

COMPUTER SCIENCE AND ENGINEERING BOARD MEETING

24-25 February 1971

ATTENDANCE

BOARD MEMBERS

Dr. Walter S. Baer
Dr. Launor Carter
Dr. Joel E. Cohen
Dr. Sidney Fernbach
Dr. Martin Greenberger
Mr. John Griffith
Dr. John Meyer
Mr. Roy Nutt
Dr. John R. Pierce
Dr. Bernhard Romberg
Dr. J. B. Rosser
Dr. Alan Westin
Dr. Ronald Wigington

Dr. A. G. Oettinger, Chairman

Representatives

Mr. Michael Baker, (Westin)
Dr. Michael Feder, (Haddad)

OBSERVERS

Col. Andrew Aines, Technical Assistant,
Office of Science and Technology

Mr. Donald Burns, U.S. Government

Dr. Ruth Davis, Director, Center for
Computer Sciences and Technology,
National Bureau of Standards

Dr. John Egan, Senior Staff Specialist,
DDR&E/OAD/Intelligence

Dr. Bruce Gilchrist, Exec. Director
American Federation of Information
Processing Societies

ATTENDING

24-25 February
24-25 February
24-25 February
24-25 February
24-25 February
24-25 February
25 February
24-25 February
24-25 February
24-25 February
24-25 February
24 February
24 February

24-25 February

25 February
24-25 February

ATTENDING

25 February

25 February

25 February

25 February

25 February

ATTENDANCE (cont.)

OBSERVERS (cont.)

ATTENDING

Mr. Ken Hunter, U.S. General Accounting Office	25 February
Mr. William Knox, Director, National Technical Information Services	25 February
Miss Ann Marie Lamb, Management Analyst, ADP Management Staff, Bureau of the Budget	25 February
Mr. Lawrence G. Livingston, Council on Library Resources	25 February
Dr. John Pasta, Head, Office of Computing Activities, National Science Foundation	25 February
Dr. Charles V. L. Smith, Division of Research, Atomic Energy Commission	25 February
Mr. Bernard Urban, Director, Urban Clearing House Service, Department of Housing and Urban Development	25 February

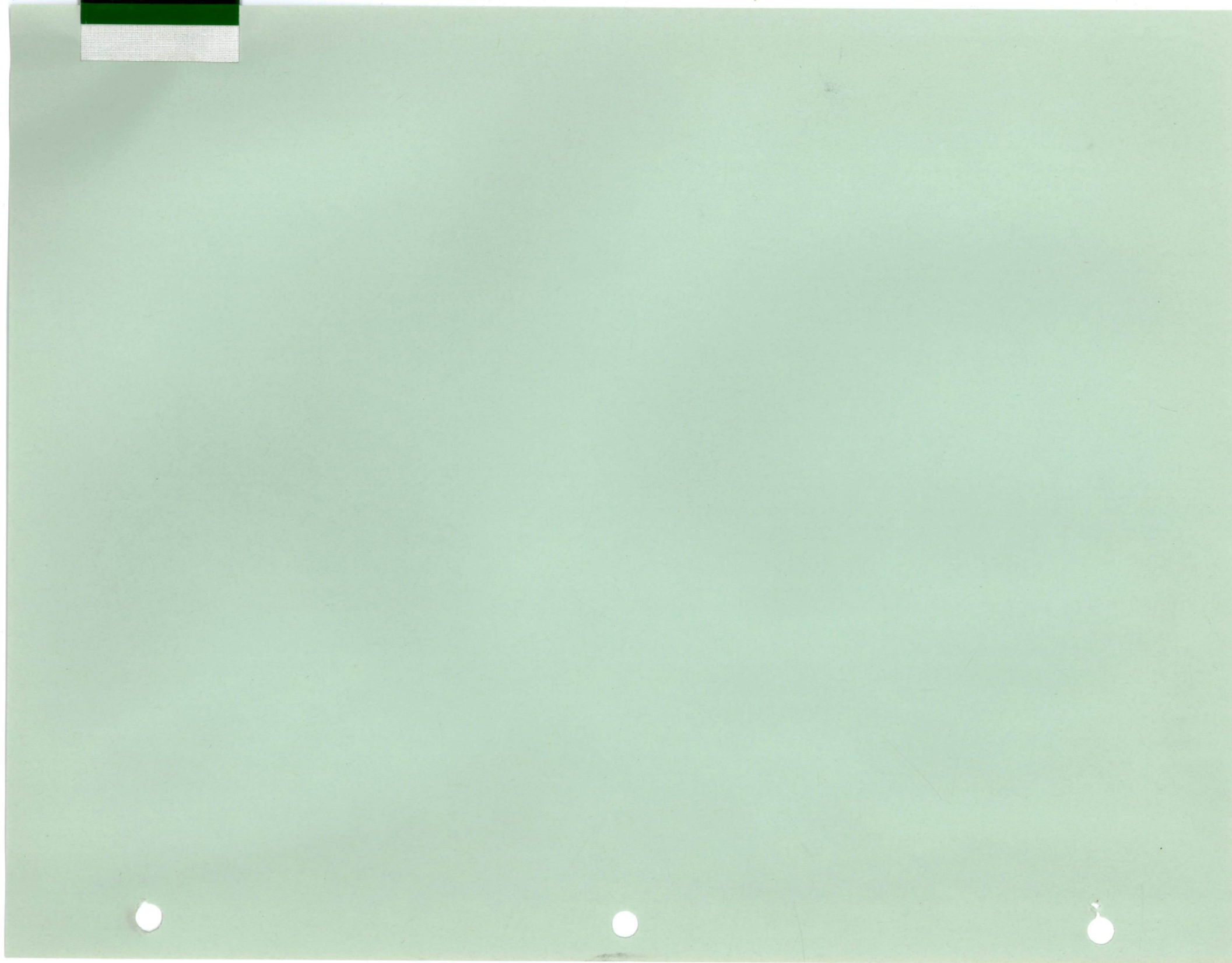
GUEST SPEAKERS

Walter Carlson, President, Association for
Computing Machinery

Robert W. Wngles, Systems Development
Division, IBM Corporation

Elmer Galbi, IBM Corporation

Evening Session



NATIONAL ACADEMY OF SCIENCES

COMPUTER SCIENCE & ENGINEERING BOARD
2101 CONSTITUTION AVENUE
WASHINGTON, D. C. 20418

COMPUTER SCIENCE AND ENGINEERING BOARD

AGENDA

Evening Session 24 February 1971

The evening session of the Computer Science and Engineering Board Meeting will be held in Room 600A of the Joseph Henry Building and begins at 6:30 p.m.

AGENDA ITEMS

NOTES FOR ACTION TAKEN OR PLANNED

1830 - 2050

(A) CS&EB Planning Operations

1. CS&EB Committees - Policy
2. Thrusts, Projects, and Funding
 - a) Computers in Medical Care
 - b) Public Computing Facilities
 - c) Software Short of the Carrier Interface
 - d) National Center/Data Banks/Westin on-go
 - e) Information Systems Projection
 - f) Impact of Computers on National Economy
- Productivity and Flexibility
3. Board Composition, Rotation and Nominations

The Chairman
EXCOM

2050 - 2100

(B) Report Reviewer Nominations/Information Systems

The Chairman

2100 - 2115

(C) Privacy & Data Banks/Forward & Preface

The Chairman

2115 - 2135

(D) Communications Between AFIPS and CS&EB

M. Feder

2135 - 2200

(E) NASA-ARPA-Illiac IV (Ames Research Center)

The Chairman

ADDENDA

Evening Agenda

2200 - 2220

(F) Computer Education (Perlis) Report
Disposition

The Chairman

NATIONAL ACADEMY OF SCIENCES
2101 CONSTITUTION AVENUE
WASHINGTON, D. C., 20418

0900 7-
FEB 8 RECD

3 February 1971

ANTHONY G. OETTINGER, CHAIRMAN
COMPUTER SCIENCE & ENGINEERING BOARD
AIKEN COMPUTATION LABORATORY
HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS 02138

TO: S. Fernbach
W. F. Miller
J. R. Pierce
R. Wigington

Gentlemen:

The renewal of the ARPA contract gives us a breathing spell, but obviously does not solve our long-term funding problem.

Our discussion of the role of the Executive Committee stressed its responsibility for devising and recommending a program and a set of priorities for the Board but did not explicitly address itself to finance. Obviously, however, the two problems are very closely related.

In the past, especially in connection with our first private fund drive, Jerry Haddad served as a kind of finance committee. Perhaps we need a more formal mechanism involving additional Board members to help think through the relation between our program and our finances and to play an active role in carrying out a fund solicitation program.

I'd appreciate it if Jerry Haddad could give this some concentrated thought and plan some time before the February meeting to call together all the addressees and recipients of copies of this memo in a conference call at which we might discuss the matter further with an eye toward having a structured debate and, I hope, some action decisions made at our February evening executive meeting.

I very much look forward to your thought, advice and action on this crucial question.

Sincerely yours,

Anthony G. Oettinger

AGO:chm

cc: J. Griffith
J. Haddad
W. C. House

A2b, c

A26
Feb 11 by Foll

29 January 1971

TO:: Warren C. House

Dear Warren,

Here are some additional items for the Board shopping list that I started in a memo a few days ago.

1. The 2 or 3 page note that Jerry Hallad prepared should be distributed to the Board and appended to the shopping list.
2. Way back in the planning group days it was suggested that the Board might usefully undertake an analysis of one or more critical computer systems in terms of the impact of current practices, personnel, etc. on life, limb, or property. The Westin study, in one sense, falls in this category. Might we not, however, usefully consider the more technical aspects of hardware and software in things like air traffic control, etc?
3. There seems to be a recurrent problem exemplified in the CLR study in educational technology and other realms concerning the circularity of relating technological possibility to need and to demand where each depends on the other and none can be pinned down without knowledge of the others. Can the Board usefully stimulate an analysis of, what I shall call for want of a better word, "marketing" at the stage intermediate between the total absence of such considerations in basic research at universities, and shorter term analyses based on commercial perception of existing or relatively easily stimulated demand?

Sincerely yours,

✓ Anthony G. Oettinger

AGO:chm

cc: S. Fernbach
W. F. Miller
J. R. Pierce
R. Wigington

HADDAD'S COMMENTARY

The role of computers in our society is a new and rapidly changing thing. There are a number of aspects in this relationship.

1. How computers can help or hinder or at least affect our society, or,
The impact of computers on our society with regard to changing patterns of structure, etc.
2. How the needs of our society should affect the thrust of computer development and computer application development, or
The priorities that should be developed in the various areas of development of computers and their applications.
3. How our societal institutions such as colleges, universities, and government should accommodate the above, or
How our societal institutions can best understand and accommodate the effects of computers on our society.

This important set of interfaces must be studied and understood not only (a) by informed and expert technical computer people, but also by (b) informed and expert people in the fields and disciplines which are directly involved in these interfaces and (c) by broad-gauge generalists who are the concerned operators and decision makers^{re} who determine or guide our social responses to complex issues, such as technology utilization. This is necessary so that the average citizen can better understand the effects of these machines on his daily life, and so that technicians can better adapt their work to reflect

Page Two
Haddad's Commentary
16 December 1970

understanding of the social implications of their work; and so that our broad gauge social leaders can guide our response.

This is the primary role that the board should play. This is the thrust that should determine its priorities, its projects, its membership, its organization, and its outside relationships.

The board should not concern itself with issues that are self-serving or issues that are unique to the computer field unless these are crucial issues or unless specifically requested by the U.S. government. Given the development of the computer field academically, industrially, and commercially to date, it is expected that crucial issues would call for Board initiative often. The Board should emphasize those interfaces that represent the relations between computers and their larger social impact.

09007-
JAN 29 RECD

NATIONAL ACADEMY OF SCIENCES

2101 CONSTITUTION AVENUE
WASHINGTON, D. C., 20418

26 January 1971

ANTHONY G. OETTINGER, CHAIRMAN
COMPUTER SCIENCE & ENGINEERING BOARD
AIKEN COMPUTATION LABORATORY
HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS 02138

TO: W. C. House

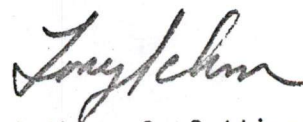
Dear Warren,

I think it might be well to keep a list of broad and specific issues that may come to the attention of Board members for continuing consideration by the Executive Committee as it formulates our program and priorities. We might issue the first one at the February meeting and ask Board members to contribute for further review by the Executive Committee.

I have the following two items:

1. The problem of software on the computer side of the computer/-communications interface is something that should concern us as much as the other side of the interface. This may be part of the generic problem of data bases and data management, of coupling systems to each other, or pieces of systems together. What, if anything, do we do about it?
2. Looking to the various constituencies the Board might consider itself responsible to, should we not study the question of computer accessibility for minorities, or for that matter individuals in general. One might look at this under the heading of desirability and necessity of "public computers" by analogy to public libraries.

Sincerely yours,



Anthony G. Oettinger

AGO:chm

cc: S. Fernbach
W. F. Miller
J. R. Pierce
R. Wigington

A2d

DISCUSSION DRAFT

Proposal to Create a National Center on Computer Data Banks

Computer Science and Engineering Board

National Academy of Sciences

January, 1971

Introduction

The development of computerized record systems is a major trend of our times. What has come to be called the "computer data bank" is appearing in a wide variety of government agencies, covering fields such as science, law enforcement, health, welfare, education, taxation, motor vehicles, planning, and many more. New, multi-agency data banks in a particular subject matter field, spanning various levels of government, are also emerging, illustrated by the New York State Identification and Intelligence System, which collects, processes, and distributes information for 3600 municipal, county, regional and state criminal justice agencies in New York State. In addition, administrative data processing centers holding information from various departments at a particular governmental level (city, county, or region) have developed during the last few years.

A parallel development has taken place in the private organizational structure of the United States. Computerized record systems are to be found in almost every area in which records on people are held by large organizations. Data banks are being installed by banks, insurance companies, religious bodies, labor unions, educational and scientific institutions,

credit card companies, reservation systems, service agencies, information suppliers, hospitals, and many more.

While the creation of such computerized record systems has been hailed as a means of reshaping the information function at a time of increasing data growth and complexity in organizational decision making, concern over the impact of computerized record systems on civil liberties interests of the citizens has become a major public policy question. The issue of invasion of privacy, along with the problem of assuring the citizen rights of access, challenge, and correction of information in computerized files (often called the "due process" issue) has been raised at a series of major congressional hearings during the late 1960s, by national television programs, in meetings of a wide variety of professional and civic associations, and has given rise to a growing literature in journals of technology, law, social science, and public policy.

The fact that this is not simply a concern of a few persons overstimulated by reading George Orwell's writings is shown by two recent opinion studies. In an article published in August 1970, the Louis Harris association reported the results of a nationwide cross section sample inquiry on the topic of invasion of privacy.¹ The survey showed that one in three Americans (34%) feel that their privacy is being invaded. Concern over "computers which collect a lot of information about you" was mentioned by one in five respondents as a matter of specific concern to them, the highest ranking of the specific violations mentioned. Demographic analysis

Page Three
National Center
17 February 1971

showed that those worried about computer data banks invading their privacy were from the more affluent sections of the population, the better educated, and those living in the suburbs.

A second recent report comes from a survey of attitudes toward technology conducted by a social science consulting firm for the Harvard Program on Technology and Society.² The inquiry was conducted in three towns in the Boston area representing different social class and urban-rural mixes, and was based on depth interviews with 201 persons. In this sample, the investigators found that 42.8% feel that government knows too much about their personal lives. 55.7% specifically oppose the creation of computerized data banks by government, with invasion of privacy as the main reason for such opposition.

It was in recognition of this strong public concern over the issues of privacy and due process in computerized data banks that the Computer Science and Engineering Board of the National Academy of Sciences proposed to Russell Sage Foundation in 1969 an empirical study of trends in the computerization of record systems containing information about individuals. The result was the Project on Computer Data Banks, currently in the last six months of its two and one-half year life. (Its report is due to be issued in June 1971, followed by a longer monograph by the Director to be published early in 1972.) This Project has looked at the entire spectrum of government and private data banks, collecting published and unpublished literature, conducting 60 detailed site visits to leading government and

private data banks, and administering a survey to 2600 organizations asking about basic trends in record automation. (A press release describing the organization, staff, and scope of the study is attached.)

It is hoped that the report of the Project, and the monograph that will follow, will help to provide fundamental information and policy options needed by legislators, administrators, scientists, social scientists, the media and the public on the data bank issue. However, it has already become clear to those associated with the Project that a one-time project of relatively narrow scope, however useful its findings, cannot serve American leaders and the American public as a steady monitoring instrument to keep abreast of the computer's role in reshaping organizational record keeping and decision making about people in American society. Hence, this proposal.

I. The Rationale for a Continuing Center on Computer Data Banks in America

Most thoughtful commentators agree that ours is an age in which wise public policy cannot allow new scientific technology to proliferate and radiate its effects before society is really aware of the consequences. In a society as complex and interconnected as the one we are entering, this is especially true of a technology that is not merely used in production and research but may be thought of as the control technology of an organizationally dominated social system. It is estimated that there are 76,812 computers in the United States today³ with a value of \$25.4 billion dollars. The annual "ADP costs" for operating these systems in 1970 has been estimated at \$23.2 billion dollars. Some 1.4 million man years are being used to conduct these

Page Five
National Center
17 February 1971

computerized activities. In the federal government itself, computers have grown from 531 in 1960 to 2,412 in 1965 to 4,756 in 1971. Federal ADP costs in 1970 are estimated at \$2.2 billion dollars, using an estimated 136,000 man years per year. As these growth figures about computer proliferation indicate, computers are being installed throughout the organizational framework of the nation. It is not too much to say that they spring up across the organizational landscape like so many mushrooms in a favorable damp climate. What is equally significant is that each such computer system is a separate spore-bearing entity. It has highly differentiated hardware, software, and terminal linkages; its operations are based on laws, administrative regulations and practices distinctive to a given area of science, industry or government; it even reflects the philosophy and style of particular departments and agencies within the larger organization. In a field such as law enforcement alone, with each city, county, and state engaging in its own development of computerized information systems, there are hundreds of diverse computerized record systems and data banks being installed or already in operation across the country.

A basic need of American society at this point, we believe, is to create some kind of continuous monitoring and reporting system, whose credibility and objectivity will be widely respected throughout our society, to provide basic information on these steadily expanding data bank developments. With an estimated billion dollars being spent annually on the creation of such data banks, it is imperative that a small fraction of that amount be

found to provide a monitoring instrument for American society - in a sense, a data bank on the data banks.

II. The Proposal: A National Center on Data Banks

To respond to the need just described, it is proposed that a National Center on Computerized Data Banks be created. While the exact form and functions of this Center will need to be explored carefully in discussions with advisors and potential funding sources, this memo will seek to suggest the design, administration, and possible funding of such a Center.

A. Scope

The Center would be designed to collect, process, and distribute information relating to government data banks at the local, state, and federal levels, plus a selected group of private data banks containing particularly important and sensitive information about individuals. It would limit itself to non-classified systems, to avoid problems of clearances, restricted data, and related matters. It would be possible to include within the system, for comparative purpose, information about selected manual record systems in which large and important bodies of data were maintained about individuals.

B. Sources

The sources for the National Center would start with a deposit of the extensive literature collection, non-privileged material from site visit reports, and other products of the Computer Data Bank Project, thus insuring an initial start with the fruits of the largest research effort in this field to date. Following this initial deposit, the Center would

then solicit directly the systems plans and reports of data banks in government agencies and the selected private systems. Since the government systems are funded with public monies, the reports and plans they issue are usually public documents. In addition, legislative hearings at which monies are requested for such systems, and the reports of the various grants-in-aid programs that support the development of many government data banks (such as Law Enforcement Assistance Administration, Housing and Urban Development and Social Security Administration programs in these areas) are a primary source of description and detail. The Center would also collect published literature, conference papers, and other report materials. To these would be added various periodic questionnaires that would be drawn up by the Center to collect factual information and trend data directly from data bank managers.

A program of selected site visits by the staff and consultants of the Center would provide for the testing of analytic categories, the development of case studies, and improvement of the reporting instruments used by the Center staff.

C. Design of the Center

The Center would be designed to be an information retrieval system geared to producing various lists, abstracts, and reports to (1) a broad user community, for any uses they desired, and (2) the Center staff, for analytical and reporting purposes to the national public. The user community would include scientists, legislators, administrators, regulatory agencies,

system designers, computer firms, organizations considering the computerization of records, technical specialists, social scientists, lawyers, the press, civil liberties and consumer groups, and many others. It would also be designed to provide a means of monitoring, of reporting exceptional developments, and of generating material for survey reports by the Center's staff. It is envisaged that the system would begin as a batch processing operation, with provision for evolution to a real-time system to facilitate the most effective user and staff inquiries of the data base.

D. Output of the Center

The following are conceived to be the products and services that the Center would provide:

1. Lists of Data Banks. The Center could provide a total list of data banks covered, by the name of the agency and the type of data bank. It would also provide lists ordered according to distinguishing characteristics. These would include lists by type of owning entity, functions of the data bank, the type of equipment used, the type of data stored, the safeguards in place or projected, by additional data or new functions being taken on by the system at later dates, and similar matters. A system of providing periodic lists of new startups of data banks, or newly announced plans, would also be feasible.

2. Short Abstracts. A 500-1,000 word abstract of each system, retrievable either for the total data base or on a selective basis by category would be a capability of the system. These abstracts would be

formulated according to critical descriptive elements set up by the Center's staff, and would be verified as to accuracy by sending back to each organization that had created a data bank the text of the abstract either for factual corrections or for explanatory commentary, when that would be appropriate.

3. Longer Descriptions. A 5,000 word narrative description for each data bank system would also be prepared, verified by the owner or operator of the data bank, and retrieved according to the various categories mentioned above. An example of such a description is included herewith. While this is not the exact form or content that the proposed Center would adopt, this description done for the Department of Housing and Urban Development by Systems Development Corporation will illustrate the type of report that is envisaged.

4. Cross-System Data. Reports as to the presence and nature of various technological, legal, administrative, and other measures and practices designed to deal with data confidentiality and individual access to its records in the various data banks contained in the National Center could be retrieved by users and the Center's staff.

5. Data Bank Literature. Lists would be produced by the Center providing either a basic citation or a short, Center-prepared abstract of all significant books and articles published on the issue of computer data banks; unpublished papers and research reports; literature about leading data bank developers; government reports and hearings; and relevant statutes, regulatory rules, and administrative regulations.

A second major output of the Center would be a series of monitoring reports that would be based upon categories and criteria created by the Center staff, in consultation with its advisory group. The concept here is that a set of indicators would be devised which registered such things as the degree to which fundamental issues in the protection of privacy and provision for due process had or had not been provided for in the data banks about whom information was being put into the system. When new systems, which did not have such safeguards, were entered into the data base, exception reporting could be used to report that fact, with the result that a variety of standing queries would be possible by government and private observers relying on the system for such response. In addition, there would be statistical trend analysis by various areas, accomplished in ways that only a computerized system of data analysis would make possible for such a large number of organizational record systems as this Center would soon have. While it is difficult to estimate how many computerized record systems would be recorded in the system at various stages of its growth, the experience of the Project on Computer Data Banks leads to the estimate that an initial pool would probably be near a thousand and could increase to several thousand within the first few years of the Center's operation.

A third output of the Center would be special reports done by the Center staff, reporting on trends for data banks as a whole, by various subject matter fields, by levels of government, and other matters that would be of high interest to the user community and to the nation.

Finally, it is envisaged that there would be a biennial report from the Center, designed to be the authoritative volume of fact and commentary on the development of computerized record systems involving individual information in the United States.

E. Sponsorship

It would seem inadvisable for such a national center on data banks to be located within a government agency, almost regardless of which agency one might select. Since the development of public confidence in the objectivity of the administration of the data bank would be of prime importance, and since it is governmental systems themselves that make up the primary object of data collection by the Center, it is believed that the National Center ought to be a private agency (though it could well rely on a mixture of government, foundation, and private funding for its basic support). Ideally, the Center would be located in Washington, and have as its staff a group of established scholars covering the various technical, legal, and social science fields necessary for the design and maintenance of such a wide-ranging enterprise. The charter of the Center should stipulate its commitment to an empirical focus, to the greatest neutrality and objectivity consistent with the obvious need for conscious analysis and categorization, and to the provision of access to the Center's data by users of all kinds, whether persons supportive or critical of data banks. Access for users could be by subscription (to recover partial costs), with provisions for low cost or free access to those unable to pay such costs.

It has long been an ideal of those thinking about the relationship of science and computer technology to democratic society that we should begin devoting both dollars and energy to the design of computer systems to be used by civic groups and the public directly, rather than having computers used solely by and for large organizations. Particularly where monitoring the effects of computer technology itself is involved, in the especially sensitive area of the effect of computerized record systems on the citizen's liberty, there is a vital social need to experiment along such lines in a way this National Center would provide. This is an important reason why private rather than public operation of the system would be critical.

It is worth noting that the Project on Computer Data Banks of the Computer Science and Engineering Board was able to obtain access to all the governmental agencies and private organizations that it chose to contact for its detailed site visits. In addition, the Project staff received extensive cooperation from computer manufacturers, software houses, system developers, congressional and state investigating committees, civil liberties and civil rights groups, client organizations, consumer groups, and professional associations. Such contacts could be incorporated into the early stages of the National Center's activity, and would save many man-years of work that would otherwise have to go into developing such contacts and information sources.

F. Administration

The administration of the Center would be set up along the following lines. There would be a director, someone with a national reputation in the

area of data banks and the public policy issues posed by them, and whose name would help insure the seriousness of the endeavor. An advisory group of approximately twelve persons would be selected, drawn from science, government, social science, industry, the computer field, minority groups, consumer groups, civil liberties organizations, and various other sectors of society, to provide supervision and strengthen public confidence in the integrity of the National Center.

An operating committee, along with the director, would determine the basic analytical categories and guidelines of the operation. This committee would consist of approximately seven persons distributed among the fields of science, computer science, law, public administration, and the social sciences. The director and operating committee would be distinguished academics, serving on a part-time basis. Such a format would make possible the recruitment of more outstanding persons than would be likely if full time service were required, and follows the pattern utilized in the Project on Computer Data Banks.

The operating staff would be full-time employees. They would consist of an administrative assistant to the director; a technical director; an abstracts staff of three persons (distributed among law, the social sciences, and organizational administration); a data analysis staff of three persons; one librarian - publications specialist, and four secretaries. Special services would be obtained as needed on a consulting basis.

III. Funding of the Center and the Timing of Its Creation

To bring such a Center into existence, to insure the collection of a first class staff, to establish its place as a working institution, and to make sure that its first period of operation had the necessary continuity, it is suggested that the Center be created with an initial three year grant. This grant is estimated at \$375,000 - 500,000 per year, a figure which includes overhead but is exclusive of starting costs

A. Operating Costs

The operating costs of the Center would include the usual items - salaries, information preparation and programming costs, computer time, materials acquisition, mailing and questionnaire expenses, travel and site visit costs, equipment, and cost of preparation for special reports and the biennial volume. Income from users, inquirers, the sale of reports, and sale of the biennial volume would reduce the cost of the operation as the Center moved into mature life, and the cost of the system to be funded in its last year of the three-year initial period might well become less as these services made contributions to its cost of operation.

It is submitted that an ideal way to fund the Center would be through a combination of grants from federal agencies, private foundations, and industry.

B. Plan of Action

In order to develop a more explicit statement of the goals and objectives of the Center, and to prepare a careful estimate of the costs involved, it is proposed that a six-month study be undertaken in order to

prepare a detailed proposal for establishing and operating the Center.

The six-month study would have the following general objectives:

a) Prepare a more detailed statement of the input, content, and output of the Center. This would probably be accomplished through consultation of the personnel of the Project on Computer Data Banks with potential users in the private and public sectors, consultants, and others concerned with the utility of the Center.

b) Prepare an estimate of the kinds and types of services, publications, reports, and lists to be offered by the Center. This would be a secondary part of the activity in (a) above.

c) Prepare a detailed estimate of the costs, expenses, resources and time needed to establish the Center as an operating entity. This estimate should include costs of space, equipment, personnel, and also the conversion of data from its present printed form to that of a mechanized data base. This estimate should also include any software conversion costs required as part of the startup.

d) Prepare a detailed estimate of the costs and resources needed to operate the Center as an operating entity. This estimate should include any foreseen plans for growth in size of data base, speed of system response, or provision of additional services beyond those of the initial installation.

e) Prepare a suggested organization of the Center with estimates of numbers of required personnel and skill mix needed for startup and follow-on operation.

This study will compare the costs and suitability of computer service bureaus with a dedicated equipment installation.

It is assumed here that this study would be undertaken by, or aided by, a small group of expert systems analysts such as can be found in consulting firms or computer manufacturing firms.

It is anticipated that this study will yield a proposal of the "Turn-Key" variety in which all necessary items of space, equipment, and personnel will be included so that the picture of startup and operating facilities will be complete in every respect.

