchers specifically and DoD generally, and might function as follows: users could input requirements consisting of operating and response time requirements, program status requirements, support requirements, among any number of other requirements and the computer system, from a knowledge base consisting of software characteristics (updated continually), would then make recommendations regarding optimal programming efficiency in structured pseudocode supplemented by graphic flowcharts of same. It might also suggest the use of pre-programmed software modules about which it has been given detailed information. The information about software form and language characteristics could be consensus "expert" data or data gleaned from empirical experiences with the software; regardless, the system would enable microcomputer programmers to benefit from existing experience with and information about microcomputer software and thereby generate more efficient code. This idea is aimed at supporting the microcomputer programmer; more advanced ideas may very well result in computer generated software in the not too distant future.

## 2.6 Software Evaluation [6]

Several analysts have developed a software characteristics tree which can be converted into a multi-attribute utility (MAU) model for the evaluation of software quality. Like all MAU models it is changeable; nevertheless, we think it is probably very useful as is. Some representative evaluative criteria are as follows:

- Accessibility;
- Accountability;
- Accuracy;
- Augmentability;
- Availability;
- Communicativeness;
- Completeness;
- Conciseness;
- Consistency;
- Efficiency;
- Human Engineering;
- Modifiability;
- Portability;
- Reliability;
- Self-containedness;
- Understandability; and
- Usability.

When these--and other--criteria are arranged hierarchically in a multi-attribute utility model they can be used to evaluate new and existing software as well as the requirements of future software systems.

## 2.7 Video-Based Systems Research Opportunities [7]

Video technologies will in the coming years lend themselves to countless DoD applications. While we have already begun to exploit the technologies for some limited research purposes we have yet to institutionalize any "production" systems. Presented below are thus a series of recommendations regarding how video technologies can be applied to current and future DoD communications and information management problems.

### 2.7.1 Video Disc Applications.

Negroponte [7] suggests a number of unique ways to conceive of video disc-based applications generally. They include the development of personalized, conversational, navigational, and synthesized movies, the use of unique user interfaces (such as touch sensitive displays, talking, and looking), and the development and use of new film techniques (such as variable consecutiveness, motion without movements, sound, still with sound over, and specialized overlays). While all of these ideas are useful and provocative, they are, from a DoD perspective, content-free. More marriages akin to the optical disc/military mapping/training variety must be encouraged. For example, optical video discs can be used to replay films of organizational behavior particularly during crisis management situations. The freeze frame, interactive nature of video disc use would be of enormous benefit during the conduct of previously impossible animated post-mortems. Similarly, personalized movies could be developed from user requirements typologies for training and re-training purposes. Conversational movies present another variation on the same theme, especially for more advanced trainees. Navigational movies could be used for order-of-battle planning and operations, security training, geographical orientation, and even animated simulation development. Synthesized movies, where the parts (data) are not arranged in any predetermined network structure, could be used for advanced weapons design where the barrel of one cannon could be placed upon the platform of another, and so forth and so on. Since possible data permutations and combinations are theoretically unlimited in synthesized movies, radically new designs could be assembled and tested. An example will highlight the process. In aircraft design but a few principles of aerodynamics govern the design and development process, but aerodynamic efficiency remains critical. Over the years many unusual designs have been generated and (experimentally) tested usually in scaled down wind tunnels with scale design models. But imagine an interactive mix-and-match design process supported by video disc-based images arranged and rearranged by a user. Outrageous experimental designs could easily be synthesized from a video disc-based data base on aircraft design.

In many respects the concept of video disc-based synthetic design is analogous to the manual procedure implemented by police artists who "compose" faces of accused felons by mixing and matching noses, eyes, glasses, jaws, hair styles, and the like. Suggested here is the development of a video disc-based system which would accomplish the same end except that it should be endowed with default routines for completely unrealistic combinations and information about optimal design configurations in order to guide and accelerate user behavior on the system.

Video disc systems offer another applications opportunity, however, in the form of enormous storage capabilities. As has already been suggested [7], "juke box" and multiple pack video disc configurations may revolutionize storage media systems development. DoD applications such as personnel recordkeeping, weapons systems capabilities statements, combat readiness/ effectiveness records, alternative battle plans, contingency plans, and even space flight information might all be stored on optical discs available immediately and capable of storing all kinds of data and information. Indeed, where enormous records must be permanently stored <u>and</u> available virtually on-line the multiple disc configuration of the future may be very appropriate and cost effective.

2.7.2 <u>Micrographics Applications</u>. The application of microfilm and microfiche has lagged far behind their capabilities.

In the intelligence field, for example, legal and procedural information about previous crisis management experiences could be easily converted to micrographics. In fact, the DARPA-developed Crisis Management Executive Aids might very well have been developed on fiche. Updating and cost problems would have been alleviated and some users would probably have preferred the computer-controlled fiche system to the system now in use. Similarly, research reports and records could be stored for quick analytical use on fiche and quick response distributed systems would benefit from large film/fiche stations. Treaties and other military and political agreements, which must be stored in exact forms, are also good fiche targets, as are designs, blueprints, proposals, and the like.