Conclusion

One of the special problems of balancing the need of organizations for more effective use of data with the deepening concern of citizens for protection of their rights to privacy and due process is the tremendous fragmentation of information collection and utilization in American society. Ironically, if we had a few omnibus agencies that maintained most of the information on citizens we would be, in that regard, in a better position to know what was happening and to install and maintain the safeguards that public policy might articulate. But we are now at a moment at which hundreds of computer data banks are spreading rapidly throughout American society, with enormously varied rules and effects. Such a situation calls for some kind of technological assessment and monitoring system. If we are to use the tools of technology to help democratic society bring the forces of technology under effective public policy controls, this Center would seem to be a major arena for such experimentation.

FOOTNOTES

1. Louis Harris, "Invasion of Privacy Worries 34%", Washington Post, August 3, 1970.
2. Public Views of Technology, A Report to the Harvard Program on Technology and Society, by Social Systems Analysts, October, 1970, Mimeo. Discussion of "Invasion of Privacy" appears at pages 16-17 of this preliminary document.
3. Sources relied on for these figures, both firm and extrapolated, are General Services Administration, "Inventory of Automatic Data Processing Equipment in the United States Government, Fiscal Year 1969," and Pat McGovern, "EDP Industry Report," December 15, 1970.

PROCEDURES AND COST ESTIMATES FOR PHASE I

Phase I, planning and development of the project, would have three main objectives:

OBJECTIVE 1: To sort, catalogue, and prepare for data conversion voluminous materials collected by the Project on Computer Data Banks during its 2½ year life. A byproduct of this enterprise would be a definitive annotated bibliography of technical, legal, social science and popular literature on the computer data bank issue, major entries would be abstracted for maximum usefulness by those consulting the bibliography. This would be completed in December 1971 and published in 1972.

OBJECTIVE 2: To prevent an interruption of the comprehensive monitoring of data bank development now being conducted by the present project. Averting such an interruption will require a basic staff to continue collection and analysis of organizational reports, government documents, technical reports, and other commentaries until the initiation of the National Center projected for January 1972.

OBJECTIVE 3: To develop a comprehensive plan for the National Center. This would specify the types and classes of computerized and manual record systems to be covered; the data sources to be used; basic users' services (as described earlier in this proposal); monitoring and reporting functions to be conducted by the National Center staff; and basic system design. Complete hardware and software requirements and costs covering progressive development from paper files to a real time computer system would be developed in this plan.

To pursue these objectives, a core group from the Project on Computer Data Banks (Westin, Baker and Hoffman) plus research assistants, supporting

staff, a consultant on abstracting and a consultant on software planning would conduct the first phase of work from July 1, 1971 until September 30, 1971. Westin would be able to devote full time to this and Baker and Hoffman half time to this. This group would carry forward Objectives 1 and 2, and would develop initial documents on Objective 3, the comprehensive plans for the Center.

This plan (produced by September 30, 1971) would then be given to an independent systems consulting firm to conduct a thorough review and independent evaluation of the objectives, procedures, and development program drawn up by the planning group. A two month study by these consultants would take place during October and November, 1971 resulting in an evaluative report by the consulting firm on November 30, 1971. This report would also include a set of specific recommendations for data conversion, EDP services, and system development. During these two months, the planning staff would be available for meetings with the consultants, while the staff would also be continuing the monitoring of data bank development and preparation of materials for future conversion into the computerized system.

In the final month, December 1971, the planning staff would study the consultant's report and draw up a final document specifying all aspects necessary to bring the National Center into initial operation. These would be presented to the Computer Science and Engineering Board of the National Academy of Science for review and commentary.

Throughout the six month life of this study, with the aid of the documents produced at each stage, conversation would be going forward with possible funding sources and users of the National Center among government agencies, private industry, professional groups and private foundations.

PRELIMINARY BUDGET

July 1, 1971 - December 31, 1971

Honoraria for Senior Staff		\$ 17,850
Westin, Baker and Hoffman		
Research Assistants (2)		6,200
Consultant on abstracting and data format		3,600
Consultant on software planning		4,000
Administrative Secretary (Washington)		4,800
File Secretary and Document Specialist (New York)		4,400
Typing Services		1,500
Travel (staff meetings, conferences with consultants)		3,000
Communication Services (telephone, postage)		3,000
Materials and Services (xerox, printing, supplies)		3,500
Subscriptions, purchase of reports and materials		2,000
		<hr/>
		\$ 53,850
Two month contract to consulting firm		20,000
		<hr/>
		\$ 73,850
Probable Overhead to NAS		
30% on \$53,850	16,200	
10% on \$20,000	2,000	
Overhead Total		<hr/>
		\$ 18,200
Estimated Total Cost of Project		<hr/>
		\$ 92,000

A2f

NATIONAL ACADEMY OF SCIENCES

2101 CONSTITUTION AVENUE

WASHINGTON, D. C., 20418

25 January 1971

ANTHONY G. OETTINGER, CHAIRMAN
COMPUTER SCIENCE & ENGINEERING BOARD
AIKEN COMPUTATION LABORATORY
HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS 02138


TO: S. Fernbach
J. Griffith
~~W. C. House~~

W. Miller
J. Pierce
R. Wigington

The question of the impact of Computers on Organizational Structure and Decision Making continues to be important as a practical matter and seems to be receiving increasing scholarly attention.

Would someone in that area be useful on the Board? Would you have a look at the enclosed list and the paper by Orden, search your own memories and advise me. Many thanks.

Sincerely yours,


Anthony G. Oettinger

AGO:chm

enclosures

Invitees to April Conference on Information Technology in
Government Decision-Making

Dean Edward Bok
Inter-University Case Program
Syracuse University
Syracuse, New York 13210

Dean Harvey Brooks
Harvard University
Division of Engineering and
Applied Physics
Pierce Hall 218
Cambridge, Mass. 02138

Dean Alan Campbell
Maxwell Graduate School for
Citizenship & Public Affairs
Syracuse University
Syracuse, New York 13210

Mr. Anthony Downs
Real Estate Research Corporation
72 West Adams Street
Chicago, Ill. 60603

Mr. John Griffith
IBM Thomas J. Watson Research
Center
P.O. Box 218
Yorktown Heights, New York 10598

Mr. Bertram Gross
Department of Urban Affairs
Hunter College
695 Park Avenue
New York, New York 10021

Mr. Charles Haar
Harvard University School of Law
Faculty Office Building
Massachusetts Avenue
Cambridge, Mass. 02138

Mr. Richard Neustadt
Director, Public Policy Program
Harvard University
Littauer Center 127
Cambridge, Mass. 02138

Mr. Anthony G. Oettinger
Harvard University
Aiken Computation Lab. 200
Cambridge, Mass. 02138

Mr. George Rathjens
Department of Political Science
M.I.T. E 53-434
Cambridge, Mass. 02139

Mr. Herbert Simon, Dean
Department of Industrial
Administration
The Carnegie-Mellon University
Pittsburgh, Pa. 15213

Professor Aaron Wildavsky
Graduate School of Public Affairs
University of California
Berkeley, Cal. 94720

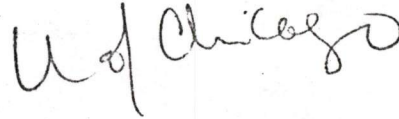
Professor Harold Wilensky
Department of Sociology
University of California
Berkeley, Calif. 94720

"ORGANIZATIONAL
INTELLIGENCE"

NSA
BUREAUCRACY
Written
while he
was at
(RAND)

INFORMATION STRUCTURE MODELING OF ORGANIZATIONS

A. Orden

A handwritten signature in cursive script, appearing to read "A. Orden", located below the printed name.Preface

The phrase "electronic data processing," which has been used for many years to indicate that digital computers provide a wide variety of services to managers, is currently being virtually abandoned in favor of "management information systems." The more recent term--ushered in by the large memory, communication capabilities, and high speed of third generation computers--invokes two perspectives:

- 1) That with the aid of computers many of the activities in an organization can be more closely coordinated than heretofore, and can thereby be regulated as though they were quasi-automatic.
- 2) Somewhat antithetical to the trend toward automation (or quasi-automation)--that the quality of the management of an organization can be improved by use of computers to provide a comprehensive data base--a storehouse of information which managers can freely manipulate as background for decision making.

The two attitudes are thought, ideally, to be complementary: high volume, relatively routine activities become computer centered and some functions become computer controlled; and at the same time, largely as a byproduct, information storage in the computer provides middle and upper level managers with a data base for planning and decision making.

This idealization is however an oversimplification; it should perhaps be called a fallacy. It does not come to grips with the fact that computer centered information systems are becoming so closely interwoven with the fabric of management that in some areas the determination of computer functions cannot be separated from the determination of the managerial structure of the organization. Some managerial tasks can be formalized and incorporated in computer programs and some cannot. Some types of information which are significant for decision making can be encoded and stored in a computer and others cannot. Thus the determination of the role of the computer requires comprehensive investigation of the structure and functions of management.

Consider a manager who supervises a lower echelon of managers or a group of white collar workers. He partitions areas of information, information handling, and control functions "vertically" and "horizontally" --vertically, in distinguishing functions and information for which his subordinates are responsible from those which he reserves for himself; and horizontally in allocating functions among his staff. In many situations he considers the functions which he delegates to his subordinates to be relatively well structured, and partially programmable, and those which he reserves for himself to be ill structured. However, except for entirely routine clerical steps, he does not assume that the work of his staff can be completely programmed. Instead he recognizes that every managerial activity involves some unstructured problem solving, and requires information which is relevant to that activity.

The development of computer centered management information systems involves similar considerations, and is in some respects more demanding. Systems analysts reconsider some (but not all) previously established vertical and horizontal partitions of responsibilities among managers and their subordinates. They focus primarily on restructuring the horizontal partitioning by finding programmable tasks which can be handled in a computer. But very few functions--as seen by the manager--can be fully automated. The ill-structured aspects of each function remain unprogrammed and--after the subordinate managerial activities have been restructured--the information in the computer must be treated as a "data base" for the subordinate staff even more than for the manager. It is fallacious to aim for symbiosis between the computer and the manager rather than the computer and the manager's subordinates--as though the subordinates are quasi-automata who do not deal with ill-structured problems. The subordinate functions are not so much eliminated as rearranged so that the computer does the more routine parts; and the subordinate staff become more fully concerned with relatively ill-structured matters than they had been.

1. INFORMATION SYSTEMS IN ORGANIZATIONS

The word "system" is used with varied connotations and in many contexts. In order to deal with its ambiguities let us use the terms, "system/concept" and "system," as follows: a system/concept is an identified class of process regularities of some specified set of entities. A system is a system/concept on which there is little or no disagreement.

There have, for example, been system/concepts about motion of the sun, earth, and planets due to Copernicus, Newton, and Einstein, but use of the term, "the solar system," reflects general acceptance of the Newtonian concept.

We imply above that system/concepts and systems are products of the human mind. Beyond this, what it means to say that certain processes and entities are inherently regular is a philosophic matter which we need not pursue. The definitions simply stress two points: (1) the focus of a system/concept is more on processes (or sometimes on relationships) than on objects. For example, "solar system" seems to refer to the sun and planets, but the underlying reason for using the word "system" is the regularity of movement of the specified set of masses with respect to each other. (2) A class of process regularities is identified--a system/concept rarely covers all of the processes in which the specified set of entities is involved.

The process regularities in a system/concept may be natural, as in the case of the solar system, or man-imposed: for example, water flow in a hydroelectric power system. They may be strictly among the selected entities, or among those entities and their environment. The specified entities may be natural, engineered, or abstract. Any or all of these situations may be brought into consideration in system/concepts about organizations. Whatever the approach may be, the first essential is that regularity of a defined process or set of processes be brought under consideration.

In the past, operational regularities of organizations have generally been put into two broad categories: (1) techno-economic processes and associated analysis,--associated mainly with areas which involve extensive mechanization--such as manufacturing and transportation; and (2) the self-imposed standardization of procedures which generally characterizes large organizations, i. e., bureaucracy. But no matter how many regularities are found, the underlying human interactions in the formation and maintenance of organizations (leadership, cooperativeness, etc.) have remained ill understood, not well identified, and relatively unpredictable. Consequently, broad system/concepts about organizations have not had wide acceptance, and organizations per se are not generally called, or thought of, as systems.

The regularities of information processing, and to some extent decision making, which are being identified in the course of computer application development, form a relatively new aspect of the study of organizations. It appears that broad information system/concepts about organizations may now be put alongside bureaucratic system/concepts or techno-economic system/concepts as an indication that major aspects of organizational activity are being brought under consideration.

On a rather narrow basis there have been significant ways in which the term "system" rather than "system/concept" is quite appropriate. In the first place, well before the advent of computers, the word "system" was widely used within organizations to indicate acceptance of regularity

in administrative and data handling procedures, such as accounting systems or personnel record systems, which cut across many or all departments. Secondly, in many organizations computer departments act as quasi-independent service centers which take responsibility for whatever information processing tasks other departments choose to sponsor, which the computer can handle. The set of jobs determines a computer hardware/software system which can be called the computer-based information system of the organization. There is little ambiguity about the jobs and equipment which are involved.

But proceeding to a broader view, the introduction of computers brought prospects not only for extension and unification of routine data handling, but also for regularization and unification in decision making, information retrieval, and complex analysis (e.g., simulation). The scope of these possibilities has been symbolized, since the early fifties, by a series of catchwords. "Integrated data processing" (first generation) was followed by "total systems" (second generation). "Data bank" and "management information system" are currently popular. "Corporate simulation model" is gaining attention. Under these headings large classes of information processes have been identified and partly incorporated in computer operations. However the computer specialists, the administrative system analysts, and the operating managers of the organization have not formed a mutually agreed view of the scope of this type of development. In the terms expressed at the beginning of this section, developments to

date can more appropriately be described as information "system/concepts" than as information "systems."

The goal of such efforts is the development of information systems which cover: (1) relatively routine administration, (2) formalized decision making -- in particular the implementation of operations research models, and (3) all of the managerial uses of computers, including on-demand production of special reports, simulation, etc. It is quite obvious that agreement on the full scope and content of these classes of process can be achieved only by dealing with the organization as a whole, and within that framework by identifying a comprehensive formal information system of the organization. Accordingly, some organizations have made elaborate arrangements for joint investigation of information processes by teams of computer analysts, administrative procedure analysts, and operating managers.

This paper proposes techniques for succinct representation of all findings about regular information processes in an organization, and for the assembly of such findings in a unified structure. It constitutes a methodology -- suitable for operating managers as well as technical specialists, for the representation of information processes in sufficient detail to exhibit the relationship of proposed new developments in information handling to existing situations, and thus to facilitate planning or assessment of proposed innovations.

2. WORK-CENTERS AND ENVIRONMENT SECTORS

The "structure of an organization" is usually taken to mean the attributes which are represented by an organization chart. Two matters are, in the main, involved:

- (1) The identification of managerial and operational units
- (2) Hierarchy relationships

The division of an organization into discrete units is so closely related to its hierarchic structure that a distinction between (1) and (2) is not usually made. Here, however, rather than "lines of authority," the links between positions are to be viewed as paths along which specific types of information flow. In general terms, each path carries plans and directives from a manager to subordinates, and performance reports in the reverse direction. In the "information structure model" of an organization these loose implications are to be replaced by specific representations of recurring information transfer activities. Recurrent types of information flow which cut across the hierarchy are also to be identified. The underlying point of view is that (a) the managerial and operational units of an organization are the nodes of an information flow network, and (b) the organizational hierarchy is broadly indicative of an important set of flow-lines in that network, but does not, of course, show all of the significant links.

All units which appear on organization charts whether managerial or operational, will for simplicity be called "work-centers" -- they are all

receivers, producers, and transmitters of information. The managerial work-centers are usually identified by job titles such as President, Research Director, or Chief Accountant. It is preferable here to use descriptive phrases such as "General Policy Formulation" (which could represent the president and board of directors), "Research Management," "Quality Control," etc. This implies that for the purpose at hand the form which each work-center may take is of no direct concern; it may be: an individual with a staff, an operating department, a board or committee, the part-time work of an individual who also has other responsibilities; perhaps even a computer.

In the terms of the previous section, the work-centers are the specified entities about which a system/concept is to be formed. Process regularities -- the essential features of systems -- are to be sought with respect to (a) information flow, and (b) information transformation.

The first stage in the construction of an information structure model of an organization is to make up a list of the work-centers. Their locations in the organizational hierarchy are significant. This will, as illustrated below, be indicated by means of hierarchic labels. The labels show, for example, that the Assembly Shop, labeled 1/1/2, is subsidiary to Plant Operation Management, labeled 1/1, which is in turn subsidiary to General Production Management, labeled 1.

1. General production Mgmt.

- 1/1 Plant operation mgmt.
 - 1/1/1 Components shop
 - 1/1/2 Assembly shop
 - 1/1/3 Inspection and packing shop
- 1/2 Production scheduling
- 1/3 Production engineering
- 1/4 Purchasing

2. Marketing Mgmt.

---subsidiary work-centers

3. Financial Mgmt.

---subsidiary work-centers

etc.

The work-centers of a homologous group, e.g. the three shops above, can be expected to have a common pattern of information flow with respect to other parts of the organization. On this basis, such groups can, until quite detailed matters arise, be represented by a single entry, e.g.

1/1/h Shop h, $h = 1, 2, \dots$

Homologous multilevel subdivision, e.g. sales offices in sales regions can be similarly represented:

2/h Region h sales Mgmt., $h = 1, 2, \dots$

2/h/j Sales office j in region h, $j = 1, 2, \dots$

The labels have several roles:

- (1) By indicating location in the organizational hierarchy they act as a frame of reference which facilitates consideration of information processes;

- (2) They provide a useful set of symbols. For example, expressions such as $1/5 \rightarrow 3/2$, or $1/1/h \rightarrow 1/2$, will be used later to symbolize information flows among pairs or groups of work centers.
- (3) In order to form an organizational information structure model in which computer files and programs can be incorporated, it is desirable that all elements be carefully identified. The labeling above, and further aspects which will be introduced later, tend to sharpen the identifications.

In order to identify the types of flow of information which take place between an organization and its environment, the sectors of the environment such as suppliers, customers, tax collectors, etc. must be identified. A partitioning of the environment is implicit in an organization chart; it is to be made explicit here by listing environment sectors as well as work-centers. It is convenient to list each environment sector adjacent to the work-center to which it relates most closely, and to label it by adjoining "/X" to the work-center label, e.g. (in Figure 1)

1/4 Purchasing

followed by

1/4/X Suppliers

A generic list of work centers and environment sectors for a medium-size manufacturing firm -- for use in illustrating further stages of information structure model development -- is shown in Figure 1.

3. INFORMATION FLOW

We now turn to the representation of regular information flows of organizations. (It is to be understood that information "flow" refers here to recurrent discrete events which take place either at regular or at random intervals, for example, the "flow" of monthly expense reports from an accounting department to managerial units.) Information system analysts commonly use network type diagrams to depict existing or proposed information flow patterns. In order, however, to cope with the great variety of information flows which must be considered, a matrix format -- such as that described below -- seems preferable.

The device for identification of information flows among work-centers and environment sectors will be called an "Info-transfer matrix," Figure 2 is a preliminary illustration. The rows and columns of the matrix are named and labeled by listing work-centers and environment sectors on both a horizontal and a vertical axis. The source and destination of each information flow are indicated on the axis. Information flows are identified with regard to source, object, and destination. In order to show the source and destination of a flow, a symbol for the subject is entered in the column which corresponds to the source and the row which corresponds to the destination, e.g. "Sq" in column 1/4/X Suppliers, row 1/4 Purchasing. The subject is named on a line below the matrix:

Supply-price and delivery quotations Sq

A source label, subject symbol, and destination label in the form such

as $1/4/X - Sq \rightarrow 1/4$ provides a convenient symbol for each flow.

A single-source, multiple destination flow, is covered by multiple entry of a subject symbol in a column, e. g. Ps, which appears several times in column 1/2, indicates that product output schedules are sent from the Production Scheduling work-center to Plant Mgmt, to Plant Operations Control, and to Purchasing. Similarly, multiple-source, single-destination flows are to be expressed by multiple entry of a subject symbol in a row.

Like work-centers -- which for the purpose of organization structure representation are identified by names which give a broad indication of function -- the representation of information flow by source-subject-destination, without further detail, will be considered adequate here for information structure modeling. Matters such as medium of transmission (whether as human communication, or in machine form), frequency, or message types, will not be considered. On this score, Fairthorne [1] contends that an information flow has six main attributes: (1) subject (which he calls "designation"), (2) source, (3) destination, (4) message-set, (5) coding, (6) channel. He suggests that various three-way combinations of the six attributes (triads) -- e. g. source/subject/destination, which we use here, and others such as source/channel/destination or channel/message-set/coding -- are "basic communications activities" to which analyses of information processes can properly refer -- without necessarily dealing with all six of the attributes.

An empty Info-transfer matrix on which all of the work-centers of an organization have been placed, but no information flows have as yet been entered, is equivalent to an organization chart. (The labels of the work centers show the hierarchic structure of the organization.) Since the Info-transfer matrix provides a vehicle for representation of information flow as well as organization structure, it may be described as an extension, or generalization, of the organization chart.

An organization chart which represents a branch of a large organization is considered to be a sector of the chart for the entire organization. Similarly, an Info-transfer matrix for a branch of an organization should be viewed as a submatrix of an overall matrix for the organization. The submatrix generalizes the branch organization chart in the same way as the full-matrix generalizes the full organization chart. For example, Figure 3A (based on part of Figure 1) is an organization chart for the Production branch of a firm. Figure 3B is a corresponding Info-transfer matrix, with a sampling of information flows.

The Info-transfer matrix, which emphasizes information flow, rather than the organization chart, which emphasizes lines of authority, will be treated as the basic vehicle for the representation of organization structure in this paper.

4. PROGRAMMES IN ORGANIZATIONS

4.1 Activites and Programme Structures

An identified process which is conducted by a work-center will be called an "activity." An activity which can be modeled or is conducted

under specified rules will be called "structured;" otherwise it is unstructured." A prospective or actual occurrence of an activity will be called an "event."

For the purpose of further development of the information structure model, three types of structured activity will be distinguished:

1. Operation -- automated physical processes, and skilled or unskilled human labor which takes place under schedules and rules of procedure.
2. Information transformation -- activities such as statistical summarization, calculation, and decision-making or collection and organization of information under procedural rules.
3. Information transfer -- activities such as recording, encoding, and transmitting information.

A sequenced set of activities without specification of time or time intervals, will be called a "programme structure," or simply a "programme."*. A programme with specific times or time intervals, and with other parameters for events, will be called a "schedule." An activity is to a programme as an event is to a schedule.

A programme structure for production planning, operation, and control for the illustrative manufacturing firm is shown in Figure 4. The

*The British spelling, "programme," is used in order to distinguish its usage here from common uses of "program," as in computer program, or -- where it is synonymous with "plan" -- as in new-product development program.

activities, identified in broad terms, are named and numbered on the left. Work-centers are listed across the top. The activity numbers in the work-center columns show the tasks of the work-centers. Arrows between these entries show the normal order of events which a specific schedule for this managerial/operational process would follow.

An activity/work-center combination will be called an "assignment," and a programme structure in the form shown in Figure 4 will be called an "assignment-profile." Like a flow chart for a computer program it identifies steps and depicts sequential structure; in addition it designates "processors," i. e. work-centers.

Whether each of the activities in a programme is structured or unstructured is left for later consideration. Step 5, "Prepare work-in-progress status reports," might well be done according to definite rules even in a small firm. On the other hand, it is unlikely that step 11, "Review costs," would go beyond the free exercise of judgement by the plant manager even in a large firm.

The following convention has been used in the construction of Figure 4 and is to be applied in general to assignment profiles:

Information transfer activities are not explicitly listed -- they are considered to be implicit in the arrows which connect the assignment entries: instead they are identified explicitly by the entries in the Info-transfer matrix of the organization.

The arrow from step 1 to step 4 in Figure 4, for example, implies that information is transferred from the Production-Scheduling work-center to the Purchasing work-center. This flow, and others which are implicit in Figure 4, have been included in Figure 3B, viz. $1/2 - Lt \rightarrow 1/4, \text{Mfg } \underline{\text{Lead}} \text{ times.}$

In this way assignment profiles are interlocked with the Info-transfer matrix. As noted earlier, the matrix -- whether all of its entires are shown in a single array, or not -- is to be considered one entity for the entire organization. Thus the basis for information structure modeling of organizations, as conceived in this paper, is an Info-transfer matrix with which diverse programme structures are interlocked.

A second convention, which has been used in Figure 4 and elsewhere in this paper, with regard to the relationship between programme structures and the Info-transfer matrix is:

Interactions between work-centers and environment sectors are identified by entries in the Info-transfer matrix. Processes in the environment are implicit in some of the work-center activities, but they are not explicitly stated on assignment profiles.

This convention need not be strictly followed. In some cases, in order to clarify programme structures, explicit identification of environmental processes on assignment profiles may be desirable.

Structured activities will be discussed further below. Before proceeding, it is worth noting that explicit programmes in which the activities are in general unstructured are not uncommon. Large firms, for example, commonly identify the steps in new-product development (such as the project proposal stage, preliminary cost and market estimation, setting research and engineering budgets, etc.) and the order in which these steps take place. The activities are basically unstructured -- they involve research, innovations in production and marketing, entrepreneurial judgement, and the like. Much of the information flow between work-centers, say between Research Management and Marketing Management, is informal, and cannot be identified in any useful way, but some aspects such as research budgets, target dates, and market forecasts become formalized, and can be identified in the Info-transfer matrix.

4.2 Subprogrammes

"Structured activity," as used above, denotes process regularity in various forms:

Physical process automation

Decision-making on the basis of mathematical models

Procedural rules within work-centers

Computer-based information processing

The structuring of activities -- viewed as the decomposition of an activity into identified subactivities -- is inherent also in the very formation and development of organizations. As organizations grow,

some of the assignments to particular work-centers evolve into "subprogrammes" -- coordinated sets of activities which take place in diverse work-centers.*

The decomposition of an activity generally occurs when parts of it can be structured in one of the forms listed above, e. g. , programmed for a computer. The introduction of computers has in fact led to the proliferation of processes which are more appropriately viewed as subprogrammes of some fairly broad programme than as independent programmes. Frequently such subprogrammes include steps (activities) which may be described as fairly well identified, but relatively unstructured.

An assignment profile for a programme at any level in a hierarchy of programmes can be interlocked with the Info-transfer matrix -- on this score no distinction need be made between programmes and subprogrammes. However, when an activity in a programme is replaced by a subprogramme, the information flow entries in the matrix which are associated with that activity must be reviewed.

Activity 5 in Figure 4 identifies the purchasing of materials in the illustrative firm. In a small firm that activity would be relatively unstructured. However in a large firm it typically becomes a subprogramme such as that shown in Figure 5. As before, the arrows between assignment

*The organization of concepts into hierarchies is one of the essential attributes of human thought (Simon [2]). We observe here that as organizations grow, programme structures, as well as authority structures, develop hierarchically.

entries indicate the order of events (to be implemented in specific "schedules"), and imply information transfer activities.

Figure 6 shows the Info-transfer matrix entries which are directly implied by Figure 5. It is a step, introduced only for clarity, toward Figure 7 which includes other information flows which pertain to the purchasing of materials:

Environment flows (suppliers and common carriers)

Managerial reports

Input data required by steps in Figure 5 which are outputs of activities outside the purchasing programme.

As illustrated by Figures 5 and 7, it is convenient to pair an auxiliary matrix with an assignment profile. Each auxiliary matrix thus formed is a family of entries to the overall matrix of the organization. With this in mind, if the flows shown in Figure 7 were copied onto Figure 3B, they would contribute to the formation of a comprehensive Info-transfer matrix for the production branch of the illustrative firm.

REFERENCES

1. Fairthorne, R.A., "Morphology of Information Flow," Jnl. of the Assoc. for Computing Machinery, Vol. 14 (Oct. 1967), pp. 710-719.
2. Simon, H.A., "The Architecture of Complexity," Proc. of the American Philosophical Society 1962 (Vol. 106), pp. 467-482.

Figure 1

Work Centers and Environment Sectors of a Manufacturing Firm

- 0. Major policy determination
- 1. Production management
 - 1/1 Plant management
 - 1/1/h Shop h, h = 1,2,...
 - 1/1/20 Shipping and receiving
 - 1/1/21 Supply storage and distribution
 - 1/1/22 Plant operations control
 - 1/1/23 Plant accounting
 - 1/1/24 Plant & equipment engineering and maintenance
 - 1/2 Production scheduling
 - 1/3 Production engineering
 - 1/4 Purchasing
 - 1/4/X Suppliers
 - 1/5 Quality control
- 2. Marketing management
 - 2/1 Sales operations mgmt.
 - 2/1/X Major customers
 - 2/1/h Sales office h, h = 1,2,...
 - 2/1/h/X Local customers
 - 2/2 Distribution management
 - 2/2/X Common carriers
 - 2/2/h Regional warehouse h, h = 1,2,...
 - 2/2/20 Sales-order processing
 - 2/2/21 Truck fleet
 - 2/3 Sales/production liaison
 - 2/4 Market research
 - 2/5 Advertising
 - 2/5/X Advertising agencies and media
- 3. Financial management
 - 3/X/1 Stockholders
 - 3/X/2 Banks and other financial institutions
 - 3/1 Accounting
 - 3/2 Budgeting
 - 3/3 Credit control
 - 3/4 Tax management
 - 3/4/X Government agencies
 - 3/5 Employee benefits and pensions
- 4. Personnel management
 - 4/1 Salaried employee administration
 - 4/2 Hourly employee administration
 - 4/3 Labor union relations
 - 4/3/X Unions and in-plant union agents
 - 4/4 Personnel-record maintenance
 - 4/5 Employee services
- 5. Research and product development
- 6. Information system management
 - 6/1 Operations research
 - 6/2 Data systems analysis and programming
 - 6/3 Computer operation

Figure 2

An Info-Transfer Matrix

	Plant Mgmt.	Plant Operations Control	Production Scheduling	Purchasing	Suppliers	Sales-Production Liaison
	1/1	1/1/22	1/2	1/4	1/4/X	2/3
1/1 Plant Mgmt.			Ps			Se
1/1/22 Plant Operations Control			Ps			
1/2 Production Scheduling						Se
1/4 Purchasing			Ps		Sq	
1/4/X Suppliers				Om		
2/3 Sales-Production Liaison						

Short term Sales expectations

Se

Product output schedules

Ps

Supply price and delivery quotations

Sq

Orders for materials

Om

Figure 3A

Production Branch of a Manufacturing Organization

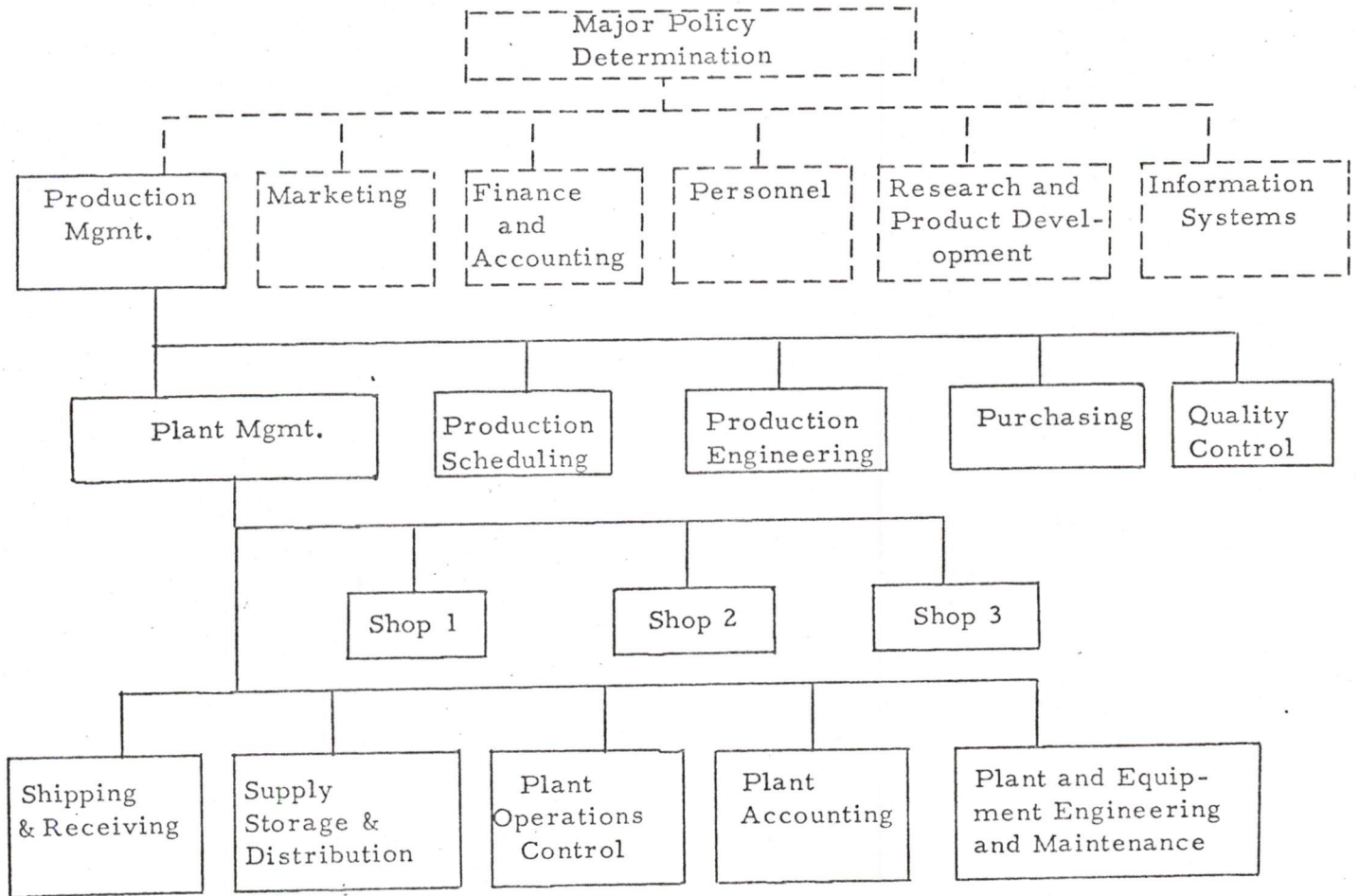


Figure 4

An Assignment Profile for Production Planning, Operation, and Control

Activities	Plant Mgmt.	Shops	Shipping and Receiving	Plant Operations Control	Plant Accounting	Production Scheduling	Purchasing	Distribution Mgmt.	Sales/Production Liaison
	1/1	1/1/h	1/1/20	1/1/22	1/1/23	1/2	1/4	2/2	2/3
1. Determine Mfg. lead time requirements						[1]			
2. Determine short term plant output requirements, based on sales expectations									[2]
3. Convert output plans into shop operation plans and material requirements						[3]			
4. Determine new orders for materials							[4]		
5. Prepare work-in-progress status reports				[5]					
6. Schedule detailed flow of work in shops				[6]					
7. Perform Mfg. operations		[7]							
8. Route products to warehouses or customers								[8]	
9. Ship products			[9]						
10. Calculate operating costs and variances from standard costs					[10]				
11. Review costs	[11]								

Figure 5

Subprogramme Assignment Profile
Raw Material Acquisition

Activities	Shipping and Receiving 1/1/20	Supplies Maintenance and Distribution 1/1/21	Production Scheduling 1/2	Production Engineering 1/3	Purchasing 1/4	Quality Control 1/5	Accounting 3/1	Research and Product Development 5	Operations Research 6/1	Computer Operation 6/3
1. Specify material characteristics				[1]				[1]		
○ (Step 3 of Fig. 4) Estimate material usage rates										
2. Recommend buffer stock levels of materials			[2]							
3. Ascertain suppliers, their product characteristics, prices, delivery capabilities					[3]					
4. Analyze transportation costs					[4]					
5. Recommend or choose suppliers				[5]				[5]		
6. Specify delivery schedules for R&D materials								[6]		
7. Evaluate material ordering and holding costs									[7]	
8. Recommend reorder point and reorder amount rules									[8]	
9. Prepare status reports on material inventories and orders outstanding										[9]
10. Check availability of materials for current Mfg. schedules		[10]								
11. Prod suppliers on past orders					[11]					
12. Formulate delivery requirements on new orders					[12]					
13. Negotiate and place new orders and reorders					[13]					
14. Receive deliveries	[14]									[14]
15. Check quality of incoming materials						[15]				
16. Store incoming material	[16]									
17. Update delivery, outstanding order, and inventory records										[17]
18. Pay supplier and transportation bills							[18]			

Figure 6

Internal Information Flows Required by
Assignment Profile for Purchasing

	1	1/1	1/1/20	1/1/21	1/2	1/3	1/4	1/4/X	1/5	2/2/X	3/1	5	6/1	6/3
1 Production Mgmt.														
1/1 Plant Mgmt.														
1/1/20 Shipping and Receiving							Os							
1/1/21 Supplies Maintenance and Distribution														Ss
1/2 Production Scheduling						Ms								
1/3 Production Engineering							Rs							
1/4 Purchasing				Sh	Ue Br	Ms Cs						Ms Cs Ds	Oq	Ss
1/4/X Suppliers														
1/5 Quality Control			Mt											
2/2/X Common Carriers														
3/1 Accounting							Os							Mr
5 Research and Product Development							Rs Os							
6/1 Operations Research					Ue Br									
6/3 Computer Operations			Mr				Os					Mr		

raw Material Specifications Ms

material Usage Estimates Ue

Buffer stock Recommendations Br

Ratings on Suppliers and delivery costs Rs

recommendation or Choice of Suppliers Cs

Delivery Schedules needed for R&D materials Ds

Supply Status: in inventory and on order Ss

Anticipated Shortages Sh

recommendations on Order Quantities and reorder points Oq

Orders on Suppliers Os

Material Received reports Mr Mr Mr

Materials issued for Testing Mt

blcc: S. Fernbach
J. Griffith
W. F. Miller
J. R. Pierce
R. Wington

0707 7
FEB 16 1971

ACADEMY OF SCIENCES
CONSTITUTION AVENUE
WASHINGTON, D. C., 20418

12 February 1971

ANTHONY G. OETTINGER, CHAIRMAN
COMPUTER SCIENCE & ENGINEERING BOARD
AIKEN COMPUTATION LABORATORY
HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS 02138

Dr. Philip Handler, President
National Academy of Sciences
2101 Constitution Avenue
Washington, D. C. 20418

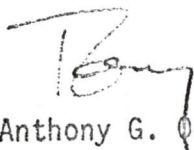
Dear Phil,

I'm enclosing an updated copy of the rotation plan we submitted in response to your request in May of last year.

The letters enclosed with the plan clarify the status of each individual Board member affected by it.

As indicated in the letters, the Board will review a slate of prospective new members at its meeting later this month. Shortly thereafter, I expect to submit this slate for your review and formal appointment.

Sincerely yours,


Anthony G. Oettinger

AGO:chm

enclosures

cc: W. C. House

COMPUTER SCIENCE AND ENGINEERING BOARD MEMBERSHIP STATUS

February 1971

FEB 16 1971

At the Pleasure of NAS President	Term ends: June 30, 1971	Term ends: June 30, 1972	Term ends: June 30, 1973	Past Members
A. G. Oettinger - Chairman	L. F. Carter W. A. Clark	W. S. Baer W. L. Lurie	M. Greenberger R. Nutt	G. Culler D. C. Evans
J. R. Pierce - Vice-Chairman	S. Fernbach J. A. Haddad J. C. R. Lickliger J. R. Meyer W. F. Miller A. J. Perlis J. B. Rosser A. F. Westin	R. Wigington		*W. Knox N. M. Neumark **K. Olsen

* Resigned upon becoming Director of the National Technical Information Service in the Dept. of Commerce.

** Resigned upon appointment to President's Science Advisory Committee.

05007-
FEB 16 1971

C



AKA

NATIONAL ACADEMY OF SCIENCES

2101 CONSTITUTION AVENUE

WASHINGTON, D. C. 20418

19 February 1971

ANTHONY G. OETTINGER, CHAIRMAN
COMPUTER SCIENCE & ENGINEERING BOARD
AIKEN COMPUTATION LABORATORY
HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS 02138

Dr. Philip Handler, President
National Academy of Sciences
2101 Constitution Avenue
Washington, D. C. 20418

Dear Phil,

I am enclosing a copy of a letter I recently received from Alan Westin regarding the publication of the forthcoming Panel Report of the Project on Computer Data Banks.

My reading of the grant letter from Russell Sage Foundation dated 25 February 1969, of which a relevant excerpt is enclosed, suggests that the procedure outlined by Dr. Westin is quite appropriate as well as highly desirable. I should, however, very much appreciate your advice concerning any policy issues this might raise of which I might be unaware.

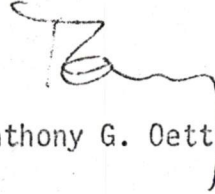
Since the Report Review Committee has been designated as the reviewing body for this report, I have kept Kisty abreast of the work schedule as it has progressed. The current plan is to have the draft report available for review by the RRC, the Board and the study's advisory group on April 24, 1971, thereby allowing a substantial portion of the month of May for such revisions as may prove necessary following the reviews.

A number of subsidiary questions will arise as we proceed which I think Warren House can work out with Alan Westin and appropriate members of the Academy and Russell Sage staff. One example is the question of whether we should issue a formal report to Russell Sage Foundation in some intermediate form like multilith reproduction or regard the proposed New York Times publication as the formal report itself. Questions of acknowledgements, incorporation of letters of transmittal or the like from yourself or Bert Brim, details regarding the number of copies to be provided for free distribution, etc. also fall in this category. I anticipate no difficulty

Dr. Philip Handler
19 February 1971
page 2

on any of these scores, having found all parties to this project most reasonable and accommodating with respect to every question that has arisen to date.

Sincerely yours,



Anthony G. Oettinger

AGO:chm

cc: W. C. House
A. F. Westin

enclosure

blcc: S. Fernbach
J. Griffith
W. Miller
J. Pierce
R. Wigington

NATIONAL ACADEMY OF SCIENCES

2101 CONSTITUTION AVENUE
WASHINGTON, D. C. 20418

COMPUTER SCIENCE AND ENGINEERING BOARD
PROJECT ON COMPUTER DATA BANKS
ALAN F. WESTIN, DIRECTOR

PROJECT HEADQUARTERS:
JOSEPH HENRY BUILDING, ROOM 536
2100 PENNSYLVANIA AVENUE, N.W.
PHONE (202) 961-1335

February 10, 1971

Professor Anthony Oettinger
Aiken Computation Laboratory
Harvard University
Cambridge, Mass. 02138

Dear Tony:

This letter will summarize the conversations that I've been having relating to various means of publication and dissemination of the final report of the Project on Computer Data Banks. As you know, I have been concerned about the problem of timely issuance of our report, since we have promised to have it out in June and any delay beyond that month would be likely to harm the public impact of our findings. At the same time, the tremendous breadth of material that we have to encompass in our report and the various reviews and clearances that we must plan for make it highly unlikely that we will have finished copy to supply to any printing or publishing agent before a date such as June 7th.

I've also been concerned about insuring the widest possible distribution of our report, at very low cost, to the highly diverse audience that ought to be interested in it, a group that includes not only scientists in the computer field, social scientists interested in data collection and utilization, and legal specialists, but the broad public concerned with the civil liberties and social implications of the use of computers and communication systems in the keeping of records about individuals. Finally, there is the matter of trying to arrange for the availability of copies of the final report simultaneously with the release date and press conference that is anticipated for June.

It was to explore these matters with our sponsor, Russel Sage Foundation, that you, myself and Dr. Orville Brim, Jr., the President of Russell Sage, met in New York City on January 26th. As you recall, Dr. Brim indicated that he was sympathetic with the concerns that we communicated, and gave his strong endorsement to the notion that we should sound out possible publishers who specialize in the very rapid issuance of reports of national importance to see whether our document might be viewed by such a commercial distributor as an appropriate item for such treatment. We agreed at this conference that our ideal target was the book division of the New York Times, since they have had a distinguished record in speedy printing and national distribution

of documents such as the Warren Commission Report and the current book by Telford Taylor on the legal implications of the Viet Nam war. It was agreed at that meeting that our objective was a dignified and low cost volume, given national distribution, in which there would be no money royalties included in the publishing arrangement, a low purchase price, and a minimum profit to the publisher in fair compensation for the publishing risks involved. Under this plan, the Project would also receive, without cost, a large number of copies necessary for distribution to members of the Project staff, the Computer Science and Engineering Board, consultants to the Project, and key members of the organizations which cooperated in the site visits and responded to the large sample survey that we have distributed.

Pursuant to this meeting, I had a conversation on February 5th with Mr. Herbert Nagurny, Associate Director of the Book Division of the New York Times. Mr. Nagurny had heard of our Project and indicated that he had even made a note to himself that he wanted to contact us to see whether we would consider a publication by the New York Times Book Division. After listening to my description of the research that we have done and the report that we plan to issue, he expressed a firm desire to publish the book. I indicated that I would report this desire back to you, for the appropriate discussions and clearances with Russell Sage Foundation and the National Academy of Sciences, and that I would get back to him after these discussions had taken place.

The various terms that Mr. Nagurny mentioned are as follows. In keeping with the length of the report as I described it to him, he indicated that this would be a paperback book of approximately 128 pages, with a small hard cover edition to insure reviews and to satisfy library orders. In keeping with my indication that this would be a no royalty contract, he expected that the book would sell for 95¢, which he indicated was the minimum amount which should be charged if the largest possible number of book stores are to be willing to handle it. On this basis, the Project on Computer Data Banks would receive at the time of publication 3500 copies of the paperback volume for distribution to a list of persons that we would supply. In addition, the New York Times would pay for mailing these copies directly to those on the list that we would supply, with a printed explanation card that these came from the Project on Computer Data Banks, which would relieve us of both the financial and administrative costs of such a distribution of copies to those to whom we are indebted. We left open the question of any additional free copies to our Project that might be desired at a later time, probably to be dependant on the sale of the book beyond the first printing of 50,000 copies that would be anticipated.

The distribution of the book would be handled with the full resources of the New York Times Book Division, as in their treatment of the Warren Commission Report. Mr. Nagurny indicated that to have his production, distribution, and sales facilities properly geared up for a June publication, he would like to have the authorization to publish by the date in early March.

In light of the distinction that publication with the New York Times would bring to our report, I have not sought to obtain quotations from the other major publisher of public affairs reports in rapid printings, Bantam Books, but I could do this if you think that such a competitive quotation would be useful. My own feeling is that the name of the New York Times and the image of serious communication with the public affairs audience of the nation that goes with this is of such importance to our effort that this firm offer from the New York Times is

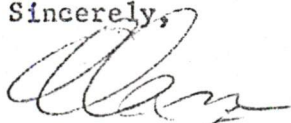
February 10, 1971

the preferred alternative, and that we ought to proceed to discuss this with the proper authorities in such a light.

I know that Dr. Brim remains very enthusiastic about the idea of disseminating our report in such a manner, and I think he will be very pleased to learn that the Book Division of the New York Times has made such a quick and strong commitment to publish the volume if we choose to make it available to them. If I can be of any further help in supplying information about my conversation with Mr. Nagurny or any other aspect of the report's dissemination, please do not hesitate to call on me. Given the rapid approach of our publication date, I hope that you will be able to follow through with this at the National Academy and let me know how to proceed.

With best personal regards,

Sincerely,



Alan F. Westin

AFW/lc

CC

Dr. Orville Brim, Jr.

P.S.

I neglected to mention above that Mr. Nagurney assured me that delivery to him of completed manuscript on June 7th will produce finished copies of the book in paperback for distribution simultaneously with the release of the report and any press conference held in the last week of June. He said that they can produce finished books from such copy in "under two weeks."

EXCERPT FROM BRIN LETTER TO COLEMAN
Dated 25 February 1969

"In regard to publication of work resulting from this appropriation, we would like to follow the procedure we established for the previous appropriation for a Study of Governmental Support and Utilization of the Behavioral Sciences in February 1966, namely, that the formal report resulting from the study be issued by the National Academy of Sciences, and that Russell Sage Foundation reserve the right to possible publication and copyright of the volume that Dr. Westin will write."

D

J. A. Haddad
P. O. Box 390, Poughkeepsie, N. Y. 12602

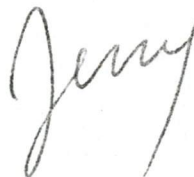
1400 Z
FEB 10 RECD

February 2, 1971

Dear Dick:

Confirming our telephone conversation, I would be delighted to act as a communications channel between the AFIPS project and the CS&E Board; and as my first step, I will report at the next meeting on whatever information I have. Since the next CS&E meeting I will attend is the 24th and 25th of February, I don't expect you will have anything further to add by that time.

Sincerely,



cc: Professor A. G. Oettinger ✓

Mr. Richard G. Canning
AFIPS Secretary
925 Anza Avenue
Vista, California 92083

bcc: Dr. M. P. Feder

J. A. HADDAD
JAN 23 10 23 AM '71
HEADQUARTERS • 210 SUMMIT AVE. • MONTVALE, N.J. 07645 • 201-391-9810

REPLY TO: 925 ANZA AVE. VISTA, CALIF. 92083

January 25, 1971

*Office
Index*

Mr. Jerrier A. Haddad
Vice President and Director
Poughkeepsie Laboratory
IBM Corporation
Box 390
Poughkeepsie, N. Y. 12602

Dear Jerry:

Bob Patrick has suggested that I write you--since you are a member of Tony Oettinger's CS&E Board--about an activity that AFIPS is undertaking. Both Keith Uncapher, AFIPS Vice President, and I concur with Bob's suggestion.

AFIPS is beginning a project to study the feasibility of system certification, particularly for systems that affect the public interest--such as vote counting systems. Bob will be chairing a workshop that will get this project under way, to be held at the end of February. I am enclosing a copy of the material that has been sent to the invited participants for this workshop.

Bob points out that the CS&E Board is studying the question of privacy and security in computer-based systems. Since this subject is closely related to system certification, we feel that it might be well to have two-way communication between the AFIPS project and the CS&E Board. We would like to initiate this communication with this letter. And we would plan to send you a copy of Bob's report of workshop results. Bob has suggested that you might be willing to be the point of contact for this communications; and can pass our information along to other interested members of your Board.

Page 2

If you feel that this communication will be mutually helpful and would like to be the person we contact, I would appreciate hearing from you.

Sincerely,

Rich
Richard G. Canning
AFIPS Secretary

cc: Keith Uncapher
Bob Patrick 19/03

Encl.

Sponsoring Societies

Members: The Association for Computing Machinery; The Institute of Electrical and Electronics Engineers Computer Group; Simulation Councils, Inc.; American Society for Information Science. Affiliates: American Institute of Certified Public Accountants; American Statistical Association; Association for Computational Linguistics; Society for Industrial and Applied Mathematics; Society for Information Display; Special Libraries Association.

AFIPS System Certification WorkshopBackground

For several years, various interested citizens, computer professionals, and elected officials have concerned themselves with security and privacy in information systems. More recently some systems involving the public's interest have failed to perform properly. There are a growing number of systems where failure could cause severe damage to a person's reputation, his finances, or in the case of medical systems, grave bodily harm.

By far the largest body of thoughtful writing lies in the domain of security and privacy. To a man, these authors describe features which should be present to protect the individual, the data base, or both. Yet these very authors neglect to propose any mechanism for determining that the desirable features are in fact present, that they are sufficient, and that they work.

In an entirely different context the professional computer community has been debating the pros and cons relating to the certification of professional competence in computing. While the debate has raged, the DPMA has certified over 11,000 individuals as having at least measurable competence in computer management matters (juxtaposed to programming skill). There are several embryonic efforts aimed at licensing individuals in their states of residency.

Independent from, but clearly related to the above matters, is the problem of certifying the adequacy of a system design and

later certifying the quality of the implementation of that design. It appears that some mechanism for certifying systems needs to be devised independent of the question of whether the persons who signed the certification are in-themselves certified, licensed, or otherwise officially recognized. The Systems Certification Workshop will investigate this question and endeavor to determine:

- ° Should such a mechanism be formally established?
- ° Does the state of technology allow such a mechanism to be established now?
- ° How should such a mechanism be established?

Definition

A system is an amalgam of hardware, software, applications programs, procedures, people, communications, and facilities which operate in concert to achieve a specific goal. For the purposes of the Workshop, systems under consideration will be limited to

- a) vote counting systems,
- b) law enforcement data banks,
- c) credit files,

and any similar systems where the public's interests are uniquely involved.

System certification is a series of ^{dis}continuous but related activities. The first of these activities, design certification, consists of carefully reviewing the requirements for a system and carefully reviewing the design for a system and attesting that the

requirements properly describe the needs for an amalgum of hardware/software/applications, etc., while simultaneously attesting that the proposed design in fact fulfills those needs.

Implementation certification occurs after the design has been implemented and probably takes place as part of acceptance and demonstration tests. To certify an implementation one certifies a mechanism to exercise that implementation under both normal and abnormal cases to be sure that no insidious oversights have occurred. When an implementation is certified, the design must be re-certified to attest its still current adequacy.

Check certification occurs periodically to a production system whose implementation has been previously certified. A check certification re-certifies both the design and the implementation following one of two conditions: either a change has been made to a critical system component; or a period of time has elapsed and a check certification is triggered to guard against changes in the environment, the work force, or changes in the legitimate needs of the application area having caused some portion of a previously adequate system to now be deficient under current operating conditions.

Goals

The goals of the Workshop are:

1. To amend or revise the above definition of a system.
2. To amend or revise the limitation of considerations to those where the public's third party interest is intrinsic.

- 4 -

3. To understand these systems so their scope may be measured, and the magnitude of the certification effort may be enumerated (certifying a total system may be beyond today's technology).
4. To propose ways of parsing the system certification problem so it may be broken up, attacked, and solved.
5. To summarize the proceedings with recommendations for future actions (if any), and the role AFIPS might play in these actions.

Workshop Format

A two day weekend shirt-sleeve workshop will be held in San Diego, California, February 27 and 28, 1971. Eight senior professionals from the computer community were chosen who are proven experts in the architecture of computer application systems and have some recent experience to bring to bear on the problem. They will be joined for the Workshop by two members of the AFIPS Board. The session will be informal, although some of the participants will be invited to bring samples of their recent work and to prepare 20 minute informal introductions to specific topics. A moderator/chairman will guide the discussions, take notes, and summarize the two day Workshop prior to adjournment.

EXTRACT FROM CATHCART LETTER 11-3-70

Contrary to the beliefs of some of my colleagues, I believe that the State of California does have an obligation to investigate and consider certification in certain areas where the public's interest as a third party is uniquely involved. As I indicated to you last month on the phone, I believe the public's interests are uniquely involved in the following three areas:

- A. Vote counting systems
- B. Law enforcement data banks
- C. Credit files.

I believe the State has the same rights and obligations to protect third party interests in these three important areas as it does licensing medical doctors or structural engineers. I also believe that even though certification is very difficult in the computer field (I was a member of the DPMA Certification Council for three years), I do believe sufficient progress could be made in these three areas to warrant some action at this time.

Further, as we discussed over the phone earlier, there is a lack of unanimity as to what that action should be. *For my part, I am leaning toward certification of systems rather than licensing individuals.* It is my belief that a two year intensive study would be required before legislation could be drafted covering these three important areas. This study would be aimed at limiting the breadth of new legislation to these three important areas, understanding the nature of the problems, predicting what problems will

- 2 -

probably occur in the future, informing the membership of the professional community, and drafting specific legislation which could be introduced together with estimates of the cost of that legislation.

As you probably know, the computer field lacks a glossary and the word meanings are sometimes conflicting and ambiguous. This has inhibited the setting up of the DPMA exam, inhibits the writing of legal contracts within the field, and has inhibited my efforts at revising the personnel regulations for the United States Air Force. Thus I conclude that any action in this very important field will require sufficient time for a careful approach and that initial action must of necessity be limited to those areas where the public's third party interest are intrinsic if we are to gain acceptance from the professional community.

Robert L. Patrick

Page & I

He
Layout

LIST OF INVITEES TO WORKSHOP

Mr. Robert Barton
 University of Utah
 1400 E. 2nd Street South
 Salt Lake City, Utah 84112

Mr. Robert Bemer
 Honeywell Information Systems, Inc.
 13430 N. Black Canyon Highway
 Phoenix, Ariz. 85029

Dr. Robert Brown
 Arcata National Corp.
 495 Arbor Road
 Menlo Park, Calif. 94025

Mr. Willis Ware
 Rand Corporation
 1700 Main Street
 Santa Monica, Calif. 90406

Mr. Clark Weissman
 System Development Corp.
 2500 Colorado Avenue
 Santa Monica, Calif. 90406

Vote counting system expert
 To be invited

Law enforcement system expert
 To be invited

Robert L. Patrick
 Workshop Chairman
 9935 Donna Avenue
 Northridge, Calif. 91324

AFIPS representatives

Mr. Keith Uncapher
 AFIPS Vice President
 Rand Corporation
 1700 Main Street
 Santa Monica, Cal. 90406

Mr. Donn Parker
 Chairman, AFIPS Professionalism Committee
 Stanford Research Institute
 333 Ravenswood Avenue
 Menlo Park, Calif. 94025

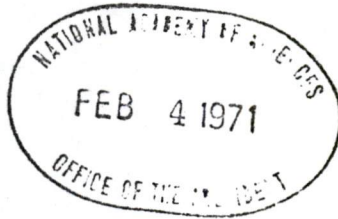
E



NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

WASHINGTON, D.C. 20546

OFFICE OF THE ADMINISTRATOR



1400 7
FEB 5 Rec'd

FEB - 2 1971

Dr. Philip Handler
President
National Academy of Sciences
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

Dear Dr. Handler:

On January 29, 1971, NASA announced the signing of a NASA-ARPA agreement whereby the Ames Research Center will act as the host site for a powerful new computer, Illiac IV, developed by the University of Illinois under contract to ARPA. The computer, unique in its capability to accomplish parallel array processing, will be used in support of ARPA sponsored research, and by Ames in the field of computational fluid dynamics.