The keys, of course, are cost and speed. Film and/or fiche can be produced and duplicated almost immediately and inexpensively, unlike, for example, the production of optical and/or magnetic storage media. Reliability is also important since research suggests that CAR and COM systems are more reliable than magnetic storage systems.

2.7.3 Shared Analysis Systems Applications. There are a variety of ways in which shared local and remote networking and/or teleconferencing systems can be used in DoD. The major applications areas seem to fall in the intelligence and crisis management areas or in any area where group conferring, problem-solving, and/or decision-making is required.

Interestingly there are very few genuinely shared analytical work stations in operation in the intelligence community, save systems which enable analysts to send, receive, and save messages, cables, and reports. Even the Community On-line Information Network System (COINS) is but a shade as capable as an LCN like Ethernet. Only old data bases are shared in any analytical sense and--most importantly--problems are still addressed primarily by individuals who may be only remotely aware of problem-solving colleagues. Proposed here, then, is the development of LCNs for group decision-making, forecasting, and evaluation, among other analytical activities. Since the networking technology *is off-the-shelf* and/or about to be developed new opportunities now exist for improved group problem-solving.

Sometime in the not too distant future the way we all meet to solve personal and professional problems will be dramatically transformed. Instead of hopping on trains and planes and traveling thousands of miles to meet face-to-face to discuss complicated problems, we will find ourselves sitting in front of television screens talking and working in much the same way we now do in person. The procedure which will make this possible is, of course, the teleconference.

In recent years, audiovisual teleconferencing research has taken us to the point where we can successfully link two or more parties who can see and hear each other with relative ease. Some of these teleconferencing systems permit participants to see pictures of each other while others enable participants to converse in full motion.

Full motion audiovisual teleconferencing is now under development at Decisions and Designs, Incorporated (DDI). With DARPA support, DDI is building a system which will enable individuals to meet electronically as though they were all face-to-face sitting around a conference table. Each participant sits in front of as many television screens as there are individuals in the teleconference. Above each of the screens are TV cameras which permit each of the participants to see all of the others. The screens are physically arranged in much the same way individuals might be seated around a conference table. Full motion video also includes overhead cameras at all of the teleconferencing stations which enable the participants to see simultaneously anything one wishes the group to see, such as charts, maps, notes, or pictures.

Each teleconferencing station has been "cabinetized"--that is, made to look like an unobtrusive piece of office furniture. Teleconferencers are thus made to feel comfortable and, hopefully, inclined to use the teleconferencing system with the ease and frequency that they now use the telephone.

The DDI system is now "local." Four teleconferencing stations, comprised of TV monitors, microphones, and cameras, have been connected within the same office building. DDI personnel are thus able to teleconference routinely and eliminate the time and trouble of walking half-way around the building to meet.

As with any good idea, however, there are problems. The most important is cost. While local, or closed circuit, audio-visual teleconferencing is relatively inexpensive, geographically remote teleconferencing is today extremely costly. Geographically dispersed full motion audiovisual teleconferencing stations can cost well over $100,000 a piece, and the necessary satellite communications costs can exceed $1M per year. However, by 1990 these costs will have been dramatically reduced. As more and more satellite circuits become available, as earth stations costs drop, and as the technology necessary for transmitting visual signals grow, we might very well teleconference instead of travel.

Other problems are "social." Just as today many people are unwilling to sit in front of a simple computer terminal, there are likely to be those who will refuse to teleconference primarily because they will see it as an unnatural way to meet.

Others may refuse to teleconference for privacy reasons, suspecting that, unlike during a meeting behind closed doors, too many colleagues or competitors may be listening in. Still others will see the teleconference as a threat to out-of-town treats.

The greatest challenge, however, will not be to enable teleconferees to meet electronically in ways which duplicate face-to-face meetings; instead, the challenge is to build teleconferencing systems which improve upon face-to-face meetings (because face-to-face meetings are notoriously inefficient). In this way we can improve crisis management decision-making and other remote group problem-solving situations.

2.7.4 Video Recording/Playback Applications. The small, lightweight, and easy-to-use recording systems enable us to accelerate the production of optical video discs, film reports and demonstrations, and capture organizational and group behavior for post mortem analysis. Curiously such applications have not occurred with any frequency in DoD.

Of particular applied interest is the use of inexpensive videotape for training purposes. One application [7] successfully interfaced a Sony SLO-320 videocassette recorder to an Apple II microcomputer. Such applications have great implications for DoD training and education, briefings, and demonstrations.

## 3.0 CONCLUSION

All of the above ideas were identified and developed over a two year period of time (1979-1981) and were advanced in consonnance with CSM's management of various GFE mini- and micro-computer systems. While since 1979 CSM's role in the design, development, demonstration, documentation, and transfer of computer- and video-based systems has changed dramatically CSM will continue to discover, develop, and apply advanced ideas to the problems connected with the development and transfer of video- and computer-based systems.