A copy of the Agreement is enclosed for your information.

Sincerely yours,

Homer E. Newell
Associate Administrator

Enclosure

Copy: Mr. Warren House ✓
Dr. Hugh Odishaw

MEMORANDUM OF UNDERSTANDING
BETWEEN
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AND
ADVANCED RESEARCH PROJECTS AGENCY
CONCERNING
THE ILLIAC IV COMPUTER SYSTEM

I. Background and Purpose

The Advanced Research Projects Agency (ARPA) of the Department of Defense conducts research in information processing technology. A specific product of this research is an advanced prototype computer, ILLIAC IV, utilizing parallel processing as a computational technique. ILLIAC IV is in final stages of assembly by Burroughs Corporation under contracts sponsored by ARPA.

The objectives of the ILLIAC IV development program are these:

1. To successfully demonstrate the efficiency and versatility of parallel array processing.
2. To make this demonstration utilizing a sufficiently powerful hardware/software system such that the cost effectiveness and importance of array processing is adequately visible.
3. To permit a variety of DoD, NASA and private sector activities to utilize the initial system sufficiently to develop and test software, evaluate the usefulness of array processing for their needs, and to solve a series of practical problems beyond the capabilities of other machines.

ILLIAC IV will be operated as a continuation of the ARPA computer research and development program with its primary goal being to define the operating envelope of the machine. Problems to be studied on the prototype machine will include global atmosphere modeling, weather prediction, fluid dynamic problems, radar signal processing and other problems amenable to parallel processing and which further the objectives of the research and development program.

ARPA has requested the assistance of the National Aeronautics and Space Administration (NASA), as provided for in paragraphs II and III below, in the completion, installation and operation of the ILLIAC IV Computer System. It is the understanding and agreement of the parties that the assistance referred to herein will be furnished by the NASA-Ames Research Center, Moffett Field, California, in accordance with the attached NASA proposal dated October 30, 1970.

II. Responsibilities

A. NASA will:

1. Provide facilities at Ames Research Center to house the ILLIAC IV Computer System, as outlined in the attached NASA proposal, subject to availability of funds.
2. Provide technical and other services relative to the ILLIAC IV Computer System, as outlined in the attached NASA proposal.
3. Inform ARPA, on a quarterly basis, of all costs incurred under this memorandum and chargeable to ARPA in accordance with Section III-B below. Reports will be rendered by the Research Support Directorate, Ames Research Center.

- B. ARPA will provide overall technical guidance relative to the completion and installation of the ILLIAC IV Computer System.
- C. ARPA and NASA jointly shall establish all policies and procedures relative to the acceptance, management, use and operation of the ILLIAC IV Computer System.

III. Funding

- A. NASA will fund the facilities referred to in Section II-A-1 above, including special construction and equipment items, and will provide the associated utilities required. NASA will gain right to 18% of available user time on ILLIAC IV based on the following investment items totalling \$2,850,000:
 - 1. Contribution of \$2 million to ARPA which represents an investment as a user in the hardware costs of the ILLIAC IV Computer System.
 - 2. Interactive graphics equipment or other peripheral hardware, as agreed upon by ARPA and Ames Research Center, not to exceed \$400,000 in cost.
 - 3. Special construction and equipment items, totalling approximately \$450,000 referred to in III-A above and included in the facility to house the ILLIAC IV, e.g., computer air conditioning equipment, the computer floor, and fire protection equipment.
- B. Except for those costs to be funded by NASA in accordance with Section III-A, above, ARPA will be responsible for all costs, including but not limited to the following:

1. Costs arising from the current contracts with University of Illinois (and subcontracts) for the development of hardware and software systems of ILLIAC IV.
2. Costs (exclusive of civil service salaries and utilities) incurred by the host installation (NASA-Ames Research Center) in carrying out jointly-approved programs for the future development of hardware and software systems of ILLIAC IV.
3. Costs (exclusive of civil service salaries and utilities) incurred by the host installation associated with completion, delivery, installation, maintenance, and operations (including user services) of ILLIAC IV.

IV. General

- A. All assistance to be provided by NASA under this memorandum will be performed in accordance with the provisions of the attached NASA proposal.
- B. Each party assumes responsibility, when physical possession is taken, for safeguarding classified information and material received from the other party. Such safeguarding will be in accordance with the regulations of the receiving party.
- C. This Memorandum of Understanding will remain in force and effect for five years, unless terminated by joint agreement.

D. With respect to administration of this memorandum, including responsibilities in paragraph IIC, the point of contact in ARPA will be the Director, Information Processing Techniques, and in NASA, the Director, Research Support, Ames Research Center.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Hans M. Mark
 Dr. Hans M. Mark, Director
 NASA, Ames Research Center

Date: January 26, 1971

APPROVED: Jacob E. Smart
 Jacob E. Smart
 Assistant Administrator for DOD and
 Interagency Affairs, NASA Headquarters

Date: January 29, 1971

S. J. Lukasik
 S. J. Lukasik
 Acting Director
 Advanced Research Projects Agency

Date: 29 January 1971

22 January 1971

TO: Mr. John S. Coleman
Executive Officer

FROM: J. F. Kettler

Two copies of the CS&EB sponsored report "Computer Science Education: Goals and Guidelines for the Planning of Four-Year College and Graduate Programs in Computer Science" are provided with this memorandum for transmittal to the National Science Foundation. No additional letter of transmittal has been drafted in consonance with guidance from your office. No arrangement has been made for publication. I can readily provide a dozen additional copies if you see a need for them.

If I can assist, please let me know.

Attachment
As stated.

JFK/laa

Page Two
Transmittal Letter

programs and staff the larger, more scientifically oriented computer installations," while "a number of participants regarded this limitation on conference scope as a serious mistake."

2. That the conference was held at a time before the full effects of the current national economic situation on the demand for scientific and technical personnel of all types could be foreseen. While this may be only a temporary situation, it does require changes in the models used for forecasting demand.

3. That difficulties in determining and projecting how many computers are installed in the U.S., and how many computer science degree holders are needed per installation also create an unknown margin of error in the demand models.

Consequently, the Board recommends that the "Goals and Guidelines for the Planning of Four-Year College and Graduate Programs in Computer Science" resulting from this conference be interpreted in the light of the foregoing comments.

Sincerely yours,

Anthony G. Oettinger
Chairman
Computer Science and Engineering Board

AGO/1aa

21 January 1971

Mr. Kent Curtis
Head, Computer Science and
Engineering Section
Office of Computing Activities
National Science Foundation
Washington, D. C.
20550

Dear Mr. Curtis:

I hereby transmit to you an account of the conference on Computer Science Education chaired by Dr. Alan Perlis in Annapolis, Maryland, in July 1969, with the support of the National Science Foundation and under the sponsorship of the Computer Science and Engineering Board.

The purpose of the conference was to prepare for the National Science Foundation a report on computer science education in the United States, with particular attention to graduate education in computer science and to education in software and hardware systems. Explicit information was to be developed about the relations among the expected needs for these types of education, the resources required to meet these needs under various response alternatives, and courses and programs responsive to the needs.

The conference proceedings present data, depict an approach to educational planning and illustrate types of analyses which the Board believes can be useful adjuncts to educational planning and management in the computer field.

However, in transmitting these proceedings, the Board also wishes to stress:

1. That a majority of the conferees interpreted their charge as dealing, as they put it, "principally with the education of those who will teach computer science in bachelor's degree and graduate level

NATIONAL ACADEMY OF SCIENCES

2101 CONSTITUTION AVENUE
WASHINGTON, D. C. 20418

WCH
Perles file
69007-
10N 24 RECD
OK
28 Jan

January 26, 1971

Dr. William D. McElroy
Director
National Science Foundation
1800 G Street, N.W.
Washington, D. C. 20550

Dear Dr. McElroy:

Transmitted herewith is an account of a conference on computer science education in the United States held in Annapolis, Maryland, July 21-24, 1969, under the aegis of the Computer Science and Engineering Board of the National Academy of Sciences. A copy of this report is also being sent to Mr. Kent Curtis in the Foundation Office of Computing Activities.

Sincerely yours,

John S. Coleman
Executive Officer

cc: Mr. Kent Curtis
Mr. W. C. House
Professor Anthony G. Oettinger

bcc: Mr. R. E. Green

Day Session

NATIONAL ACADEMY OF SCIENCES

COMPUTER SCIENCE & ENGINEERING BOARD
2101 CONSTITUTION AVENUE
WASHINGTON, D. C. 20418

COMPUTER SCIENCE AND ENGINEERING BOARD

AGENDA

Day Session 25 February 1971

Joseph Henry Building
Room 600A

AGENDA ITEMS

NOTES FOR ACTION TAKEN OR PLANNED

0900 - 0915

(A) Recapitulation of Executive Committee
Organization

The Chairman

0915 - 0930

(B) CLR Status

R. Wigington/The Chairman

0930 - 1000

(C) Russell Sage Status

A. Westin

1000 - 1100

(D) Patent, Copyright and Associated Considerations
in the Software Field

E. W. Galbi, IBM

1100 - 1200

(E) Professional Societies

W. Carlson, President, ACM

1200 - 1230

LUNCH

1230 - 1300

(F) National Bureau of Standards Highlights

R. Davis

1300 - 1315

(G) NTIS Considerations

W. Knox

1315 - 1430

(H) Data Base Organization

R. Engles/M. Feder

AGENDA ITEMS (cont.)

NOTES FOR ACTION TAKEN OR PLANNED

1430 - 1515

(I) Files Security

M. Feder

1515 - 1530

(J) Review List of Tentative Projects
The Chairman

1530 - 1545

(K) New Business

The Chairman

ADDENDA

Day Session

(L) NASA-ARPA-Illiack IV (Ames Research Center)
The Chairman

E

January 12, 1971

To: WH - Please reproduce
for Exec Com Evening Agenda -
A Progress Report

Projects to Combat the "Blame the Computer" Syndrome

Submitted by: Walter M. Carlson, ACM President

Note esp,
p. 7, which
we'll discuss.

Nearly all the recipients of this progress report have been in touch with my efforts on this subject since last September. The consolidation and evaluation phase is over, and work is under way in four selected areas of activity. Each shall be described after a brief introduction to the problem and a description of the steps taken to date.

The Problem

Two events in early September, 1970 crystallized the problem for me neatly:

- . during a late-night radio interview show at ACM-70, a group of distinguished computer experts were extolling the accomplishments of our profession by saying "the computer does this, the computer does that, etc.". It was clear that the host of the show (and by extension the radio audience) were accepting this personification of the computer.
- . a senior government official ascribed a near-miss aircraft incident to computer failure.

These two events explained why an ever-increasing segment of the mass media and the public have begun to blame computers for the problems they are suffering at the hands of data processing applications. It is because the computer professionals, in using a convenient shortcut, have created the widespread impression that the machines are the active intelligence of the applications rather than humans.

My purpose, therefore, has been to examine the possibilities for a broad-gauge set of projects to redress this imbalance and to achieve a better public grasp of the actual roles played by people, the data, and the machines in our modern information processing systems based upon computers.

The Investigation

This has been essentially a one-man task force kind of effort. It was apparent at the outset that a large number of people had sensed the problem and had solid ideas for attacking it. What was not needed was a committee to define the problem and to propose solutions. What was needed was someone willing to communicate with all corners of the profession and the industry and to recommend priority projects for prompt execution.

There was uniformly high interest among the dozens of people contacted, and there has been a high degree of enthusiasm for the project proposals. Only time will tell whether the projects, in their collective impact, can have the desired effect of educating the public properly on how to assess the faults in computer applications.

The Objectives Considered

After about one month's review, it became evident that a very broad framework is needed to accommodate the kinds of ideas that are available for combatting the "blame the computer syndrome." Accordingly four main objectives were identified, and several possible actions to achieve each objective were postulated. Without further elaboration, the four broad objectives are:

1. Independent, public review of facts involved in each "incident"
2. Protection of public health, safety, and welfare against mis-use of computers
3. Public education on need for effective interaction between data, people, and machines.
4. Clarify competence levels required for successful use of computers

It is not feasible to fulfill all of these objectives through a brief investigation of this nature. They did serve, however, as an excellent framework for evaluating alternatives and for obtaining commitments to specific projects.

The Resultant Strategy

As the different ideas were compared against these objectives, there emerged two short-range approaches and two long-range approaches.

It is evident that someone whose credit card has been confiscated in a restaurant or whose wife is in tears over a store billing controversy is interested in immediate access to assistance. A project has been proposed to deal with the problems of aggrieved individuals.

A newspaper, TV, radio, or magazine story which places the blame in error because of poor facts or poor understanding on the part of a journalist does not wait for the long range. Immediate corrections must be provided. A project has been proposed to provide fast reaction and response to press stories that need correction.

On the other hand, gaining the attention and understanding of thought leaders having high responsibility for public activities is a matter that requires careful preparation and lengthy periods for implementation. A project has been proposed to carry out effective work with such thought leaders under the most prestigious technical auspices available.

On an even longer time scale, the general public must learn to feel comfortable about the ways computers are used and to recognize the essential role that human choice plays in their successful use. A project is being explored for using our most pervasive mass medium, TV, for achieving a strongly favorable public attitude.

The Four Projects

1. Ombudsmen

The basic idea is to provide a technically astute person in a locality to respond to calls for assistance from citizens or establishments who are having trouble because of some computer application and do not know what to do about it.

The logical responsibility for such a project lies with the local units of the major technical societies interested in computing. Accordingly, ACM has undertaken to outline a detailed program for creating local ombudsmen in cooperation with DPMA, the IEEE Computer Society, and with others where it may be appropriate.

A draft of an instruction to all local ACM chapters is now circulating within ACM, IEEE, DPMA, and AFIPS for review and comment. It is expected that the ACM Executive Committee will decide on February 16 whether to proceed with the project and, if so, under what guidelines.

In its present form, the proposal for local ombudsmen places several key limitations on the way in which such people could operate. They must restrict their scope to technical findings and advice on matters brought to them by others. The only instance in which the ombudsman might act on his own initiative would be to seek out the reporter who wrote an erroneous description of a computer-related problem for the local paper; the ombudsman would be permitted to give the reporter the technical background necessary to understand his error.

As the project is currently planned, the selection of the local volunteers for this public service would be the joint responsibility of the local units of the computer societies. In large metropolitan areas, more than one person may be required. ACM Headquarters will be expected to collect reports from the ombudsmen and to issue periodic newsletters on the incidents and the techniques used for resolving them.

2. Fast Response

The basic idea is to give misleading treatment of computer applications in mass media a prompt response with a balanced, technically accurate statement.

The logical responsibility for this project lies with the industry trade association, in this case the Business Equipment Manufacturers Association (BEMA). The 1971 budget in BEMA carries a specific project on the "blame the computer syndrome", and implementation details and final approvals are now being developed by the BEMA staff.

There seems to be an increasing intensity and frequency of articles and statements that use the classic shortcut of ascribing all manner of evil habits to the computer without even mentioning the failure to provide adequate data or failure to foresee events or conditions when the programs were written. There are important problems that arise when the incident under discussion involves a single system manufacturer and a single customer. These problems are best handled through the more detached (but professionally competent) channels of the trade association.

This short term, fast-response effort has a longer term impact, too. The continuing contact of the journalist and his editors with responsible and accurate treatment of the facts will make them more sensitive for future stories.

3. Reaching Thought Leaders

The basic idea is to give important public executives and legislators a better background in uses of computers than they now possess. Until now, the computer community has tended to deal only with governmental units having high technology missions, and very little contact has been developed with the leadership responsible for people-oriented missions.

The logical responsibility for sponsoring a project of this nature lies with the Board of Computer Science and Engineering in the National Academy of Sciences. This Board would not assume the operational responsibility for the detailed efforts undertaken, but its sponsorship and coordination of the action bodies is essential to the ultimate success of the project. The Board has begun to examine this proposal and plans to discuss its adoption at a meeting scheduled on February 24.

The current thinking is that the Board would convene a meeting of the leadership in the computer societies and other technical organizations to develop broad guidelines to be used in several programs designed to assist the thought leaders to learn more about computers and their applications. The organizations involved would probably include AFIPS, ACM, IEEE Computer Society, SCI, SIAM, DPMA, ASIS, and ASM (Assoc. for Systems Mgt.), plus government units such as the Center for Computer Science and Technology and the NSF Computer Division.

The detailed programs undertaken by these organizations would be developed and funded by the individual organizations to give maximum flexibility and timeliness.

4. TV Games

The basic idea is to extend the present popular formats for daytime TV games into activities that depend upon human choices requiring some form of computer logic for timely implementation. Whatever the degree of computer sophistication involved, it would always be subordinated to the human players, and the role of the computer itself would rarely be formally sensed by the audience.

The logical responsibility for this project is a production company especially funded with risk capital to develop and exploit the more promising games. No such company appears to exist, so a series of discussions has been started toward formation of one.

A keystone to such an enterprise is useful ideas for TV games based upon computers. Fay Baker, incorporated as Lovelace, has been collecting these games for about four years, and she has several in sufficiently developed stages of concept and planning to be considered as candidates. The ideas for this kind of game appear to be numerous, even numberless.

The end objective, if a commercially viable format can be found, is to make the American daytime TV viewer appreciate that interesting, challenging, and even joyous things can be accomplished by people whose thinking process is aided by computers. When this message has sunk in, there will be no need to worry about the "blame the computer syndrome."

14007-
~~0900~~
FEB 22 REC'D

SDD POUGHKEEPSIE
Dept. B11, Bldg. 706
Extension 3-7284
February 18, 1971

Memorandum To: Warren House

Subject: Data-Base Tutorial - February 24, 1971

The attached is an abstract of the presentation to be made by Mr. Robert Engles concerning Data-Base Organization. You might want to include it in the Board workbook for the February meeting.

Michael P. Feder

M. P. Feder

MPF/drs

A TUTORIAL ON DATA-BASE ORGANIZATION

Robert W. Engles

The purpose of this presentation is to clarify certain issues of data-base support. The main issues are data independence, security, integrity, search, and the integrated data base. The intent of the presentation is tutorial, and the viewpoint is that of a systems programmer.

The first part of the presentation is an introduction, which includes data-management history, trends, and terminology. The second section presents a theory of operational data based on the notions of entity sets and data maps. The third section is an exposition of data-bank design, emphasizing structure, search, and maintenance. The fourth section shows why data independence is a necessary feature of a viable data base support.

Key Words:

Data independence
Data integrity
Data management
Data security

Integrated data base
Search
Systems programming

Summary of the Presentation

A TUTORIAL ON DATA BASE ORGANIZATION

by

Robert W. Engles
International Business Machines Corporation
Systems Development Division, Poughkeepsie, New York

given at

GUIDE 29
November 3-7, 1969
Denver, Colorado

A TUTORIAL ON DATA BASE ORGANIZATION

(Summary of the Presentation)

by

Robert W. Engles

International Business Machines Corporation
Systems Development Division, Poughkeepsie, New York

The heart of an information system is its files or data base. The purpose of this tutorial is to clarify certain issues of data base support. The main issues are data independence, search, and the integrated data base. Data organizations are described and the problems involved in the representation, storage, and retrieval of information are analyzed from a programming point of view.

The first section of the tutorial is an introduction which includes data management history, trends, and terminology. The evolution of data management software is viewed in terms of the growing distinction between file organization (logical structure), and data organization (physical structure). The issues of search, data independence, and the integrated data base are introduced and their relationships explained. Data access is distinguished from data organization. A system structure is described to provide a framework for the definition of data base concepts and terminology. Units of storage (volume, physical record, etc.) are distinguished from units of data, and system data units (data bank, data set, extent, block, stored-record, and data element) are distinguished from application data units (file, logical record, and field).

The second section of the tutorial presents a theory of operational data based on the notions of entity sets and data maps. Entities are the things in the real world about which we record facts. Facts are relationships, and data maps are a means of defining relationships. Twelve types of data maps are defined, and data base organization is viewed as the process of defining, representing, storing, and maintaining data maps. The regular organization and the inverted organization are defined as the two major types of data organizations. Various types of retrieval requests are defined, and the need for both the regular and the inverted organizations is developed by means of examples.

The third section of the tutorial is an exposition of data bank design, emphasizing structure, search, and maintenance. Starting with the problems of representing and updating complex data maps, this section explores various data organizations capable of representing complex structure. Hierarchical, multi-list, variable- and fixed-symbol list organizations are described and compared. The principles of entity set organizations are defined, and the requirements for handling networks of relationships are explained in terms of a product structure example.

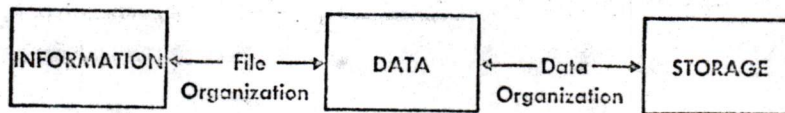
The final section of the tutorial is about data independence. Starting with a list of data dependencies, this section presents the reasons why data independence is widely considered a necessary feature of viable data base support. Types of data independence are classified, evaluated, and related to other issues such as security and integrity. The presentation emphasizes the need for a logical data organization against which application programmers can define file organizations, and against which data base administrators can define data organizations. The previously described notion of the entity record set is suggested as the basis of such a logical data organization. The presentation concludes with requirements for a data description language for data base administrators. The opinions expressed in the tutorial are personal and do not represent a corporate position.

INTRODUCTION

A Tutorial on Data Base Organization

- **Introduction**
History, trends, and terminology
- **Operational Data**
What it is and how it is organized
- **Data Banks**
Their structure, search, and maintenance
- **Data Independence**
When, where, and why - who should specify what

SECTION I



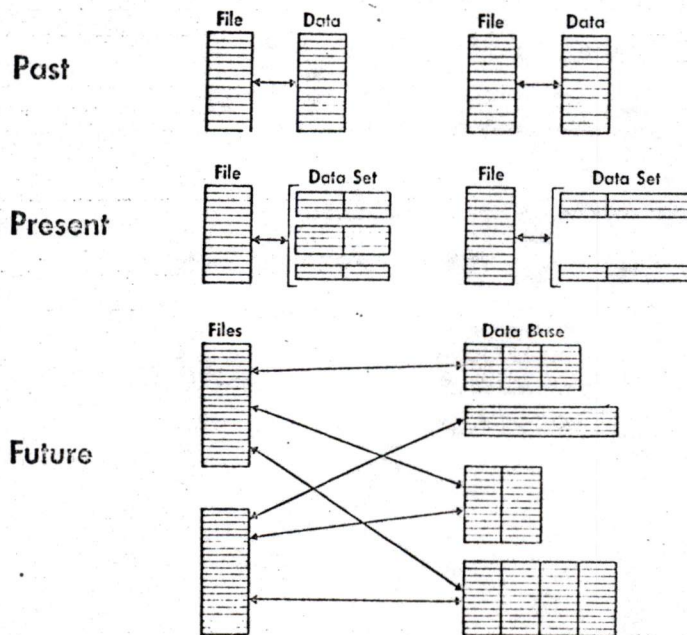
Information: The Meaning Assigned to Data by Known Conventions.

Data: Any Representations to Which Meaning May be Assigned.

Storage: A Device into Which Data Can be Inserted, in Which it Can be Retained, and From Which it Can be Retrieved.

Data Organization: The Correspondence Between the Structure of Data and the Structure of Storage.

File Organization: The Correspondence Between the Information Structure and the Structure of the Data.



Present Levels

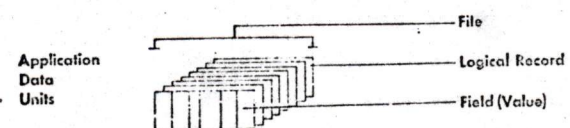
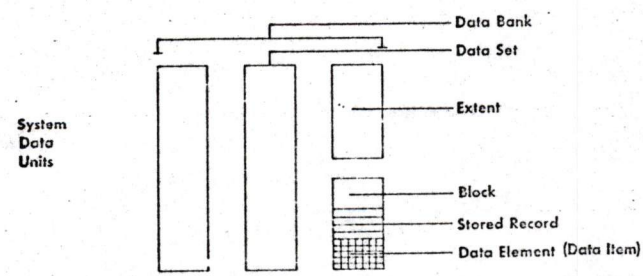
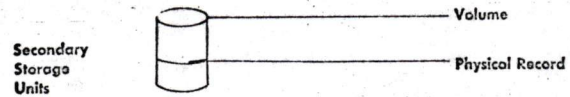
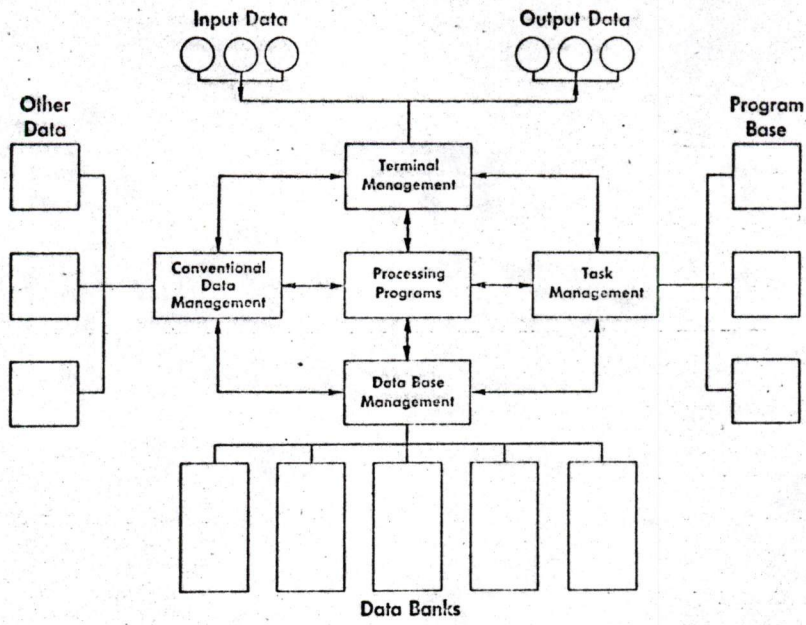
1. Input/Output Control (e.g., DOS)
2. Data Set Control (e.g., OS/360)
3. Data Base Control (e.g., IMS)

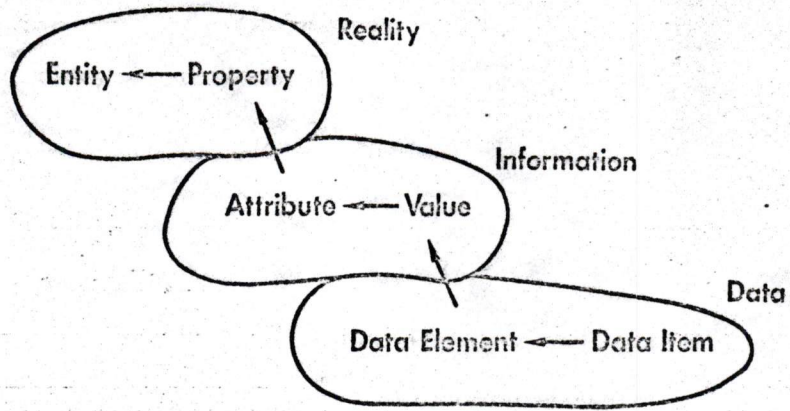
Major Issues

- Search
- Security and Integrity
- Data Independence
- Integrated Data Base

Terminology

- Enterprise
- Operational Data
- Entity
- Event
- Entity Record
- Event Record





Reality: Entity Sets
 Entities
 Properties

Information: Entity Record Sets
 Entity Records
 Attributes (Values)

Data

Application: Files
 Logical Records
 Fields (Values)

System: Data Base
 Data Banks
 Data Sets
 Space & Form Extents
 Blocks
 Stored Records
 Data Elements (Data Items)

SECTION II

An Entity Set is a Collection of Similar Entities, i.e., Things that Have the Same Kind of Properties

For Each Entity Set, There is an Identity Attribute

The Values of an Identity Attribute are Unique Entity Identifiers, for Example, Part Numbers

A Fact is Represented by a Correspondence Between Values of Two Attributes, One of Which is an Identity Attribute; for Example: The Quantity-on-Hand of Part# 3256 is 800

A Data Map is the Totality of Relations Between the Values of an Identity Attribute and the Values of Another Attribute Associated with the Entity Set
Part# → Quantity-on-Hand

THING # → CLASS

THING # is the identity attribute
for the set of THINGS

The values of THING # are:
1549, 1648, 1985, 2003, ...

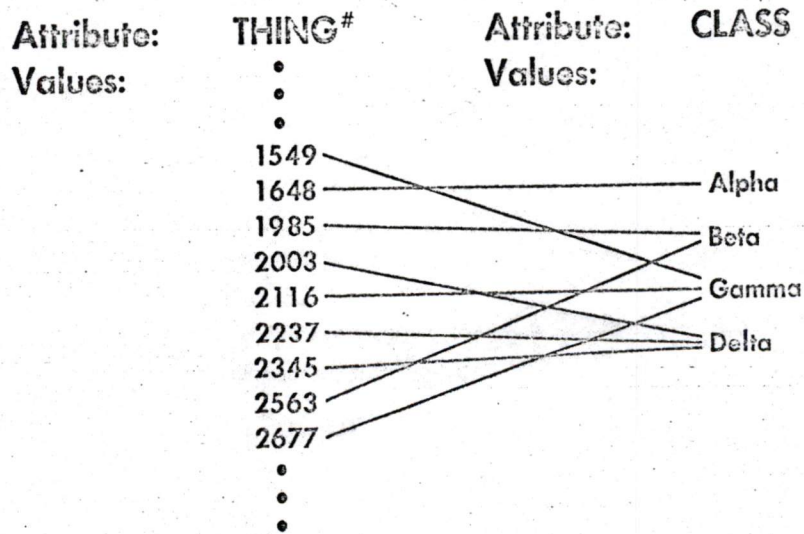
These values are the unique entity
identifiers for the set of THINGS

CLASS is an information attribute
defined for the set of THINGS

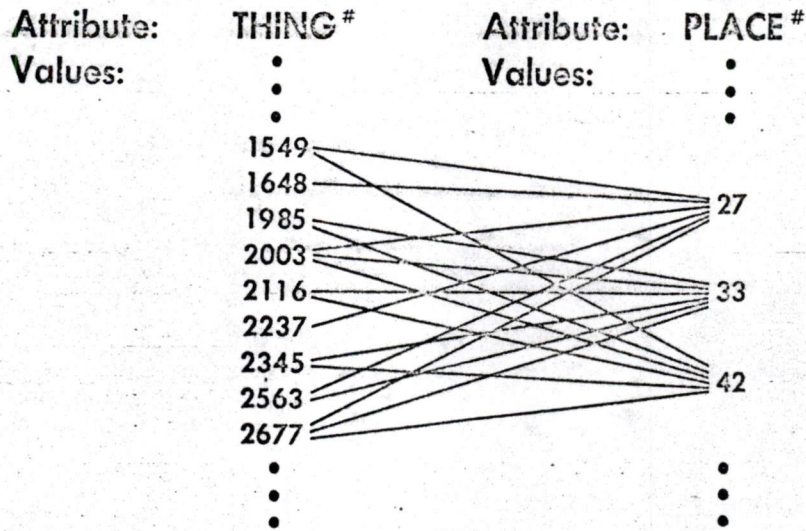
The values of CLASS are:
Alpha, Beta, Gamma, Delta

These values are not entity
identifiers for some other entity set.

Simple, Nonstructural Data Map



Complex, Structural Data Map



A Data Map Specifies the Relations Between Two Sets of Attribute Values

Let V Denote a Set of Any Type

Let E Denote a Set of Entity Identifiers

Let W Denote a Set of Values that are not Entity Identifiers

Consider Maps of the Form: $E \rightarrow V$

There are Three Types of Maps:

$E \rightarrow W$ $E \rightarrow E$ $E \rightarrow E'$

To Each Element of E, A Map may Assign None, One, or Many Elements of V

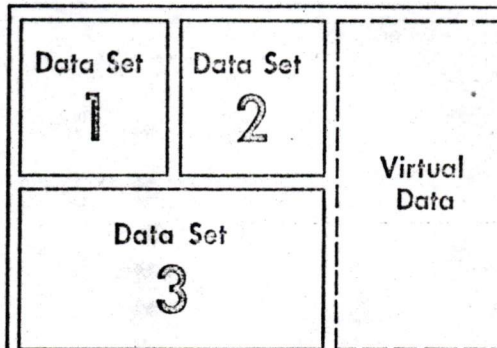
There are Four Types of Mappings: Simple/Simple, Simple/Complex, Complex/Simple, Complex/Complex

Attributes of THINGS

THING#	NAME	CLASS	STATUS	COLOR	DATE
⋮					
11549	Joe	Beta	Out	Red	122032
11648	Sam	Alpha	Out	Blue	081352
11985	Bob	Alpha	In	White	101547
12003	Ray	Gamma	Out	Red	012853
12116	Jim	Beta	In	Green	042939
12237	Joe	Delta	In	Black	081148
12345	Max	Gamma	In	White	122032
12563	Jim	Beta	Out	Yellow	111140
12677	Irv	Delta	In	Blue	033045
⋮					

Part of an Entity Record Set of simple data maps in a regular, fixed-length, sequential organization representing facts about the set of entities called THINGS

The Entity Record Set



Illustrating Vertical and Horizontal
Partitioning of an Entity Record Set into
Three Data Sets

Data Set Organizations

	Not Indexed	Indexed
Sequential	1	2
Random	3	4

Attribute Requests:

In Regard to Things, What is the Color of 12345 ?

In Regard to Things, What is the Name, Class
and Status of 11549, 12003, 12237, and 12677 ?

Classification Requests:

List the Things with Color=Red

List the Things with Date > 042939

Classification and Attribute Request:

What is the Name and Status of Things with
Class=Alpha ?

Compound Classification Request:

List the Things with Color = Blue and Class = Alpha

Compound Classification and Attribute Request:

What is the Thing#, Name, and Class of Things with Status = In and Color = Green or Date < 123140 ?

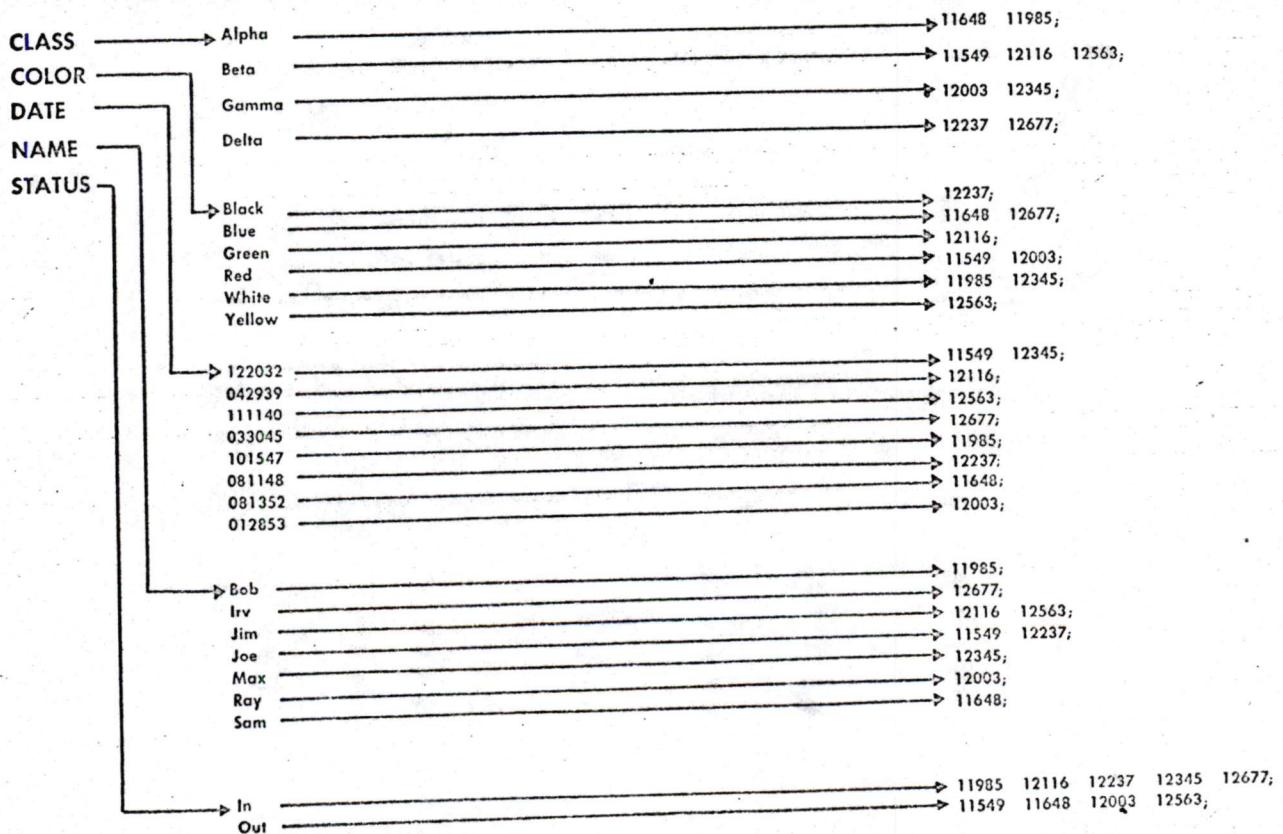
Classification and Function Request:

Count the Things with Status = Out

Compound Classification, Attribute, and Function Request:

What is the Name and Date of Things with Class = Beta and Status = In; Sort Ascending on Name

THINGS in an Inverted Data Organization



SECTION III

Complex Data Maps

**Complex/Simple and Complex/Complex
Retrieval and Update
Representations**

Structural Data Maps

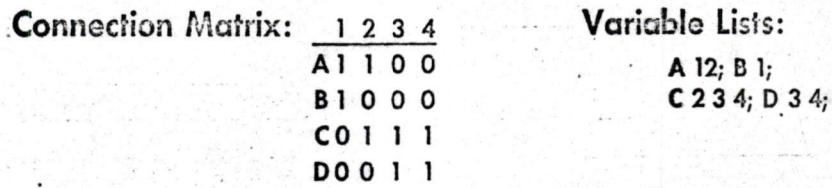
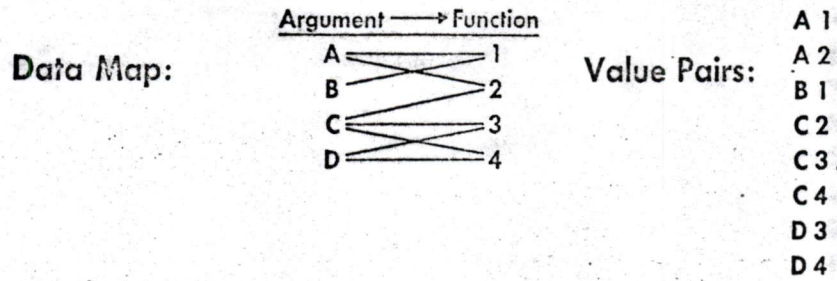
**E→E' and E→E
Related Data Maps
Implied and Derived Data Maps**

Update: Additions, Changes, and Deletions

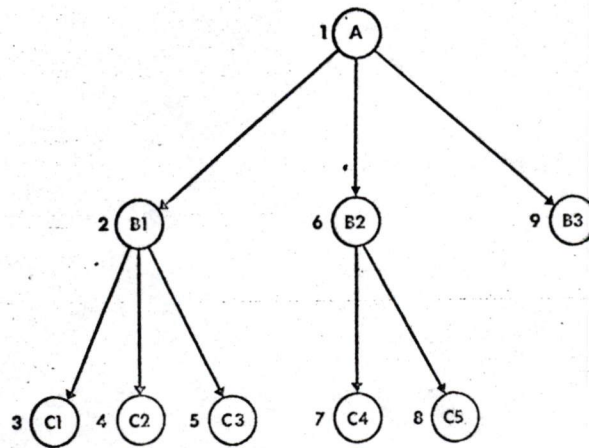
All the Requirements of Retrieval Plus the Problems of:

- . **Storage Management**
- . **Integrity / Validity**
- . **Interlock / Deadlock**
- . **Consistency of Related Data Maps**
- . **Copies for Recovery and "As Of Retrieval"**

Three Methods of Representing a Complex Data Map Without Pointers



Linear Representation of a Tree Structure



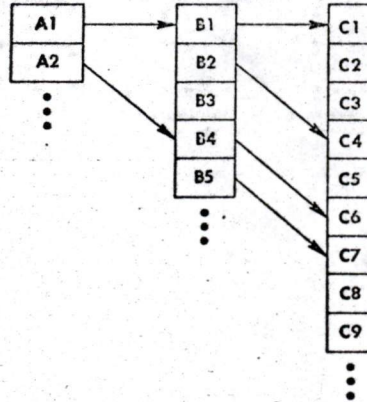
A(B1(C1, C2, C3)B2(C4, C5)B3)

Contiguous Hierarchical Data Organization

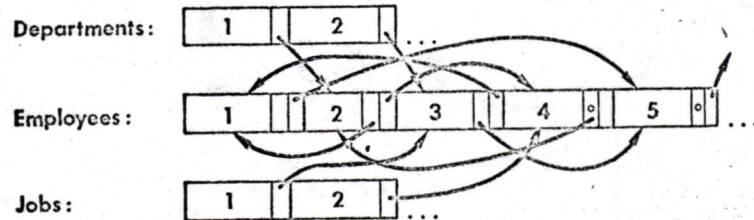
A1	B1	C1	C2	C3	B2	C4	C5	B3
A2	B4	C6	B5	C7	C8	C9		

⋮

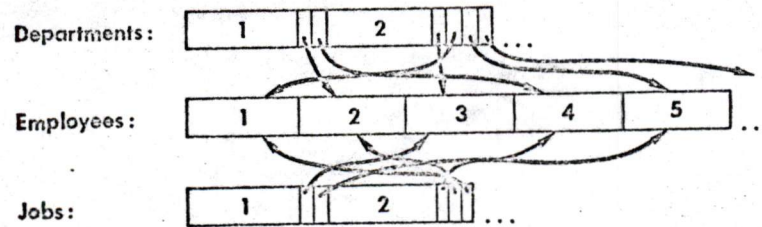
Noncontiguous Hierarchical Data Organization



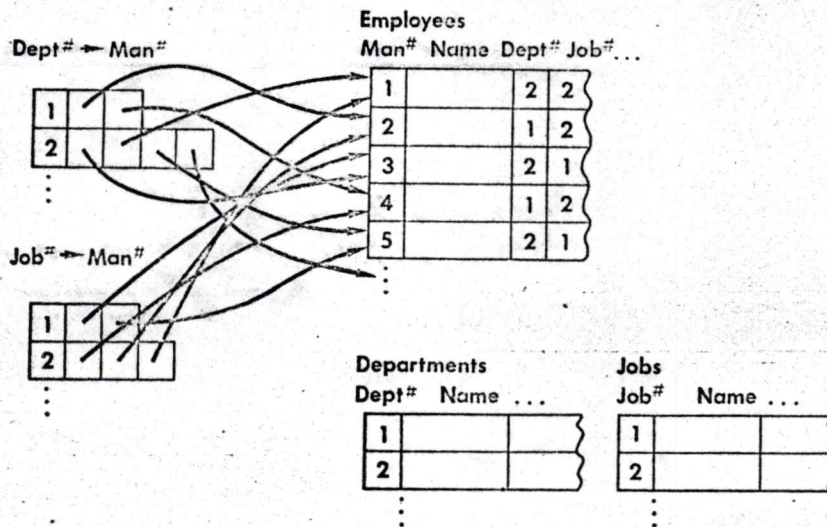
Chained List Organization



Variable Pointer List Organization



**Regular Organization with Secondary Indexes used to
Represent all Complex Data Maps**



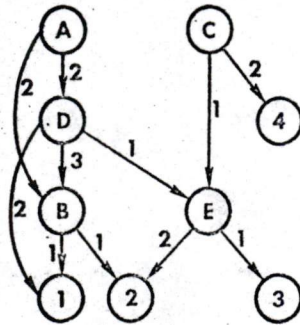
**Suggested Principles of Data Base
Organization**

1. A Data Bank Represents a Network of Relations among Entity Sets.
2. Data Banks Will Change.
3. Entities Must Be Uniquely Identified in the Context of the Entity Set.

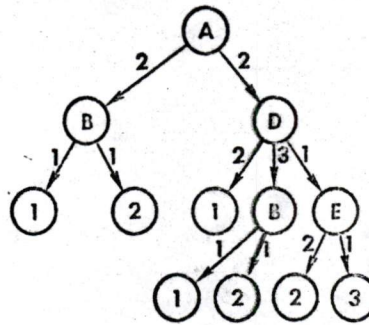
Suggested Principles of Data Base Organization

4. Separate Data Access from Data Organization.
5. Separate Data Organization from File Organization.
6. Separate Complex Maps from Simple Data Maps.
7. Whenever There Is a Complex Data Map of the Type Whose Inverse is Also Complex, This Type of Relation Defines Another Entity Set.

1. File in B/M Order



2. Tree Formed by Explosion of Part A



Pairs

Pair #	Quantity
A B	2
A D	2
B 1	1
B 2	1
C E	1
C 4	2
D B	3
D E	1
D 1	2
E 2	2
E 3	1

Parts

Part #	Type	Description	QOH
A	.	.	.
B	.	.	.
C	.	.	.
D	.	.	.
E	.	.	.
1	.	.	.
2	.	.	.
3	.	.	.
4	.	.	.

SECTION IV

The Declaration of Data Independence

When, in the course of event processing, it becomes necessary for us to dissolve the bonds which have connected programs and data, we should declare the causes which impel us to this separation.

We hold these truths to be self-evident, that all programs are created equal, that they are endowed by their creator with certain inalienable rights, that among these are sufficient storage, protection, and **DATA INDEPENDENCE**.

A Program can be Bound to its Data at the Following Times:

- Writing of the source program--implying or specifying the data descriptors
- Compiling of the object program--implying or including the data descriptors
- Linking with precompiled tables or routines containing the data descriptors
- Opening of the file--associating the file description with the data descriptors
- Accessing of a data element or a record of data elements--dynamically utilizing the descriptors

In Regard To Using A Data Set:

1. How Is It Accessed? i.e.,
 - How Is It Located?
 - What Access Method Should Be Used?
 - Is Access Constrained By Device Characteristics?
2. Where Is It? i.e.,
 - What Volume(s) Is It On?
 - What Device Is The Volume On?
 - What Computer Is The Device On?

In Regard To Using A Data Set:

3. What Is It? i.e.,
 - How Is It Related To The File?
 - What Is The Data Set Organization?
 - What Are The Record Storage Parameters?
4. How Is The Access Method Used? i.e.
 - Buffer Requirements?
 - Blocking/Spanning?
 - Interlock Procedures?
 - Control Blocks And Linkages?

In Regard To Using A Single Data Item:

1. How Is It Accessed? i.e.,

Is It Stored Or Computed?

What Is The Search Algorithm?

Is There An Index?

What Do I Have To Do And Know To Use The Index?

2. Where Is It? i.e.

Where Is The Data Item In The Segment Or Record?

Where Is The Segment Or Record In The Data Set?

What Is The Name Or Extent Of The Data Set?

In Regard To Using A Single Data Item:

3. What Is It? i.e.

How Do I Tell If It's Null?

What Is Its Length?

What Is The Unit Of Measure Of The Value?

What Type Of Value? (number, string, boolean, pointer)

4. How Is The Value Represented? i.e.

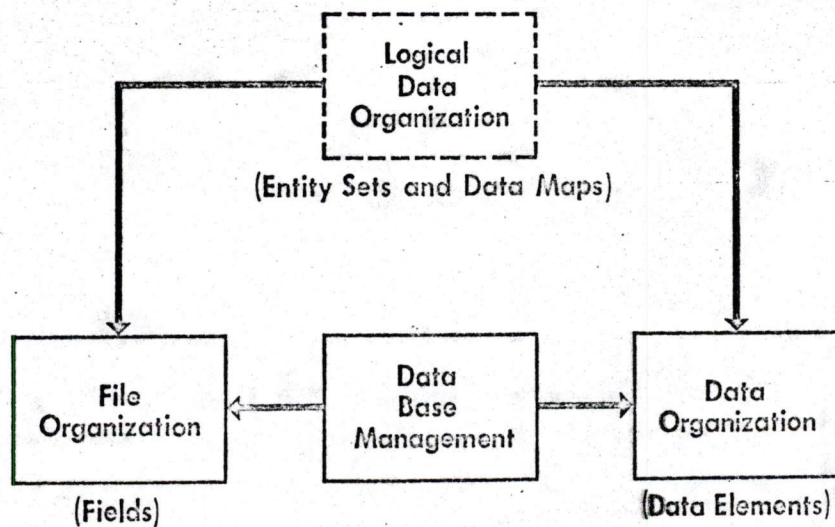
What Is The Code?

What Is The Format?

What Is The Level Of Representation?

In Regard to Update:

1. What checks should be made pertaining to:
 - Authorization for the change?
 - Validity of the new data?
 - Consistency of related data?
2. What other data should be changed?
 - Other copies of the values?
 - Related data maps?
 - Indexes or inverted data maps?
3. What are the procedures for:
 - Interlock/Deadlock
 - Copies for recovery and history?
 - Handling additions and deletions?
 - Allocating and freeing storage?



A Data Description Language for the Data Base Administrator to Specify:

- What the attribute values designate, i. e., identity, structural, or simple facts; status, summary or historical facts.
- The materialization type, i. e., direct, indirect, factored, computed, or coded; and depending on the type, the data set(s) and byte offset, function names, etc.
- The representation of the values, i. e., binary or decimal, integer or real, digit or character string, length, justification, padding, scale, units, etc.

A Data Description Language for the Data Base Administrator to Specify:

- The security and integrity procedures, i. e., authorization tables, edit masks, range limits, related attributes, update rules, function names, etc.
- Search mechanisms, i. e., whether the attribute values can be used as keys, whether the attribute is to be indexed, the type of index, search technique, etc.

Reasons for Data Independence

Allows data base administrator to make changes in the content, location, representation, and organization of a data bank without reprogramming the application programs.

Allows supplier of data processing equipment and software to introduce new technologies without reprogramming of customers' applications.

Facilitates data sharing by allowing the same data to appear to be organized differently for different application programs.

Simplifies application program development to facilitate development of programs for interactive data base processing.

Provides centralization of control needed by data base administrator to ensure the security, integrity, and consistency of the data base.



IBM

International Business Machines Corporation

Feb 14 1971

0900 7-
FEB 22 RECD

Poughkeepsie, New York 12602

Office of Vice President

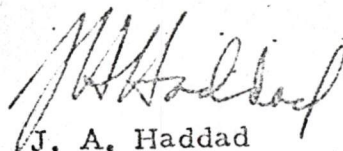
February 4, 1971

TO: Computer Science & Engineering Board Members

SUBJECT: CS&EB Data Security Study

Enclosed is a Prospectus for a Data Security Study to be undertaken by the Computer Science and Engineering Board. Discussion of this item will be on the agenda during the February CS&EB meeting.

Your comments on the Prospectus and ideas for possible sponsors are solicited.



J. A. Haddad

JAH:am

Attachment

NATIONAL ACADEMY OF SCIENCES
(Computer Science and Engineering Board)
PROSPECTUS
Data Security Study

I. Introduction

In view of the rapidly growing importance of data security as a problem with far reaching social, financial, and legal implications, it is appropriate that there be established a committee of the Computer Science and Engineering Board to conduct a study and prepare recommendations for further action by that Board.

The purpose of this study will be to assess the information on data security available to computer designers, users, and the general public, and to recommend policies for generating the required information where it is needed. This study will be concerned with the security of data used and stored in electronic data processing systems which are commercial in nature. *find better way*

This effort will study thoroughly the following three items:

1. Identification of the possible types of data security violations and contingencies.
2. Identification of security measures which can be provided through physical means or operational procedures.
3. Assessment of the availability and adequacy of security measures in computer hardware, supporting programs, and operational procedures to guard against possible violations and contingencies.

II. Statement of Problem

At the outset, data security must not be confused with data privacy.

It is important to understand that privacy and security are not synonyms nor is one a part of the other. Privacy is the claim of individuals, groups

or institutions to determine when, how, and to what extent information about them is communicated to others,¹ and includes questions such as

what data ought to be put into a computer system at all. Security is protecting the integrity of the data once it is in (or being put into) the

system by such means as physical protection (i. e. locked rooms) environmental protection (i. e. electromagnetic shielding), encrypting of data,

operating system procedures, etc. Defining a computer system as a

collection of people, devices, processes, and procedures assembled to

process information, the security of this information is then a function of

the measures taken by each element of the system. The elements of this

protection must be provided by both the computer industry and the users

of the systems.

We may think of data security as the provision of "hooks" and features in hardware and software which will allow users to apply system engineering

principles and obtain configurations, procedures, and operations which

implement the desired profile of security in a given environment, application,

and set of ^a threats extant now or in the future.

Extensive means for protecting data from unauthorized disclosure

(whether accidental or intentional), from modification, and from destruction,

have been limited until quite recently to a few specialized computer systems.

However, the management of major enterprises are aware of their dependence

¹Westin, A. E., "Privacy & Freedom", Atheneum, New York 1967

*includes concern for integrity of data
public data*

*Wishler
+ contingency*

integrity means more than

→ see also integrity on next page

3
on the integrity and continued availability of data in their systems. The growth of these concerns suggests the need for an objective evaluation of the ability of the users of computer systems to determine and to achieve an adequate level of data security now or in the reasonably near future.

The efficiency and effectiveness of many federal, state, and local government functions will depend on the timely availability of information. Unwise or malformed control (i. e. legislative, regulatory, administrative, etc.) enacted in response to the security issue, may so limit the use of electronic data processing as to preclude its use in many of these important applications and can adversely and seriously impact the operation of essential commercial enterprises, such as retail credit. Conversely, unless necessary legal constraints are provided, taking into account the limitations of today's and tomorrow's technologies in providing data security, it may be impossible to establish important new computer applications such as regional, state, or national banks of medical records.

III. Plan of Action

The Computer Science and Engineering Board (CSEB) will establish a committee composed of representatives of industry, academics, users, the legal profession and the general public to study this problem. The CSEB as a part of NAS, has a unique capability of assembling open and proprietary information from both the public and private sectors. Inasmuch as there is relatively little information available in the open literature, access to proprietary information will be important to making an accurate assessment of the availability and adequacy of security measures.

The committee will form appropriate panels to study the three areas mentioned above and will issue a report at the conclusion of the study. The report will discuss the committee's assessment of the availability of knowledge in the three areas and will make recommendations for improving the quality and quantity of information available. It is expected that the study will require one year to perform. Followon efforts may be recommended if the committee thinks they are desirable.

On the Defense

Computer Companies Are Hauled Into Court By Flurry of Lawsuits

Users Say Systems Erase Data, Create Confusion And Cause Red Ink to Flow

'Who, Us?' Say the Makers

By WILLIAM M. CARLEY

Staff Reporter of THE WALL STREET JOURNAL

When Scientific American magazine decided to use a computer to handle its files on subscribers, it hired System Development Corp., of Santa Monica, Calif., to plan the operation, select the proper computer and write the "software" instructions that would guide the computer in its tasks.

Chaos ensued. "The system failed to perform many essential functions; much of what it did do was inaccurate," Scientific American says. When a subscriber notified the magazine of a change of address, for example, the computer was supposed to create a new record for that subscriber and render the old one inaccessible. In fact, the computer rendered all records inaccessible, wiping out service to that subscriber, the magazine charges.

Some readers revolted and canceled subscriptions. Others who kept their subscriptions didn't get their magazines and refused to pay. Still other bewildered readers got duplicate copies.

Scientific American soon found it didn't even know its correct circulation. To make sure it had enough copies, it resorted to costly printing of more magazines than it needed. Meanwhile, because readers were canceling subscriptions, the magazine was losing advertising revenue.

This, at least, is the story told in court papers filed by Scientific American. The magazine is suing System Development in Federal court for \$2.5 million in damages it says it suffered. System Development vigorously denies any wrongdoing.

Tip of an Iceberg

Whoever is at fault, Scientific American isn't alone. A rapidly growing number of companies are finding their gleaming new computers won't work, or work wrong. And like Scientific American, they're suing the computer makers or the software suppliers for damages.

Besides System Development, those now under legal fire for computers that allegedly won't perform properly are International Business Machines Corp., Burroughs Corp., Sperry Rand Corp.'s Univac division and Xerox Data Systems Inc., a subsidiary of Xerox Corp.

Lawyers expect still more suits to be filed. "As far as legal battles over faulty computers go, I think we're seeing only the tip of the iceberg," says William Fenwick, a New York attorney who specializes in computer law.

Are computers really that bad? Certainly sometimes.

Computer companies admit they are having troubles and say privately that computer systems have grown so complex—and short-lived—that it's all but impossible to know how to fix everything that can go wrong with them. "As soon as we develop good systems checks for one generation of computers the industry leaps to the next generation, and sometimes the checks can't fully match the greater complexity involved," says a vice president for one computer company.

Critics say that's certainly true, but they add that there are other reasons for the sudden onslaught of court suits. Like the "know-it-all" air that customers say computer companies exude when dealing with clients who know nothing about computers. "The computer companies assure customers, 'We'll give you a whiz-bang system, we know all about it, you can depend on us,'" says Roy Freed, a Boston attorney. "All too often the customer relies on this and fails to negotiate a carefully drawn contract including detailed specifications that the computer must meet. When something goes wrong, the gullible customer is left holding the bag."

Major computer companies deny that they assume such an attitude in dealing with clients. They contend that many computer snafus are the result of negligent customers failing to carefully follow instructions on the care and feeding of computers.

Fed-Up Consumers

Lawsuits also may be increasing simply because the public is getting increasingly intolerant of computer errors. "Look at the mistakes in your department store bills," says one lawyer involved in a computer case. "People are just getting fed up with this sort of thing." When enough consumers get fed up, of course, complaints deluge a corporation and business may even fall off. That's when computer companies receive invitations to defend themselves in court.

The suits are not small potatoes. The one filed last month by TWA against Burroughs asks for \$70 million in damages. TWA contracted with Burroughs for a computerized passenger reservations system. "During the negotiations of the contract," TWA alleges, Burroughs represented itself as "a pioneer in the design and development of large-scale military and commercial electronic data processing systems" with "extensive experience in system design, hardware development and software design." Burroughs also claimed, the suit charges, that the system it was peddling TWA was "reliable and flexible and had been proven in many previous applications."

In fact, TWA charges, the Burroughs system "has proven to be unreliable, incomplete and defective with resulting breakdowns and failures, and the system is totally unfit for TWA's purposes." A spokesman for the airline, who says the Burroughs system never did go into operation for TWA, declines to specify just how it failed.

Burroughs has denied TWA's charges, saying that it "has in no way made any misrepresentations" and that "the Burroughs equipment meets or exceeds all requirements and intended use conditions of the TWA system." Burroughs has filed a countersuit for the \$11.5 million it says TWA owes it for the computer system.

In a suit against Univac, United Engines

Please Turn to Page 21, Column 3

Inc., a small Shreveport, La., distributor of truck diesel engines, says it ran into two problems. First, the company says it discontinued its manual accounting system in anticipation of the computer's arrival Oct. 1, 1967. That created problems, since the computer arrived in December. And when it did arrive, the company says, it didn't work right, either because of programming errors or because it was inherently incapable of performing the assigned chores.

The net result, the company says, was that its accounting was "thrown into confusion, chaos and catastrophe." There were countless inaccurate invoices which would not balance with customers' own tabulations of their accounts. Customers were returning invoices as fast as United Engines could mail them. Many customers refused to pay until their invoice was properly balanced. United Engines lost count of inventory and therefore of inventory control, "with the result that inventory increased more than \$250,000 above average."

Inventory soared and sales dropped. United Engines says it encountered an "acute cash shortage" and had to borrow money to meet its payroll.

United Engines says it tried everything to solve the snafu. When the computer was late, a United Engines executive pleaded with Univac's local representative for delivery. When that didn't work he called a Univac man in New Orleans, then a Univac executive at the company's Philadelphia headquarters, and finally vainly tried to phone the president of Sperry Rand, Univac's parent company, at his home.

Nothing Helped

When the computer arrived and then allegedly couldn't do the job, United Engines says it tried to use a Univac in a local hospital as a sort of pinch-hitter. United Engines also worked employees overtime, hired new ones and flew executives into Shreveport headquarters from various United Engines offices to help rectify accounts. None of that solved the problem, the company says. United Engines has switched to another computer supplier and is suing Univac for \$271,000 in alleged damages.

Univac declines to comment on the case.

In some cases, computer users charge that computers don't work because of poor service and maintenance. Megsystems Inc., a New York firm that leases computers and then rents time on them to various customers, is suing Xerox Data Systems on these grounds. That suit alleges that XDS provided "inadequate, poor, inexperienced, negligent, incompetent and unskillful personnel to service and maintain the computer." Maintenance workers simply couldn't keep the computer going, the suit alleges.

"It was inoperational for days, in some cases weeks," says a Megsystems attorney. "Things got so bad that one of our customers had to be flown to the West Coast so the computer work could be done on XDS equipment there."

Win Some, Lose Some

Another problem involved a serviceman cleaning computer discs, on which data are stored. "He used the wrong solution—solution A instead of B—and as a result he totally destroyed our customer's records on that disc," the Megsystems attorney says.

XDS says the Megsystem suit is "without merit."

In the handful of cases that have come to trial and been decided, computer companies have won a few and lost a few.

When Lithonia Lighting Inc., a Conyers, Ga., maker of industrial lighting fixtures, became dissatisfied with its leased Honeywell computer, it tossed the computer out prior

to expiration of the lease and ordered IBM equipment. Honeywell sued for the profits it would have made on the balance of the lease.

Lithonia Lighting counterclaimed in court that Honeywell breached the lease contract by, among other things, providing a defective reader of punched cards. But Federal Judge Newell Edenfield ruled that "it was finally discovered that the trouble lay in a voltage meter which Lithonia itself had incorrectly wired when the system was installed." The judge held that Honeywell was entitled to \$159,922.

In an IBM case, however, the computer maker lost—at least the first round. An IBM subsidiary, Service Bureau Corp., provided a computerized inventory control system for Clements Auto Co., a Minnesota distributor of auto and electrical products, and assured it the system would "provide iron clad controls to insure accurate reports."

The Old Dust Method

But because a device used to feed data into the computer was allegedly error-prone, the system was riddled with mistakes. Clements auto executives testified in court that computer-produced inventory reports showed no relation to the actual amount of stock on the shelves. A Clements officer also testified that some computer reports were so voluminous as to be unusable.

"You have no idea of the amounts of paper that thing ground out. It was just more paper than the people could possibly get through," he said.

In ruling on the case, Federal District Court Judge Miles W. Lord noted that in earlier years one criterion for determining slow moving items in inventory at Clements was the level of dust on the shelves. "After the system had been in operation for three years and after hundreds of pages of reports had been turned out by IBM, at an expense to Clements Auto of (hundreds of thousands of dollars), Clements still had no more reliable guide to the obsolescence of its inventory than the level of dust upon the merchandise," the judge said. The judge granted \$481,000 in damages to Clements Auto.

IBM appealed the ruling to the Court of Appeals in St. Louis and has argued its case before that court, but a ruling hasn't yet been issued.

When they can, computer makers often try to settle suits out of court, if only to avoid bad publicity and the expense of litigation. That's what happened when Band-it Co., a Denver manufacturer of industrial fasteners, sued National Cash Register Co., for \$35,000, alleging that NCR had provided a computer that didn't even perform as well as Band-it's existing accounting system. NCR won't disclose the amount for which Band-it settled out of court.

Computer makers don't like to talk about lawsuits against their products. Companies including IBM, Burroughs and Honeywell won't disclose to reporters which courts they're being sued in, or who is suing them, even after such suits become a matter of public record in the various courts.

Cheer Up, Computer Users: Tomorrow Will Be Worse

By a WALL STREET JOURNAL Staff Reporter

If computer snafus seem bad now, just wait. They promise to get much worse.

William A. Fenwick, a New York attorney who is an expert on computer law, notes that computers are increasingly being interconnected so they can "talk" to each other. "This raises the possibility that if something goes wrong with a retailer's computer that's hooked up to a supplier's, the errors in the retailer's machine can creep into the supplier's and disrupt both companies' businesses," Mr. Fenwick says.

Another problem, the lawyer notes, is that companies are increasingly changing their computers to "on line" systems, meaning that the computer is constantly accepting data and turning out information on which management decisions are based, which in turn are fed back into the computer. If an error creeps into such an "on line" system, it can rapidly have a cumulative effect and spell chaos for a company's data and decision-making processes.

"It can turn everything topsy-turvy in a real hurry," says Mr. Fenwick.

Security controls in the ADEPT-50 time-sharing system

by C. WEISSMAN

System Development Corporation
Santa Monica, California

*"Authority intoxicates/And makes mere
sots of magistrates"—Butler*

FOREWORD

At present, the system described in this paper has not been approved by the Department of Defense for processing classified information. This paper does not represent DOD policy regarding industrial application of time- or resource-sharing of EDP equipment.

INTRODUCTION

Computer-based, resource sharing systems are, and contain, things of value; therefore, they should be protected. The valuables are the information data bases, the processes that manipulate them, and the physical plant, equipment, and personnel that form the system plexus. An extensive lore is developing on the subject of system protection.^{1,2} Petersen and Turn³ discuss in considerable detail the substance of protection of non-military information systems in terms of threats and countermeasures. Ware^{4,5} contrasts "security" and "privacy" for viewing protection in military systems as well. This paper describes the security controls implemented in the ADEPT-50 time-sharing system⁶—a resource sharing system designed to handle sensitive information in classified government and military facilities.*

Our approach to security control is based on a set

* Development of ADEPT was supported in part by the Advanced Research Projects Agency of the Department of Defense.

theoretic model of access rights. This approach appears natural, since the important objects of security are sets of things—users, terminals, programs, files—and the operators of set theory—membership, intersection, union—are easily programmed for, and quickly performed by, computer. The formal model defines time-sharing security control of user, terminal, job and file security objects in terms of equations of access based upon their security profiles—a triplet of Authority, Category, and Franchise property sets. The correspondence of these properties to government and military Classification, Compartments, and Need-to-Know is demonstrated. Implementation of the model in the ADEPT-50 Time-Sharing System is described in detail, as are features that transcend the model including initialization of the security profiles, the LOGIN decision procedure, system integrity checks, security residue control, and security audit trails. Other novel features of ADEPT security control are detailed and include: automatic file classification based upon the cumulative security history of referenced files; the "security umbrella" of the ADEPT job; and once-only passwords. The paper concludes with a recapitulation of the goals of ADEPT security control, approximate costs of implementation and operation of the security controls, and suggested extensions and improvements.

Historically, protection of a sensitive computer facility has been attained by limiting physical access to the computer room and shielding the computer complex

from electromagnetic radiation. This "sheltered" approach promotes one-at-a-time, batch usage of the facility. Modern hardware and software technology has moved forward to more powerful and cost/effective time-shared, multi-access, multiprogrammed systems. However, three features of such systems pose a challenge to the sheltered mode of protection: (1) concurrent multiple users with different access rights operating remote from the shielded room; (2) multiple programs with different access rights co-resident in memory; and (3) multiple files of different data sensitivities simultaneously accessible. These features appear to violate traditional methods of accountability based upon a single user (or multiple users with like clearances) operating within strictly controlled facilities. The problem is of such magnitude that no time-sharing system has yet been certified for use in the manner described! However, some multi-access systems are in operation in a classified mode,^{7,8} and a number of design approaches have been suggested.^{9,10,11,12}

In addition to the usual goal of building an effective time-sharing system,¹³ the ADEPT project began with a number of security objectives as well:

1. Build a security control mechanism that supports heterogeneous levels and types of classifications.
2. Design the security control mechanism in such a manner that it is itself unclassified until primed by security configuration parameters, a point strongly supported by Baran¹⁴ regarding communications security.
3. Construct the security control mechanism as an isolated portion of the total time-sharing system so that it may be carefully scrutinized for correctness, completeness, and reliability.
4. Do the above in as frugal a manner as possible, considering costs to design, fabricate, and operate. Good system performance is our principal criterion in selecting among alternative technical solutions, as noted by the author elsewhere.¹⁵

In approaching our task, we recognize security as a total system problem involving hardware, communication, personnel, and software safeguards. However, our focus is primarily on monitor software, and its interfaces with the other areas. This view is not parochial: our hardware is a standard IBM 360 model 50; communication security is an established field of study with considerable technological know-how;¹⁴ and the policy, doctrine, and procedures for personnel behavior in classified environments are extensive, with legal founda-

tions. Thus, our only degree of freedom is the control we build into the time-sharing executive software.

A security control formalism

A formal model of software security control for access to sensitive portions of ADEPT is developed here.

Security objects

Four kinds of security objects are to be managed by our model: user, terminal, job, and file. Let u denote some user; t some terminal; j some job; and f some file.

Security properties

Each security object is described by a security profile that is an ordered triplet of security properties—Authority (A), Category (C), and Franchise (F). Authority is a set of hierarchically ordered security jurisdictions. Category is a set of discrete security jurisdictions. Franchise is a set of users licensed with privileged security jurisdiction.

The property "Authority" is defined as a set A, where

$$A = \{a^0 < a^1 < \dots < a^n\} \quad (1)$$

and the specific members, a^i , of the set are security jurisdictions hierarchically ordered.

"Category" is a discrete set of specific compartments, c^i ,

$$C = \{c^0, c^1, \dots, c^p\} \quad (2)$$

Compartments are mutually exclusive security sanctuaries with discrete jurisdictions.

"Franchise" is a security jurisdiction privileged to a given set of users, i.e.,

$$F = \{u | u \text{ is a user}\} \quad (3)$$

For a given terminal, t , let a given Authority set, A, be denoted by A_t , or in general, let a given security object, α , denote a given property, P, for α as P_α . Hence we can speak of A_u , or C_j , etc., to mean the specific Authority set for a given user, u , or the specific Category set for a given job, j , respectively.

Four important sets (of users) arise with respect to the Franchise property, namely, Franchise for files, terminals, jobs, and users. To distinguish the sense in which a given user is being considered, we subscript u by the security object under consideration. Hence, u_f means the user with jurisdiction to file f ; u_t and u_j are similarly defined. For completeness, we define u_u as

apply u . We can now define Franchise for each security object.

$$F_u = \{u\} \quad (4)$$

$$F_t = \{u_t^0, u_t^1, \dots, u_t^\lambda\} \quad (5)$$

$$F_j = \{u_j^0, u_j^1, \dots, u_j^\mu\} \quad (6)$$

$$F_f = \{u_f^0, u_f^1, \dots, u_f^\nu\} \quad (7)$$

Equation (4) states that the Franchise for a user is restricted to himself; his jurisdiction is unique, and no other user is so endowed. Equation (5) states that the terminal Franchise is possessed by λ different users who have jurisdiction over the terminal t . Likewise, equations (6) and (7) define the job and file Franchise sets.

In security discussions, one hears the familiar phrase, "he needs a higher-level clearance." We can now define "higher level" with our model.

Let α and β be security objects and let ρ be some function such that $\rho(A_\alpha) \in A$.

Then,

$$A_\alpha \geq A_\beta \leftrightarrow \rho(A_\alpha) \geq \rho(A_\beta) \quad (8)$$

$$C_\alpha \geq C_\beta \leftrightarrow C_\alpha \supseteq C_\beta \quad (9)$$

$$F_\alpha \geq F_\beta \leftrightarrow F_\alpha \supseteq F_\beta \quad (10)$$

Equation (8) claims that the Authority of a security object, A_α is at a "higher level" than another security object A_β when the specific authority, a_α is greater than the specific authority, a_β .

It is implicit in equations (1) and (8) that the specific authorities, a_i , must be numerically encoded for the magnitude relationships to hold. Equations (9) and (10) define P_α to be greater than P_β if and only if P_β is a subset of P_α .

Events may alter the membership of property sets. Let P_j^e be the e th P_j in a given context.

Define the Authority history, A_h , at the e th event as

$$A_h(0) = a_j^0 \quad (11)$$

$$A_h(e) = \max(A_h(e-1), \rho(A_j^e)), e > 0 \quad (12)$$

Likewise, define the Category history C_h , at the e th event as

$$C_h(0) = \phi \quad (13)$$

$$C_h(e) = C_h(e-1) \cup C_f^e, e > 0 \quad (14)$$

Equations (11) through (14) recursively define two useful sets that accumulate a history of file references as a function of file reference events, e . A history of the highest Authority, A_h , is defined by equation (12) as either the previous set, $A_h(e-1)$, or the current set, $\rho(A_j^e)$, whichever is larger in the sense of equation (8). Equation (11) gives the initial condition as some low specific file authority, a_j^0 . Equation (14) defines the highest Category history as the union of the previous set, $C_h(e-1)$, and the current set, C_f^e ; while equation (13) states that the union is initially the empty set.

Though F_h could be defined in our model, no need is seen at this time for a Franchise history. More will be said about these history sets later.

Property determination

Table I presents in a 3×4 matrix a summary of the rules for determining the security profile triplets, P_α . We shall examine these rules here. For the user u , A_u and C_u are given constants, and F_u is given by equation (4). For the terminal t , A_t and C_t are given constants, and F_t is given by equation (5). Given A_u and A_t , we determine A_j as:

$$A_j = \min(A_u, A_t) \quad (15)$$

Likewise, given C_u and C_t , we determine C_j as:

$$C_j = C_u \cap C_t \quad (16)$$

Equation (6) gives F_j to complete the job security profile triplet.

An existing file has its security profile predetermined with A_f and C_f as given constants, and F_f as given by equation (7). However, a new file—one just created—derives its security profile from the job's file access history according to the following:

$$A_f = A_h(e) \quad (17)$$

$$C_f = C_h(e) \quad (18)$$

$$F_f = u_j^i \quad (19)$$

From equations (11) through (14) we see how the Authority and Category histories accumulate as a function of event e . These events are the specific times when files are accessed by a job. To maintain security

TABLE I—Security property determination matrix

Object \ Property	Authority A	Category C	Franchise F
User, u	Given Constant	Given Constant	u
Terminal, t	Given Constant	Given Constant	u_t^i
Job, j	$\min(A_u, A_t)$	$C_u \cap C_t$	u_j^i
File, f	<i>Existing file</i> Given Constant	<i>Existing file</i> Given Constant	u_f^i
	<i>New file</i> $\max(A_k(e-1), \rho(A_k^i)), e > 0$	<i>New file</i> $C_k(e-1) \cup C_k^i, e > 0$	u_f^i

integrity, these histories can never exceed (i.e., be greater than) the job security profile. This is specified as,

$$A_k(\infty) \rightarrow A_j \quad (20)$$

$$C_k(\infty) \rightarrow C_j \quad (21)$$

For $e=0$, we see the properties initialized to their simplest form. However, as e gets large, the histories accumulate, but never exceed the upper limit set by the job. $A_k(e)$ and $C_k(e)$ are important new concepts, discussed in further detail later. We speak of them, affectionately, as the security "high-water mark," with analogy to the bath tub ring that marks the highest water level attained.

The Franchise of a new file is always obtained from the Franchise of the job given by equation (6). When $i = \mu = 0$, the job is controlled by the single user u , who becomes the owner and creator of the file with the sole Franchise for the file.

Access control

Our model is now rich enough to express the equations of access control. We wish to control access by a user to the system, to a terminal, and to a file. Access is granted to the system if and only if

$$u \in U \quad (22)$$

where U is the set of all sanctioned users known to the system.

Access is granted to a terminal if and only if

$$u \in F_t \quad (23)$$

If equations (22) and (23) hold, then by definition

$$u = u_t = u_j \quad (24)$$

Access is granted to a file if and only if

$$P_j \geq P_f \quad (25)$$

for properties A and C according to equations (8) and (9), and

$$u_j \in F_f \quad (26)$$

If equations (25) and (26) hold, then access is granted and $A_k(e)$ and $C_k(e)$ are calculated by equations (12) and (14).

Model interpretation

Three different dimensions for restricting access to sensitive information and information processes are possible with the security profile triplet. The generality of this technique has considerable application to public and military systems. For the system of interest, however, the Authority property corresponds to the Top Secret, Secret, etc., levels of government and military security; Category corresponds to the host of special control compartments used to restrict access by project and area; such as those of the Intelligence and Atomic Energy communities; and the Franchise property corresponds to access sanctioned on the basis of

need-to-know. With this interpretation, the popular security terms "classification" and "clearance" can be defined by our model in the same dimensions—as a min/max test on the security profile triplet. Classification is attached to a security object to designate the minimum security profile required for access, whereas clearance grants to a security object the maximum security profile it has permission to exercise. Thus, legal access obtains if the clearance is greater than or equal to the classification, i.e., if equation (25) holds.

Another observation on the model is the "job umbrella" concept implied by equations (22) through (26); i.e., the derived clearance of the job (not the clearance of the user) is used as the security control triplet for file access. The job umbrella spreads a homogeneous clearance to normalize access to a heterogeneous assortment of program and data files. This simplifies the problem of control in a multi-level security system. Also note how the job umbrella's high-water mark (equations (11) through (14)) is used to automatically classify new files (equations (17) and (18)); this subject is discussed further below.

A final observation on the model is its application of need-to-know to terminal access, equation (23). This feature allows terminals to be restricted to special people and/or special groups for greater control of personnel interfaces—i.e., systems programmers, computer operators, etc.

Security control implementation

The selection of a set theoretic model of security control was not fortuitous, but a deliberate choice biased toward computational efficiency and ease of implementation. It permits the clean separation and isolation of security control code from the security control data, which enables ADEPT's security mechanisms to be openly discussed and still remain safe—a point advocated by others.^{14,16} We achieve this safety by "arming" the system with security control data only once at start-up time by the SYSLOG procedure discussed later. Also, the model improves the credibility of the security system, enhancing its understanding and thereby promoting its certification.

Security objects: Identity and structure

Each security object has a unique identification (ID) within the system such that it can be managed individually. The form of the ID depends upon the security-object type; the syntax of each is given below.

User identification

For generality of definition, each user is uniquely identified by his *userid*, which must be less than 13 characters with no embedded blanks.

The *userid* can be any meaningful encoding for the local installation. For example, it can be the individual's Social Security number, his military serial number, his last name (if unique and less than 13 characters), or some local installation man-number convention. The set of all *userid*s constitutes the universal set, *U*.

Terminal identification

All peripheral devices in ADEPT are identified uniquely by their IBM 360 device addresses. Besides interactive terminals, this includes disc drives, tape drives, line printer, card reader-punch, drums, and 1052 keyboard. Therefore, *terminal:id* must be a two-digit hexadecimal number corresponding to the unit address of the device.

Job identification

ADEPT consists of two parts: the Basic Executive (BASEX), which handles the allocation and scheduling of hardware resources, and the Extended Executive (EXEX), which interfaces user programs with BASEX. ADEPT is designed to operate itself and user programs as a set of 4096-byte pages. BASEX is identified as certain pages that are fixed in main core, whereas EXEX and user programs are identified as sets of pages that move dynamically between main and swap memory. A set of user programs are identified as a job, with page sets for each program (the program map) described in the job's environment area, i.e., the job's "state tables." Every job in ADEPT has an environment area that is swapped with the job. It contains dynamic system bookkeeping information pertinent to the job, including the contents of the machine registers (saved when the job is swapped out), internal file and I/O control tables, a map of all the program's pages on drum, *userid*, and the job security control parameters. The environment page(s) are memory-protected against reading and writing by user programs, as they are really swappable extensions of the monitor's tables.

The *job:id* is then a transitory internal parameter which changes with each user entrance and exit from the system. The *job:id* is a relative core memory address used by the executive as a major index into central system tables. It is mapped into an external two-digit number that is typed to the user in response to a successful LOGIN.

File identification

ADEPT's file system is quite rich in the variety of file types, file organization, and equipment permitted. There are two file types: temporary and permanent.

Temporary files are transitory "scratch" disc files, which disappear from the system inventory when their parent job exits from the system. They are always placed on resident system volumes, and are private to the program that created them.

Permanent files constitute the majority of files cataloged by the system. Their permanence derives from the fact that they remain inventoried, cataloged, and available even after the job that created or last referenced them is no longer present, and even if they are not being used. Permanent files may be placed by the user on resident system volumes or on demountable private volumes.

There are six file organizations from which a user may select to structure the records of his file: Physical-sequential, S1; non-formatted, S2; index-sequential, S3; partitioned, S4; multiple volume fixed record, S5; and single volume fixed record, S9. Regardless of the organization of the records, ADEPT manages them as a collection, called a file. Thus, security control is at the file level only, unlike more definitive schemes of sub-element control.^{8,10-12}

All the control information of a file that describes type, organization, physical storage location, date of creation, and security is distinct from the data records of the file, and is the catalog of the file.

All cataloged ADEPT files are uniquely identified by a four-part name; each part has various options and defaults (system assumptions). This name, the *file: id*, has the following form:

$$file: id ::= name, form, user: id, volume: id$$

Name is a user-generated character string of up to eight characters with no embedded blanks. It must be unique on a private volume as well as for Public files (described below).

Form is a descriptor of the internal coding of a file. Up to 256 encodings are possible, although only these seven are currently applicable:

- 1 = binary data
- 2 = relocatable program
- 3 = non-relocatable program
- 4 = card images
- 5 = catalog
- 6 = DLO (Delayed Output)
- 7 = line images

User: id corresponds to the owner of the file, i.e., the creator of the file.

Volume: id is the unique file storage device (tape, disc, disc pack, etc.) on which the file resides. For various reasons, including reliability, ADEPT file inventories are distributed across the available storage media, rather than centralized on one particular volume. Thus, all files on a given disc volume are inventoried on that volume.

Security properties: Encoding and structure

Implementation of the security properties in ADEPT is not uniform across the security objects as suggested by our model, particularly the Franchise property. Lack of uniformity, brought about by real-world considerations, is not a liability of the system but a reflection of the simplicity of the model. Extensions to the model are developed here in accordance with that actually implemented in ADEPT.

Authority

Authority is fixed at four levels ($\omega = 3$ for equation (1)) in ADEPT, specifically, UNCLASSIFIED, CONFIDENTIAL, SECRET, and TOP SECRET in accordance with Department of Defense security regulations. The Authority set is encoded as a logical 4-bit item, where positional order is important. Magnitude tests are used extensively, such that the high-order bits imply high Authority in the sense of equation (8).

Category

Category is limited to a maximum of 16 compartments ($\psi \leq 15$ for equation (2)), encoded as a logical 16-bit item. Boolean tests are used exclusively on this datum. The definition of (and bit position correspondence to) specific compartments is an installation option at ADEPT start-up time (see SYSLOG). Typical examples of compartments are EYES ONLY, CRYPTO, RESTRICTED, SENSITIVE, etc.

Franchise

Property Franchise corresponds to the military concept of need-to-know. Essentially, this corresponds to a set of *user: ids*; however, the ADEPT implementation of Franchise is different for each security object:

1. User: All users wishing ADEPT service must be known to the system. This knowledge is imparted by SYSLOG at start-up time and limited to approximately 500 *user: ids* ($\max(U) \leq 500$).

2. Terminal: Equation (5) specifies the Franchise of a given terminal, F_t , as a set of *user:ids*. In ADEPT, F_t does not exist. One may define all the users for a given terminal, i.e., F_t ; or alternatively, all the terminals for a given user. Because SYSLOG orders its tables by *user:id*, the latter definition was found more convenient to implement.
3. Job: The Franchise of a job is the *user:id* of the creator of the job at the time of LOGIN to the system. Currently, only one user has access to (and control of) a job ($\mu = 0$ for equation (6)).
4. File: Implementation of Franchise for a file (F_f), is more extensive than equation (7). In ADEPT, we wish to control not only who accesses a file, but also the quality of access granted. We have defined a set of four exclusive qualities of access, such that a given quality, q , is defined if

$$q \in \{\text{READ, WRITE, READ-AND-WRITE, READ-AND-WRITE-WITH-LOCKOUT-OVERRIDE}\} \quad (27)$$

ADEPT permits simultaneous access to a file by many jobs if the quality of access is for READ only. However, only one job may access a file with WRITE, or READ-AND-WRITE quality. ADEPT automatically locks out access to a file being written to avoid simultaneous reading and writing conflicts. A special access quality, however, does permit lockout override. Equation (7) can now be extended as a set of pairs,

$$F_f = \{(u_j^0, q^0), (u_j^1, q^1), \dots, (u_j^\gamma, q^\gamma)\}; \quad (28)$$

where q^i are not necessarily distinct and are given by equation (27).

The implementation of equation (28) is dependent upon γ , the number of franchised users. When $\gamma = 0$, we have the ADEPT Private file, exclusive to the owner, u_j^0 ; for $\gamma = \max(U)$, we have the Public file; values of γ between these extremes yield the Semi-Private file. γ is implicitly encoded as the ADEPT "privacy" item in the file's catalog control data, and takes the place of F_f for all cases except a Semi-Private file. For that case exclusively, equation (28) holds and an actual F_f list of *user:id, quality* pairs exists as a need-to-know list. The owner of a file specifies and controls the file's privacy, including the composition of the need-to-know list.

Security control initialization: SYSLOG

SYSLOG is a component of the ADEPT initialization package responsible for arming the security controls. It operates as one of a number of system start-up options prior to the time when terminals are enabled. SYSLOG sets up the security profile data for *user:id* and *terminal:id*, i.e., the "given constants" of Table I.

SYSLOG creates or updates a highly sensitive system disc file, where each record corresponds to an authorized user. These records are constructed from a deck of cards consisting of separate data sets for *compartment* definitions, *terminal:id* classification, and *user:id* clearance. The dictionary of *compartment* definitions contains the less-than-9-character mnemonic for each member of the Category set. Data sets are formed from the card types shown in Table II. Use of *passwords* is described later in the LOGIN procedure.

An IDT card must exist for each authorized user; the PWD, DEV, SEC, and CAT card types are optional. Other card types are possible, but not germane to security control, e.g., ACT for accounting purposes. More than one PWD, DEV, and CAT card is acceptable up to the current maximum data limits (i.e., 64 *passwords*, 48 *terminal:ids*, and 16 *compartments*).

A variety of legality checks for proper data syntax, quantity, and order are provided. SYSLOG assumes the following default conditions when the corresponding card type is omitted from each data set:

PWD	No <i>password</i> required
DEV	All <i>terminal:ids</i> authorized
SEC	A = UNCLASSIFIED
CAT	C = null (all zero mask)

This gives the lowest user clearance as the default, while permitting convenient user access. Various options exist in SYSLOG to permit maintenance of the internal SYSLOG tables, including the replacement or deletion of existing data sets in total or in part.

The sensitivity of the information in the security control deck is obvious. Procedures have been developed at each installation that give the function of deck creation, control, and loading to specially cleared security personnel. The internal SYSLOG file itself is protected in a special manner described later.

Access control

A fundamental security concern in multi-access systems is that many users with different clearances will be simultaneously using the system, thereby raising the

TABLE II—SYSLOG control cards

Card Type	Purpose
DICT <i>compartment₁ ... compartment₁₆</i>	Identifies start of data set of <i>compartment</i> definitions. Defines up to 16 <i>compartments</i> .
TERMINAL UNIT <i>terminal:id</i> IDT <i>user:id</i>	Identifies start of data sets of terminal definitions. Identifies start of a terminal data set. Identifies start of a user data set.
PWD <i>password ... password</i> DEV <i>terminal:id₁ ... terminal:id₄₈</i>	Defines legal <i>passwords</i> for <i>user:id</i> up to 64. Defines legal terminals for <i>user:id</i> up to 48.
SEC Authority CAT <i>compartment₁ ... compartment₁₆</i>	Defines <i>user:id</i> Authority. Defines <i>user:id</i> Category set.

possibility of security compromise. Since programs are the "active agents" of the user, the system must maintain the integrity of each and of itself from accidental and/or deliberate intrusion. A multifile system must permit concurrent access by one or more jobs to one or more on-line, independently classified files.

ADEPT is all these things—multiuser, multiprogram, and multifile system. Thus, this section deals with access control over users, programs, and files.

User access control: LOGIN

To gain admittance to the system, a user must first satisfy the ADEPT LOGIN decision procedure. This procedure attempts to authenticate the user in a fashion analogous to challenge-response practices.

The syntax of the ADEPT LOGIN command, typed by a user on his terminal, is as follows:

```
/LOGIN user:id password accounting
```

Figure 1 pictorially displays the LOGIN decision procedure based upon the user-specified input parameters. *User:id* is the index into the SYSLOG file used to retrieve the user security profile. If no such record exists (i.e., equation (22) fails), the LOGIN is unsuccessful and system access is denied. If the security profile is found, LOGIN next retrieves the *terminal:id* for the keyboard in use from internal system tables, and searches for a match in the *terminal:id* list for which the *user:id* was franchised by SYSLOG. An unsuccessful search is an unsuccessful LOGIN.

If the terminal is franchised, then the current *password* is retrieved from the SYSLOG file for this *user:id* and matched against the *password* entered as a keyboard parameter to LOGIN. An unsuccessful match is again

an unsuccessful LOGIN. Furthermore, the terminal is ignored (will not honor input) for approximately 30 seconds to frustrate high-speed, computer-assisted, penetration attempts. If, however, the match is successful (equation (22) holds), the current *password* in the SYSLOG file for this *user:id* is discarded and LOGIN proceeds to create the job clearance.

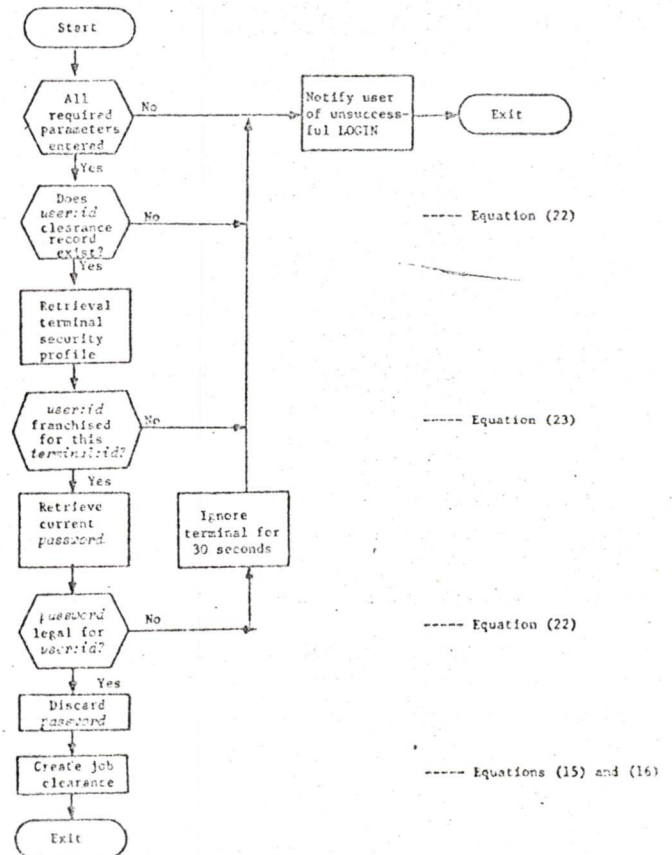


Figure 1—LOGIN decision procedure

Passwords in ADEPT obey the same syntax conventions as *user: id*. (See the earlier description of User Identification.) Although easily increased, currently SYSLOG permits up to 64 *passwords*. Each successful LOGIN throws away the user *password*; 64 successful LOGINS are possible before a new set of *passwords* need be established. If other than random, once-only *passwords* are desired, the 64 *passwords* may be encoded in some algorithmic manner, or replicated some number of times. Once-only *passwords* is an easily implemented technique for user authentication, which has been advocated by others.^{2,7} It is a highly effective and secure technique because of the high permutability of 12-character-*passwords* and their time and order interdependence, known only to the user.

Once the authentication process is completely satisfied, LOGIN creates the job security profile according to equations (15) and (16) of our model. That is, the lower Authority of the user and the terminal becomes A_j , and the intersection (logical AND) of the user and terminal Category sets becomes the Category of the job, C_j . For example, a user with TOP SECRET Authority and a Category set (1001 1001 0000 1101) operating from a SECRET level terminal with a Category set (0000 0000 0000 0010) controls a job cleared to SECRET with an empty Category set.

Program access control: LOAD

As noted earlier, the ADEPT Executive consists of two parts: BASEX, the resident part, and EXEX, the swapped part. EXEX is a body of reentrant code shared by all users; however, it is treated as a distinct program in each user's job. Up to four programs can exist concurrently in the job. Each operates with the job clearance—the job clearance umbrella.

LOAD is the ADEPT component used to load the programs chosen by the user; it is part of EXEX and hence operates as part of the user's job with the job's clearance. Programs are cataloged files and as such may be classified with a given security profile. As is described in "File Access Control" below, LOAD can only load those programs for which the job clearance is sufficient. Once loaded, however, the new program operates with the job clearance.

In this manner, we see the power of the job umbrella in providing smooth, flexible user operation concurrent with necessary security control. Program files may be classified with a variety of security profiles and then operate with yet another, i.e., the job clearance. By this technique security is assured and programs of different classifications may be operated by a user as one job. It

permits, for example, an unclassified program file (e.g., a file editor) to be loaded into a highly classified job to process sensitive classified data files.

File access control: OPEN

Before input/output can be performed on a file, a program must first acquire the file by an OPEN call to the Cataloger. Each program must OPEN a file for itself before it can manipulate the file, even if the file is already OPENed for another program. A successful OPEN requires proper specification of the file's descriptors—some of which are in the OPEN call, others of which are picked up directly by the Cataloger from the job environment area (e.g., job clearance, *user: id*)—and satisfactory job clearance and *user: id* need-to-know qualifications according to equations (25) and (26) of our model. Equation (25) is implemented as (8) as a straightforward magnitude comparison between A_j and A_f . Equation (25) is implemented as (9) as an equality test between C_j and $(C_j \wedge C_f)$. We use $(C_j \wedge C_f)$ to ensure that C_f is a subset of the job categories; i.e., the job umbrella. Lastly, equation (26) is a NOP if the file is Public; a simple equality test between u_j and u_f if the file is Private; and a table search of F_f for u_j if the file is Semi-Private. These tests do increase processing time for file access; however, the tests are performed only once at OPEN time, where the cost is insignificant relative to the I/O processing subsequently performed on the file.

The quality of access granted by a successful OPEN, and subsequently enforced for all I/O transfers, is that requested, even if the user has a greater Franchise. For example, during program debugging, the owner of a file may OPEN it for READ access only; even though READ-AND-WRITE access quality is permitted. He thereby protects his file from possible uncontrolled modification by an erroneous WRITE call.

Considerable controversy surrounds the issue of automatic classification of new files formed by subset or merger of existing files. The heart of the issue is the poor accuracy of many such classification techniques¹⁷ and the fear of too many over-classified files (a fear of operations personnel) or of too many under-classified files (a fear of the security control officers). ADEPT finesses the problem with a clever heuristic—most new files are created from existing files, hence classify the new file as a private file with the composite Authority and Category of all files referenced. This is achieved in ADEPT by use of the "high-water mark."

Starting with the boundary conditions of equations (11) and (13), the Cataloger applies equations (12) and

(14) for each successful file OPEN, and hence maintains the composite classification history of all files referenced by the job. For each new and temporary file OPEN, the Cataloger applies equations (17), (18), and (19); they are reapplied for each CLOSE of a new file, to update the classification (due to changes in the high-water mark since the OPEN) when the file becomes an existing cataloged file in the inventory. The scheme rarely underclassifies, and tends to overclassify when the new file is created late in the job cycle, as shown by boundary equations (20) and (21).

Trans-formal security features

ADEPT contains a host of features that transcend the formalism presented earlier. They are described here because they are integral to the total security control system and form a body of experience from which new formalisms can draw.

Computer hardware

ADEPT operates on an IBM System 360/50 and is, therefore, limited to the hardware available. Studies by Bingham⁹ suggest a variety of hardware features for security control, many of which are possessed by System 360.

IBM System 360 can operate in one of two states: the Supervisor state, or the Problem state. ADEPT executive programs operate in the Supervisor state; user programs operate in the Problem state.

A number of machine instructions are "privileged" to the Supervisor state only. An attempt to execute them in the Problem state is trapped by the hardware and control is returned to the executive program for remedial action. ADEPT disposes of these alarms by suspending the guilty job. (A suspended job may be resumed by the user.) Clearly, instructions that change the machine state are privileged to the executive only.

Another class of privileged instructions consists of those dealing with input/output. Problem state programs cannot directly access information files on secondary memory storage devices such as disc, tape, or drum. They must access these files indirectly by requests to the executive system. The requests are subjected to interpretive screening by the executive software.

Main memory is selectively protected against unauthorized change (write protected). We have also had the 360/50 modified to include fetch protection, which guards against unauthorized reading of—or executing from—protected memory. The memory protect instruc-

tions are also privileged only in the Supervisor state.

ADEPT software protects memory on a 4096-byte "page" basis (the hardware permits 2048-byte pages), allowing a non-contiguous mosaic of protected pages in memory for a given program. To satisfy multiprogramming, many different protection groups are needed. Through the use of programmable 4-bit hardware masks, up to 15 different protection groups can be accommodated in core concurrently. ADEPT executive programs operate with the all-zero "master key" mask, permitting universal access by all Basic and Extended Executive components.

There are five classes of interrupts processed by System/360 hardware: input/output, program, supervisor call, external, and machine check. Any interrupts that occur in the Problem state cause an automatic hardware switch to the Supervisor state, with CPU control flowing to the appropriate ADEPT executive interrupt controller. All security-vulnerable functions including hardware errors, external timer and keyboard actions, user program service requests, illegal instructions, memory protect violations, and input/output, are called to the attention of ADEPT by the System/360 interrupt system. The burden for security integrity is then one for ADEPT software.

Monitor software

Inducing the system to violate its own protection mechanisms is one of the most likely ways of breaking a multi-access system. Those system components that perform tasks in response to user or program requests are most susceptible to such seduction.

On-Line debugging

The debugging program provides an on-line capability for the professional programmer to dynamically look at and change selected portions of his program's memory. DEBUG can be directed to access sensitive core memory that would not be trapped by memory protection, since, as an EXEX component operating in the Supervisor state, DEBUG operates with the memory protection master key. To close this "trap door," DEBUG always performs interpretive checks on the legality of the debugging request. These checks are based upon address-out-of-bounds criteria, i.e., the requested debugging address must lie within the user's program area. If not, the request will be denied and the user warned, but he will not be terminated as has been suggested.⁷

Input/output

Input/output in System/360 is handled by a number of special-purpose processors, called Selector Channels. To initiate any I/O, it is necessary for a channel program to be executed by the Selector Channel.

SPAM, the BASEX component that permits symbolic input/output calls from user programs, is really a special-purpose compiler that produces I/O channel programs from the SPAM calls. These channel programs are subsequently delivered and executed by the ADEPT Input/Output Supervisor, IOS.

SPAM permits a variety of calls to read, write, alter, search for, and position to records within cataloged files. To achieve these ends, SPAM depends upon a variety of control tables dynamically created by the Cataloger in the job environment.

The initiating and subsequent monitoring of channel program execution is the responsibility of the BASEX Input/Output Supervisor, IOS. IOS is called to execute a channel program (EXCP). System components, such as SPAM, branch to IOS at a known entry point that is fetch-protected against entry in the Problem state. IOS is off-limits to user programs attempting to access cataloged storage. For protection against unauthorized EXCP requests, IOS always performs legality checks before executing a channel program. These checks begin by examination of the device addressed by the channel program. If it is the device address for cataloged storage, further checks are made to determine the machine state of the calling program. That state must be Supervisor state for the call to be honored. A call in the Problem state would indicate an illegal EXCP call from a user program.

IOS makes other checks to guarantee the validity of an I/C request. It checks to see that the specified buffer areas for the transfer do not overlay the channel program itself, nor lie within the user's program memory area, i.e., do not modify or access system or protected memory.

Covert I/O violations are also forestalled since I/O components take direction from information stored in the job environment—an area read- and write-protected from Problem state programs.

Classified residue

Classified residue is classified information (either code or data) left behind in memory (i.e., core, drum, or disc) after the program that referenced it has been dismissed, swapped out, or quit from the system. The standard solution to the problem is to dynamically purge the contaminated memory (e.g., overwrite with

random numbers, or zeros). In a system supporting over $\frac{1}{4}$ billion bytes of memory, that solution is unreasonable and in conflict with high performance goals. ADEPT's solution to the dilemma of denying access to classified residue while maintaining high performance depends upon techniques of controlled memory allocation.

1. *Core Residue*

As noted earlier, all core storage is allocated as 4096-byte pages. These pages are always cleared to zero when allocated, thereby overwriting any potential residue.

Via the program's page map, the ADEPT executive system labels all code and data pages (they need not be contiguous) belonging to a given program with a single hardware memory protection key, thereby prohibiting unauthorized reading or writing by other, potentially co-resident user programs that may be in execution. Furthermore, BASEX keeps a running account of the status and disposition of all pages of core.

The Loader and Swapper components of ADEPT always work with full 4096-byte pages. Unfilled portions of pages at load time are kept cleared to zero as when they were allocated, and the full 4096 bytes are swapped into core, if not already resident, each scheduled time slice. Further, newly allocated pages are marked as "changed" pages, thus guaranteeing subsequent swap out to drum.

With these procedures, ADEPT denies access by a user or program to those pages of core not identified as part of his program, and clears core residue by over-writing accessible core at load and swap times.

2. *Drum Residue*

ADEPT always clears a drum page to zero before it is allocated. The page may subsequently be cleared again to user-specified data. ADEPT also maintains a drum map that notes the disposition of all drum pages (800 pages for the IBM 2303 drum). Drum input/output, like all ADEPT I/O, is controlled by executive privileged instructions.

3. *Disc Residue*

Disc files in ADEPT are maintained as "dirty" memory. That is, the large capacity of the file system makes it infeasible to consider automatic over-writing techniques for residue control; therefore, deleted disc tracks are returned to the available storage pool contaminated and unclean. It then becomes the burden of the

ADEPT file system to control any unauthorized file access, whether to cataloged files or uncataloged disc memory.

Team work between the Cataloger, SPAM and IOS components of ADEPT achieves this control via legality checking of all OPEN and I/O requests.

For example, all disc packs are labeled internally and externally with their *volume:id*, and this label is checked at the time of mounting by the Cataloger OPEN procedure to assure proper volume mounting. Tapes may also be labeled and checked as a user option.

Of particular note, SPAM always assumes that an end-of-file (EOF) immediately follows the last record written in a new file, and it prohibits reading beyond that EOF. Contaminated tracks allocated to new files cannot be read until they are first written. The act of writing advances the EOF and the user simultaneously over-writes the classified residue with his own data. The user cannot skip over the EOF, and the EOF location is itself protected in the job environment area.

4. Tape Residue

No special features for tape residue control are implemented in ADEPT. Tape residue control is easily satisfied by manual, off-line tape degaussing prior to ADEPT use.

System files

Equation (28) led us to examine Private, Semi-Private, and Public files. ADEPT possesses two additional file privacies that transcend our model; both are system files. Privacy-4 system files are the need-to-know lists created by the Cataloger itself for Semi-Private files. Privacy-5 system files are private system memory for the SYSLOG files and the catalogs themselves.

Access to these files is restricted to the system only. Special access checks are made that differ from those of equations (25) and (26). First, a special *user:id* is required that is not a member of *U* (i.e., not in the SYSLOG file). Second, the program making the OPEN call must be in Supervisor state. Third, the program making the OPEN call must be a member of a short list of EXEX programs. The list is built into the Cataloger at the time of compilation. In this manner, access to system files is severely restricted, even to system programs.

Security service commands

ADEPT provides a variety of service commands that involve security control. The commands are listed in Table III. Note that commands VARYON, VARYOFF, REPLACE, LISTU, AUDIT, AUDOFF, and WRAPUP are restricted to a particular terminal—the Security Officer's Station.

TABLE III—Security service commands

Command	Purpose
AUDIT*	Turns on security audit recording.
AUDOFF*	Turns off security audit recording.
CHANGE	Enables the owner of a file to change any of the access control information of the file.
CREATE	Enables a user to create a Semi-Private file and its need-to-know list.
LISTU*	Lists by <i>terminal:id</i> all the current logged in <i>user:ids</i> .
RECLASS	Enables a user to raise or lower his job clearance between the bounds of the original LOGIN and current high-water mark clearance.
RELOG	Like LOGIN, but reconnects a user to an already existing job, as when a remote terminal drops off the communications line.
REPLACE*	Enables a user to move his job to another terminal or to reclassify a given device.
SECURITY	Print on the user's terminal approximately every 100 lines (or only by request) the job high-water mark (or clearance by request) as a reminder to the user and as a classification stamp of the level of current security activity.
VARYON/VARYOFF*	Permits terminals to be varied on- and off-line for flexibility in system maintenance and configuration control.
WRAPUP*	Shuts down system after a specified elapsed time.

* Restricted to Security Officer's Station only.

Audit

The AUDIT function records certain transactions relating to files, terminals, and users, and is the electronic equivalent of manual security accountability logs. Its purpose is to provide a record of user access in order to determine whether security violations have occurred and the extent to which secure data has been compromised. The AUDIT function may be initiated only at start-up time, but may be terminated at any time. All data recorded on disc or tape in real time so the data is safe if the system malfunctions. An auxiliary utility program, AUDLIST, may be used to list the AUDIT file. The information recorded is shown in Table IV.

Implementation of AUDIT is quite straightforward, a product of general ADEPT recording and instrumentation.^{18,19} AUDIT is an EXEX component that is called by, and at the completion of, each function to be recorded. The information to be recorded is passed to AUDIT in the general registers. Additional I/O overhead is the primary cost incurred in the operation of AUDIT, for swapping and file maintenance. This cost is nominal, however, amounting to less than one percent of the CPU time.

SUMMARY

In summary we may ask: How well have we met our goals? First, we believe we have developed and success-

TABLE IV—Security events and information audited by ADEPT-50

EVENT	TIME	STATUS	JOB SECURITY PROFILE	USER SECURITY PROFILE	ACCOUNT NUMBER	USER:ID	TERMINAL:ID	CPU TIME	NEW TERMINAL	TERMINAL SECURITY PROFILE	FILE NAME	FILE OWNER ID	FILE FORM PROFILE	FILE SECURITY PROFILE	FILE VOLUME NUMBER	PROSE CATEGORY NAMES
LOGIN		X	X	X	X	X	X	X	X							
LOGOUT		X	X							X						
OPEN FILE		X	X										X	X	X	X
REOPEN ¹ FILE		X	X										X	X	X	X
CHANGE FILE		X	X										X	X	X	X
CLOSE FILE		X	X										X	X		X
DELETE FILE		X	X										X	X		X
RECLASS		X	X	X												
REPLACE		X	X						X		X	X				
DEVICE LIST ²		X										X				
CATEGORY DICTIONARY ³		X														X
RESTART ⁴		X														
WRAPUP ⁵		X														

¹ This is the "OPEN existing file" command.
² A list of all the terminal devices and their assigned security and categories is recorded at each system load.
³ A list of the prose category names is recorded at each system load.
⁴ Whenever the system is restarted on the same day (and AUDIT had been turned on earlier that day) the time of the restart is recorded.
⁵ The time that the AUDOFF action was taken, or the time that the WRAPUP function called AUDIT, to terminate the AUDIT function.

fully demonstrated a security control mechanism that more than adequately supports heterogeneous levels and types of classification. Of note in this regard is the LOGIN decision procedure, access control tests, job umbrella, high-water mark, and audit trails recording. The approach can be improved in the direction of more compartments (on the order of 1000 or more), extension of the model to include system files, and the implementation of a single Franchise test for all security objects. The implementation needs redundant encoding and error detection of security profile data to increase confidence in the system—though we have not ourselves experienced difficulty here. The increase in memory requirements to achieve these improvements may force numerical encoding of security data, particularly Category, as suggested by Peters.⁷

Second, SYSLOG has been highly successful in demonstrating the concept of "security arming" of the system at start-up time. Our greatest difficulty in this area has been with the human element—the computer operators—in preparing and handling the control deck. In opposition to Peters,⁷ we believe the operator should not be "designed out of the operation as much as possible," but rather his capabilities should be upgraded to meet the greater levels of sophistication and responsibility required to operate a time-sharing system.²⁰ He should be considered part of line management. ADEPT is oriented in this direction and work now in progress is aimed at building a real-time security surveillance and operations station (SOS).

Third, we missed the target in our attempt to isolate and limit the amount of critical coding. Though much of the control mechanism is restricted to a few components—LOGIN, SYSLOG, CATALOGER, AUDIT—enough is sprinkled around in other areas to make it impossible to restrict the omnipotent capabilities of the monitor, e.g., to run EXEX in Problem state. Some additional design forethought could have avoided some of this dispersal, particularly the wide distribution in memory of system data and programs that set and use these data. The effect of this shortcoming is the need for considerably greater checkout time, and the lowered confidence in the system's integrity.

Lastly, on the brighter side, we were surprisingly frugal in the cost of implementing this security control mechanism. It took approximately five percent of our effort to design, code, and checkout the ADEPT security control features. The code represents about ten percent of the 50,000 instructions in the system. Though the code is widely distributed, SYSLOG, security commands, LOGIN, AUDIT, and the CATALOGER account for about 80 percent of it. The overhead cost of

operating these controls is difficult to measure, but it is quite low, in the order of one or two percent of total CPU time for normal operation, excluding SYSLOG. (SYSLOG, of course, runs at card reader speed.) The most significant area of overhead is in the checking of I/O channel programs, where some 5 to 10 msec are expended per call (on the average). Since this time is overlapped with other I/O, only CPU bound programs suffer degradation. AUDIT recording also contributes to service call overhead. In actuality, the net operating cost of our security controls may be zero or possibly negative, since AUDIT recordings showed us numerous trivial ways to measurably lower system overhead.

ACKNOWLEDGMENTS

I would like to acknowledge the considerable encouragement I received in the formative stages of the ADEPT security control design from Mr. Richard Cleaveland, of the Defense Communications Agency (DCA). I would like to thank Mrs. Martha Bleier, Mr. Peter Baker, and Mr. Arnold Karush for their patient care in designing and implementing much of the work I've described. Also, I wish to thank Mr. Marvin Schaefer for assisting me in set theory notation. Finally, I would like to applaud the ADEPT system project personnel for designing and building a time-sharing system so amenable to the ideas discussed herein.

REFERENCES

- 1 A HARRISON
The problem of privacy in the computer age: An annotated bibliography
RAND Corp Dec 1967 RM-5495-PR/RC
- 2 L J HOFFMAN
Computers and privacy: A survey
Stanford Linear Accelerator Center Stanford Univ Aug 1968 SLAC-PUB-479
- 3 H E PETERSEN R TURN
System implications of information privacy
Proc SJCC Vol 30 1967 291-300
- 4 W H WARE
Security and privacy in computer systems
Proc SJCC Vol 30 1967 279-282
- 5 W H WARE
Security and privacy: Similarities and differences
Proc SJCC Vol 30 1967 287-290
- 6 R LINDE C WEISSMAN C FOX
The ADEPT-50 time-sharing system
Proc FJCC Vol 35 1969 Also issued as SDC Doc SP-3344
- 7 B PETERS
Security considerations in a multi-programmed computer system
Proc SJCC Vol 30 1967 283-286
- 8 RYE CAPRI COINS OCTOPUS SADIE Systems

- NOC Workshop National Security Agency Oct 1968
- 9 H W BINGHAM
Security techniques for EDP of multi-level classified information
Rome Air Development Center Dec 1965 RADC-TR-65-415
- 10 R M GRAHAM
Protection in an information processing utility
ACM Symposium on Operating Systems Principles Oct 1967 Gatlinburg Tenn
- 11 L J HOFFMAN
Formularics—Program controlled privacy in large data bases
Stanford Univ Working Paper Feb 1969
- 12 D K HSIAO
A file system for a problem solving facility
Dissertation in Electrical Engineering Univ of Pa 1968
- 13 J I SCHWARTZ C WEISSMAN
The SDC time-sharing system revisited
Proc ACM Conf 1967 263-271
- 14 P BARAN
On distributed communications: IX, security, secrecy, and tamper-free considerations
- RAND Corp Aug 1964 RM-3765-PR
- 15 C WEISSMAN
Programming protection: What do you want to pay?
SDC Mag Vol 10 No 8 Aug 1967
- 16 J P TITUS
Washington commentary—Security and privacy
CACM Vol 10 No 6 June 1967 379-380
- 17 I ENGER et al
Automatic security classification study
Rome Air Development Center Oct 1967 RADC-TR-67-472
- 18 A KARUSH
The computer system recording utility: Application and theory
System Development Corp March 1969 SP-3303
- 19 A KARUSH
Benchmark analysis of time-sharing systems: Methodology and results
System Development Corp April 1969 SP-3343
- 20 R R JJNDE P E CHANEY
Operational management of time-sharing systems
Proc 21st Nat ACM Conf 1966 149-159

Hardware aspects of secure computing

by LEE M. MOLHO

System Development Corporation
Santa Monica, California

INTRODUCTION

It makes no sense to discuss software for privacy-preserving or secure time-shared computing without considering the hardware on which it is to run. Software access controls rely upon certain pieces of hardware. If these can go dead or be deliberately disabled without warning, then all that remains is false security.

This paper is about hardware aspects of controlled-access time-shared computing.* A detailed study was recently made of two pieces of hardware that are required for secure time-sharing on an IBM System 360 Model 50 computer: the storage protection system and the Problem/Supervisor state control system.¹ It uncovered over a hundred cases where a single hardware failure will compromise security without giving an alarm. Hazards of this kind, which are present in any computer hardware which supports software access controls, have been essentially eliminated in the SDC ADEPT-50 Time-Sharing System through techniques described herein.²

Analysis based on that work has clarified what avenues are available for subversion via hardware; they are outlined in this paper. A number of ways to fill these security gaps are then developed, including methods applicable to a variety of computers. Administrative policy considerations, problems in security certification of hardware, and hardware design considerations for secure time-shared computing also receive comment.

FAILURE, SUBVERSION, AND SECURITY

Two types of security problem can be found in computer hardware. One is the problem of hardware failure.

*The relationship between "security" and "privacy" has been discussed elsewhere.^{3,4} In this paper "security" is used to cover controlled-access computing in general.

This includes not only computer logic that fails by itself, but also miswiring and faulty hardware caused by improper maintenance ("Customer Engineer") activity, including CE errors in making field-installable engineering changes.

The other security problem is the cloak-and-dagger question of the susceptibility of hardware to subversion by unauthorized persons. Can trivial hardware changes jeopardize a secure computing facility even if the software remains completely pure? This problem and the hardware failure problem, which will be considered in depth, are related.

Weak points for logic failure

Previous work involved an investigation of portions of the 360/50 hardware.¹ Its primary objective was to pinpoint single-failure problem locations. The question was asked, "If this element fails, will hardware required for secure computing go dead without giving an alarm?" A total of 99 single-failure hazards were found in the 360/50 storage protection hardware; they produce a variety of system effects. Three such logic elements were found in the simpler Problem/Supervisor state (PSW bit 15) logic. A failure in this logic would cause the 360/50 to always operate in the Supervisor state.

An assumption was made in finding single-failure logic problems which at first may seem more restrictive than it really is: A failure is defined as having occurred if the output of a logic element remains in an invalid state based on the states of its inputs. Other failure modes certainly exist for logic elements, but they reduce to this case as follows: (1) an intermittent logic element meets this criterion, but only part of the time; (2) a shorted or open input will cause an invalid output state at least part of the time; (3) a logic element which exhibits excessive signal delay will appear to have an invalid output state for some time after any input transition; (4) an output wire which has been con-

nected to an improper location will have an invalid output state based on its inputs at least part of the time; such a connection may also have permanently damaged the element, making its output independent of its input. It should be noted that failure possibilities were counted; for those relatively few cases where a security problem is caused whether the element gets stuck in "high" or in "low" state, two possibilities were counted.

A situation was frequently encountered which is considered in a general way in the following section, but which is touched upon here. Many more logic elements besides those tallied would cause the storage protection hardware to go dead if they failed, but fortunately (from a security viewpoint) their failure would cause some other essential part of the 360/50 to fail, leading to an overall system crash. "Failure detection by faulty system operation" keeps many logic elements from becoming security problems.

Circumventing logic failure

Providing redundant logic is a reasonable first suggestion as a means of eliminating single failures as security problems. However, redundancy has some limits which are not apparent until a close look is taken at the areas of security concern within the Central Processing Unit (CPU). Security problems are really in control logic, such as the logic activated by a storage protect violation signal, rather than in multi-bit data paths, where redundancy in the form of error-detecting and error-correcting codes is often useful. Indeed, the 360/50 CPU already uses an error-detecting code extensively, since parity checks are made on many multi-bit paths within it.

Effective use of redundant logic presents another problem. One must fully understand the system as it stands to know what needs to be added. Putting it another way, full hardware certification must take place before redundancy can be added (or appreciated, if the manufacturer claims it is there to begin with).

Lastly, some areas of hardware do not lend themselves too easily to redundancy: There can be only one address at a time to the Read-Only-Storage (ROS) unit whose microprograms control the 360/50 CPU.^{5,6} One could, of course, use such a scheme as triple-modular redundancy on all control paths, providing three copies of ROS in the bargain. The result of such an approach would not be much like a 360/50.

Redundancy has a specialized, supplementary application in conjunction with hardware certification. After the process of certification reveals which logic elements can be checked by software at low overhead, redundant

logic may be added to take care of the remainder. A good example is found in the storage protection logic. Eleven failure possibilities exist where protection interrupts would cause an incorrect microprogram branch upon failure. These failure possibilities arise in part from the logic elements driven by one control signal line. This signal could be provided redundantly to make the hardware secure.

Software tests provide another way to eliminate hardware failure as a security problem. Code can be written which should cause a protection or privileged-operation interrupt; to pass the test the interrupt must react appropriately. Such software must interface the operating system software for scheduling and storage-protect lock alteration, but must execute in Problem state to perform its tests. There is clearly a tradeoff between system overhead and rate of testing. As previously mentioned, hardware certification must be performed to ascertain what hardware can be checked by software tests, and how to check it.

Software testing of critical hardware is a simple and reasonable approach, given hardware certification; it is closely related to a larger problem, that of testing for software holes with software. Software testing of hardware, added to the SDC ADEPT-50 Time-Sharing System, has eliminated over 85 percent of present single-failure hazards in the 360/50 CPU.

Microprogramming could also be put to work to combat failure problems. A microprogrammed routine could be included in ROS which would automatically test critical hardware, taking immediate action if the test were not passed. Such a microprogram could either be in the form of an executable instruction (e.g., TEST PROTECTION), or could be automatic, as part of the timer-update sequence, for example.

A microprogrammed test would have much lower overhead than an equivalent software test performed at the same rate; if automatic, it would test even in the middle of user-program execution. A preliminary design of a storage-protection test that would be exercised every timer update time (60 times per second) indicated an overhead of only 0.015 percent (150 test cycles for every million ROS cycles). Of even greater significance is that microprogrammed testing is specifiable. A hardware vendor can be given the burden of proof of showing that the tests are complete; the vendor would have to take the testing requirement into account in design. The process of hardware certification could be reduced to a design review of vendor tests if this approach were taken.

Retrofitting microprogrammed testing in a 360/50 would not involve extensive hardware changes, but some changes would have to be made. Testing microprograms would have to be written by the manu-

facturer; new ROS storage elements would have to be fabricated. A small amount of logic and a large amount of documentation would also have to be changed.

Logic failure can be totally eliminated as a security problem in computer hardware by these methods. A finite effort and minor overhead are required; what logic is secured depends upon the approach taken. If microprogram or software functional testing is used, miswiring and dead hardware caused by CE errors will also be discovered.

Subversion techniques

It is worthwhile to take the position of a would-be system subverter, and proceed to look at the easiest and best ways of using the 360/50 to steal files from unsuspecting users. What hardware changes would have to be made to gain access to protected core memory or to enter the Supervisor state?

Fixed changes to eliminate hardware features are obvious enough; just remove the wire that carries the signal to set PSW bit 15, for example. But such changes are physically identical to hardware failures, since something is permanently wrong. As any functional testing for dead hardware will discover a fixed change, a potential subverter must be more clever.

In ADEPT-50, a user is swapped in periodically for a brief length of time (a "quantum"). During his quantum, a user can have access to the 360/50 at the machine-language level; no interpretive program comes between the user and his program unless, of course, he requests it. Thus, a clever subverter might seek to add some hardware logic to the CPU which would look for, say, a particular rather unusual sequence of two instructions in a program. Should that sequence appear, the added logic might disable storage protection for just a few dozen microseconds. Such a small "hole" in the hardware would be quite sufficient for the user to (1) access anyone's file; (2) cause a system crash; (3) modify anyone's file.

User-controllable changes could be implemented in many ways, with many modes of control and action besides this example (which was, however, one of the more effective schemes contemplated). Countermeasures to such controllable changes will be considered below, along with ways in which a subverter might try to anticipate countermeasures.

Countermeasures to subversion

As implied earlier, anyone who has sufficient access to the CPU to install his own "design changes" in the hardware is likely to put in a controllable change, since

a fixed change would be discovered by even a simple software test infrequently performed. A user-controllable change, on the other hand would not be discovered by tests outside the user's quantum, and would be hard to discover even within it, as will become obvious.

The automatic microprogrammed test previously discussed would have a low probability of discovering a user-controllable hardware change. Consider an attempt by a user to replace his log-in number with the log-in number of the person whose file he wants to steal. He must execute a MOVE CHARACTERS instruction of length 12 to do this, requiring only about 31 microseconds for the 360/50 CPU to perform. A microprogrammed test occurring at timer interrupts—once each 16 milliseconds—would have a low probability of discovering such a brief security breach. Increasing the test rate, though it raises the probability, raises the overhead correspondingly. A test occurring at 16 *microsecond* intervals, for example, represents a 15 percent overhead.

A reasonable question is whether a software test might do a better job of spotting user-controllable hardware changes. One would approach this task by attempting to discover changes with tests inserted in user programs in an undetectable fashion. One typical method would do this by inserting invisible breakpoints into the user's instruction stream; when they were encountered during the user's quantum, a software test of storage protection and PSW bit 15 would be performed.

A software test of this type could be written, and as will be discussed, such a software test would be difficult for a subverter to circumvent. Nevertheless, the drawbacks of this software test are severe. Reentrant code is required so that the software test can know (1) the location of the instruction stream, and (2) that no instructions are hidden in data areas. Requiring reentrant programs would in turn require minor changes to the ADEPT-50 Jovial compiler and major changes to the F-level Assembler. A small microprogram change would even be required, so that software could sense the difference between a fetch-protect interrupt and an execute-protect interrupt. Changes would be required to the ADEPT-50 SERVIS, INTRUP, DEBUG, and SKED modules. Were such a software test implemented, run-time overhead would likely be rather high for frequent breakpoint-insertions, since each breakpoint inserted would require execution of 50 or more instructions at run time. Lastly, programmers might not always like having to write reentrant code.

These implementation problems, though described for one specific test approach, are likely to extend to any test which attempts to interrupt running user pro-

grams via software, since any such test must isolate and augment every user's instruction stream.

Although the above does not demonstrate that detecting user controllable subversion hardware is impossible, it should indicate that by no means is such detection trivial. It is even harder than the above may suggest, for, as the next section indicates, while the subverter is busily building his controllable changes into the CPU he may add insult to injury by inactivating anti-failure and anti-subversion hardware.

Defeat of countermeasures

Hardware countermeasures that are easy to install are easy to remove. Redundant logic can be disabled with the same wire cutters or wire-unwrapper as other logic. This is also true of state-testing and parity-check logic. Alternatively, a clever subverter may add logic that picks up a test signal and uses it to inactivate his own logic while the test is performed. The subverter may, alternatively, arrange things so that tests always pass during his quantum.

Microprogrammed countermeasures are somewhat more difficult for the subverter to inactivate, since he must decode particular ROS addresses to know that a microprogrammed test routine has begun. Once he has done this, though, he may proceed as if he were dealing with hardware, since on a short-term basis, microprogram contents and addresses do not change. The net result of putting countermeasures in ROS rather than in logic is to raise the "work function" for subversion; that is, the subverter must do more work to achieve his goal, but he can reach it.

On the other hand, a thorough software test, such as the breakpoint test previously described, would be nearly impossible to defeat by hardware techniques. This is because of the transient nature of software: a subverter would have to decode program variables such as addresses via hardware; then even minor changes in the breakpoint-placing routines would make that hardware useless. One must recall, however, the large implementation and overhead problems inherent in a user-interrupting software test. In summary, countermeasures can be devised which have a high "work function," but they entail major costs in implementation and system efficiency.

Two assumptions have been inherent in this discussion; namely, that the subverter has both knowledge of system hardware (including subversion countermeasures) and means of changing the hardware. This need not be the case, but whether it is depends on administrative rather than technical considerations. Administrative considerations are the next subject.

Administrative policy

Special handling of hardware documentation and engineering changes may be worthwhile when commercial lines of computers are used for secure time-sharing. First, if hardware or microprograms have been added to the computer to test for failures and subversion attempts, the details of the tests should not be obtainable from the computer manufacturer's worldwide network of sales representatives. The fact that testing is done and the technical details of that testing would seem to be legitimate security objects, since a subverter can neutralize testing only if he knows of it. Classification of those documents which relate to testing is a policy question which should be considered. Likewise, redundant hardware, such as a second copy of the PSW bit 15 logic, might be included in the same category.

The second area is that of change control. Presumably the "Customer Engineer" (CE) personnel who perform engineering changes have clearances allowing them access to the hardware, but what about the technical documents which tell them what to do? A clever subverter could easily alter an engineering-change wire list to include his modifications, or could send spurious change documentation. A CE would then unwittingly install the subverter's "engineering change." Since it is asking too much to expect a CE to understand on a wire-by-wire basis each change he performs, some new step is necessary if one wants to be sure that engineering changes are made for technical reasons only. In other words, the computer manufacturer's engineering changes are security objects in the sense that their integrity must be guaranteed. Special paths of transmittal and post-installation verification by the manufacturer might be an adequate way to secure engineering changes; there are undoubtedly other ways. It is clear that a problem exists.

Finally, it should be noted that the 360/50 ROS storage elements, or any equivalent parts of another manufacturer's hardware that contain all system microprogramming, ought to be treated in a special manner, such as physically sealing them in place as part of hardware certification. New storage elements containing engineering changes are security objects of even higher order than regular engineering-change documents, and should be handled accordingly, from their manufacture through their installation.

GENERALIZATIONS AND CONCLUSIONS

Some general points about hardware design that relate to secure time-sharing and some short-range and long-range conclusions are the topics of this section.

Fail-secure vs. fail-soft hardware

Television programs, novels, and motion pictures have made it well known that if something is "fail-safe," it doesn't blow up when it fails. In the same vein, designers of high-reliability computers coined the term "fail-soft" to describe a machine that degrades its performance when a failure occurs, instead of becoming completely useless. It is now proposed to add another term to this family: "Fail-secure: to protect secure information regardless of failure."

The ability to detect failures is a prerequisite for fail-secure operation. However, all system provisions for corrective action based on failure detection must be carefully designed, particularly when hardware failure correction is involved. Two cases were recently described wherein a conflict arose between hardware and software that had been included to circumvent failures.* Automatic correction hardware could likewise mask problems which should be brought to the attention of the System Security Officer via security software.

Clearly, something between the extremes of system crash and silent automatic correction should occur when hardware fails. Definition of what *does* happen upon failure of critical hardware should be a design requirement for fail-secure time-sharing systems. Fail-soft computers are not likely to be fail-secure computers, nor vice versa, unless software and hardware have been designed with both concepts in mind.

Failure detection by faulty system operation

Computer hardware logic can be grouped by the system operation or operations it helps perform. Some logic—for example, the clock distribution logic—helps perform only one system operation. Other logic—such as the read-only storage address logic in the 360/50—helps perform many system operations, from floating point multiplication to memory protection interrupt handling. When logic is needed by more than one system operation, it is cross-checked for proper performance: Should an element needed for system operations A and

*At the "Workshop on Hardware-Software Interaction for System Reliability and Recovery in Fault-Tolerant Computers," held July 14-15, 1969 at Pacific Palisades, California, J. W. Herndon of Bell Telephone Labs reported that a problem had arisen in a developmental version of Bell's "Electronic Switching System." It seems that an elaborate setup of relays would begin reconfiguring a bad communications channel at the same time that software in ESS was trying to find out what was wrong. R. F. Thomas, Jr. of the Los Alamos Scientific Laboratory, having had a similar problem with a self-checking data acquisition system, agreed with Herndon that hardware is not clever enough to know what to do about system failures; software failure correction approaches are preferable.

B fail, the failure of system operation B would indicate the malfunction of this portion of operation A's logic.

Such interdependence is quite useful in a fail-secure system, as it allows failures to be detected by faulty system operation—a seemingly inelegant error detection mechanism, yet one which requires neither software nor hardware overhead. Some ideas on its uses and limitations follow.

The result of a hardware logic failure can usually be defined in terms of what happens to the system operations associated with the dead hardware. Some logic failure modes are detectable, because they make logic elements downstream misperform unrelated system operations. Analysis will also reveal failure modes which spoil only the system operation which they help perform. These failures must be detected in some other way. There are also, but more rarely, cases where a hardware failure may lead to an operation failure that is not obvious. In the 360/50, a failure could cause skipping of a segment of a control microprogram that wasn't really needed on that cycle. Such failures are not detectable by faulty system operation at least part of the time.

Advantage may be taken of this failure-detection technique in certifying hardware to be fail-secure as well as in original hardware design. In general, the more interdependencies existing among chunks of logic, the more likely are failures to produce faulty system operation. For example, in many places in a computer one finds situations as sketched in Figure 1. Therein,

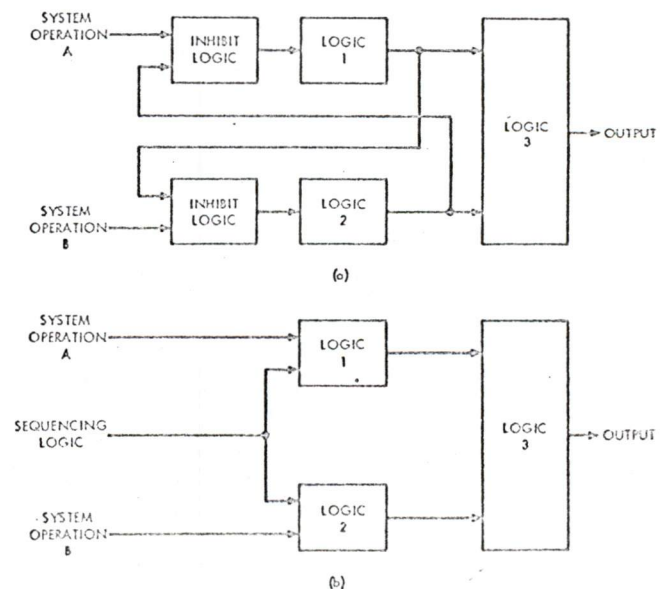


Figure 1—Inhibit logic vs sequencing logic

TABLE 1—Control Signal Error Detection by Odd Parity Check on Odd-Length Data Field

DATA BITS		MEANING
012 P		
000 0		data error or control logic error*
000 1		0
001 0		1
001 1		data error
010 0		2
010 1		data error
011 0		data error
011 1		3
100 0		4
100 1		data error
101 0		data error
101 1		5
110 0		data error
110 1		6
111 0		7
111 1		data error or control logic error**

*Control logic incorrectly set all bits to zero.
 **Control logic incorrectly set all bits to one.

System Operation A needs the services of Logic Group 1 and Logic Group 3, while System Operation B needs Logic Group 2 and Logic Group 3. Note at this point that, as above, if System Operation A doesn't work because of a failure in Logic Group 3, we have concurrently detected a failure in the logic supporting System Operation B.

A further point is made in Figure 1. Often System Operations A and B must be mutually exclusive; hardware must be added to prevent simultaneous activation of A and B. Two basic design approaches may be taken to solve this problem. An "inhibiting" scheme may be used, wherein logic is added that inhibits Logic Group 1 when Logic Group 2 is active, and vice versa. This approach is illustrated by Figure 1(a). Alternatively, a "sequencing" scheme may be used, wherein logic not directly involved with 1 or 2—such as system clock, mode selection logic, or a status register—defines when A and B are to be active. This approach is illustrated by Figure 1(b).

Now, "inhibit" logic belongs to a particular System Operation, for its function is to asynchronously, on demand, condition the hardware to perform that System Operation. It depends on nothing else; if it fails by going permanently inactive, only its System Operation is affected, and no alarm is given. On the other hand, "sequencing" logic feeds many areas of the machine; its failure is highly likely to be detected by faulty system operation.

A further point can be made here which may be somewhat controversial: that an overabundance of "inhibit"-type asynchronous logic is a good indicator of sloppy design or bad design coordination. While a certain amount must exist to deal with asynchronous pieces of hardware, often it is put in to "patch" problems that no one realized were there till system checkout time. Evidence of such design may suggest more thorough scrutiny is desirable.

System Operations can be grouped by their frequency of occurrence: some operations are needed every CPU cycle, some when the programmer requests them, some only during maintenance, and so on. Thus, some logic which appears to provide a cross-check on other logic may not do so frequently or predictably enough to satisfy certification requirements.

To sum up, the fact that a system crashes when a hardware failure occurs, rather than "failing soft" by continuing to run without the dead hardware, may be a blessing in disguise. If fail-soft operation encompasses hardware that is needed for continued security, such as the memory protection hardware, fail-soft operation is not fail-secure.

Data checking and control signal errors

Control signals which direct data transfers will often be checked by logic that was put in only to verify data purity. The nature and extent of this checking is dependent on the error-detection code used and upon the length of the data field (excluding check bits).

What happens is that if logic fails which controls a data path and its check bits, the data will be forced to either all zeros or all ones. If one or both of these cases is illegal, the control logic error will be detected when the data is checked. (Extensive parity checking on the 360/50 CPU results in much control logic failure detection capability therein.) Table 1 demonstrates an example of this effect; Table 2 describes the conditions for which it exists for the common parity check.

TABLE 2—Control Signal Error Detection by Parity Checking

DATA FIELD LENGTH:	PARITY:	CONTROL LOGIC ERROR CAUSES:	
		all zeros	all ones
even	odd	CAUGHT	MISSED
even	even	MISSED	CAUGHT
odd	odd	CAUGHT	CAUGHT
odd	even	MISSED	MISSED

CONCLUSIONS

From a short-range viewpoint, 360/50 CPU hardware has some weak spots in it but no holes, as far as secure time-sharing is concerned. Furthermore, the weak spots can be reinforced with little expense. Several alternatives in this regard have been described.

From a longer-range viewpoint, anyone who contemplates specifying a requirement for hardware certification should know what such an effort involves. As reference, some notes are appropriate as to what it took to examine the 360/50 memory protection system to the level required for meaningful hardware certification. The writer first obtained several publications which describe the system. Having read these, the writer obtained the logic diagrams, went to the beginning points of several operations, and traced logic forward. Signals entering a point were traced backward until logic was found which would definitely cause faulty machine operation outside the protection system if it failed. During this tedious process, discrepancies arose between what had been read and what the logic diagrams appeared to show. Some discrepancies were resolved by further study; some were accounted for by special features on the SDC 360/50; some remain.

After logic tracing, the entire protection system was sketched out on eight $8\frac{1}{2} \times 11$ pages. This drawing proved to be extremely valuable for improving the writer's understanding, and enabled failure-mode charting that would have been intractable by manual means from the manufacturer's logic diagrams.

For certifying hardware, documentation quality and currentness is certainly a problem. The manufacturer's publications alone are necessary but definitely not sufficient, because of version differences, errors, oversimplifications, and insufficient detail. Both these and machine logic diagrams are needed.

Though the hardware certification outlook is bleak, an alternative does exist: testing. As previously described, it is possible to require inclusion of low-overhead functional testing of critical hardware in a secure

computing system. The testing techniques, whether embedded in hardware, microprograms, or software, could be put under security control if some protection against hardware subversion is desired. Furthermore, administrative security control procedures should extend to "Customer Engineer" activity and to engineering change documentation to the extent necessary to insure that hardware changes are made for technical reasons only.

Careful control of access to computer-based information is, and ought to be, of general concern today. Access controls in a secure time-sharing system such as ADEPT-50 are based on hardware features.⁷ The latter deserve scrutiny.

REFERENCES

- 1 L. MOLHO
Hardware reliability study
SDC N-(L)-24276/126/00 December 1969
- 2 R. LINDE, C. WEISSMAN, C. FOX
The ADEPT-50 time-sharing system
Proceedings of the Fall Joint Computer Conference Vol 35
p 39-50 1969
Also issued as SDC document SP-3344
- 3 W. H. WARE
Security and privacy in computer systems
Proceedings of the Spring Joint Computer Conference
Vol 30 p 279-282 1967
- 4 W. H. WARE
Security and privacy: Similarities and differences
Proceedings of the Spring Joint Computer Conference
Vol 30 p 287-290 1967
- 5 S. G. TUCKER
Microprogram control for system/360
IBM Systems Journal Vol 6 No 4 p 222-241 1967
- 6 G. C. VANDLING, D. E. WALDECKER
The microprogram control technique for digital logic design
Computer Design Vol 8 No 8 p 44-51 August 1969
- 7 C. WEISSMAN
Security controls in the ADEPT-50 time-sharing system
Proceedings of the Fall Joint Computer Conference Vol 35
p 119-133 1969
Also issued as SDC document SP-3342

Security and privacy: similarities and differences

by WILLIS H. WARE
The RAND Corporation
Santa Monica, California

For the purposes of this paper we will use the term "security" when speaking about computer systems which handle classified defense information, and "privacy" in regard to those computer systems which handle non-defense information which nonetheless must be protected because it is in some respect sensitive. It should be noted at the outset that the context in which security must be considered is quite different from that which can be applied to the privacy question. With respect to classified military information there are federal regulations which establish authority, and discipline to govern the conduct of people who work with such information. Moreover, there is an established set of categories into which information is classified. Once information is classified Confidential, Secret, or Top Secret, there are well-defined requirements for its protection, for controlling access to it, and for transmitting it from place to place. In the privacy situation, analogous conditions may exist only in part or not at all.

There are indeed Federal and State statutes which protect the so-called "secrecy of communication." But it remains to be established that these laws can be extended to cover or interpreted as applicable to the unauthorized acquisition of information from computer equipment. There are also laws against thievery; and at least one case involving a programmer and theft of privileged information has been tried. The telephone companies have formulated regulations governing the conduct of employees (who are subject to "secrecy of communication" laws) who may intrude on the privacy of individuals; perhaps this experience can be drawn upon by the computer field.

Though there apparently exist fragments of law and some precedents bearing on the protection of information, nonetheless the privacy situation is not so neatly circumscribed and tidy as the security situation. Privacy simply is not so tightly controlled. Within computer networks serving many companies, organi-

zations, or agencies, there may be no uniform governing authority; an incomplete legal framework; no established discipline, or perhaps not even a code of ethics among users. At present there is not even a commonly accepted set of categories to describe levels of sensitivity for private information.

Great quantities of private information are being accumulated in computer files; and the incentives to penetrate the safeguards to privacy are bound to increase. Existing laws may prove inadequate, or may need more vigorous enforcement. There may be need for a monitoring and enforcement establishment analogous to that in the security situation. In any event, it can not be taken for granted that there now exist adequate legal and ethical umbrellas for the protection of private information.

The privacy problem is really a spectrum of problems. At one end, it may be necessary to provide only a very low level of protection to the information for only a very short time; at the opposite end, it may be necessary to invoke the most sophisticated techniques to guarantee protection of information for extended periods of time. Federal regulations state explicitly what aspect of national defense will be compromised by unauthorized divulgence of each category of classified information. There is no corresponding particularization of the privacy situation; the potential damage from revealing private information is nowhere described in such absolute terms. It may be that a small volume of information leaked from a private file may involve inconsequential risk. For example, the individual names of a company's employees is probably not even sensitive, whereas the complete file of employees could well be restricted. Certainly the "big brother" spectre raised by recent Congressional hearings on "invasion of privacy" via massive computer files is strongly related to the volume of information at risk.

Because of the diverse spread in the privacy situation, the appearance of the problem may be quite different from its reality. One would argue on principle that maximum protection should be given to all information labeled private; but if privacy of information is not protected by law and authority, we can expect that the owner of sensitive information will require a system designed to guarantee protection only against the threat as he sees it. Thus, while we might imagine very sophisticated attacks against private files, the reality of the situation may be that much simpler levels of protection will be accepted by the owners of the information.

In the end, an engineering trade-off question must be assessed. The value of private information to an outsider will determine the resources he is willing to expend to acquire it. In turn, the value of the information to its owner is related to what he is willing to pay to protect it. Perhaps this game-like situation can be played out to arrive at a rational basis for establishing the level of protection. Perhaps a company or governmental agency—or a group of companies or agencies, or the operating agent of a multi-access computer service—will have to establish its own set of regulations for handling private information. Further, a company or agency may have to establish penalties for infractions of these regulations, and perhaps even provide extra remuneration for those assuming the extraordinary responsibility of protecting private information.

The security measures deemed necessary for a multi-processing remote terminal computer system operating in a military classified environment have been discussed in the volume.* This paper will compare the security situation with the privacy situation, and suggest issues to be considered when designing a computer system for guarding private information. Technology which can be applied against the design problem is described elsewhere.†

First of all, note that the privacy problem is to some extent present whenever and wherever sharing of the structures of a computer system takes place. A time-sharing system slices time in such a way that each user gets a small amount of attention on some periodic basis. More than one user program is resident in the central storage at one time; and hence, there are obvious opportunities for leakage of information from one program to another, although the problem is alleviated to some extent in systems operating in an interpretive software mode. In a multi-programmed

computer system it is also true that more than one user program is normally resident in the core store at a time. Usually, a given program is not executed without interruption; it must share the central storage and perhaps other levels of storage with other programs. Even in the traditional batch-operated system there can be a privacy problem. Although only one program is usually resident in storage at a time, parts of other programs reside on magnetic tape or discs; in principle, the currently executing program might accidentally reference others, or cause parts of previous programs contained on partially re-used magnetic tape to be outputted.

Thus, unless a computer system is completely stripped of other programs—and this means clearing or removing access to all levels of storage—privacy infractions are possible and might permit divulgence of information from one program to another.

Let us now reconsider the points raised in the Peters* paper and extend the discussion to include the privacy situation.

(1) The problem of controlling user access to the resource-sharing computer system is similar in both the security and privacy situations. It has been suggested that one-time passwords are necessary to satisfactorily identify and authenticate the user in the security situation. In some university time-sharing systems, permanently assigned passwords are considered acceptable for user identification. Even though printing of a password at the console can be suppressed, it is easy to ascertain such a password by covert means; hence, repeatedly-used passwords may prove unwise for the privacy situation.

(2) The incentive to penetrate the system is present in both the security and privacy circumstances. Revelation of military information can degrade the country's defense capabilities. Likewise, divulgence of sensitive information can to some extent damage other parties or organizations. Private information will always have some value to an outside party, and it must be expected that penetrations will be attempted against computer systems handling such information. It is conceivable that the legal liability for unauthorized leaking of sensitive information may become as severe as for divulging classified material.

(3) The computer hardware requirements appear to be the same for the privacy and security situations. Such features as memory read-write protection, bounds registers, privileged instructions, and a privileged mode of operation are required to protect

*Peters, B., "Security Considerations in a Multi-Programmed System".

†Petersen, H. E., and R. Turn, Systems Implications of Privacy."

*Peters, B., *loc cit.*

information, be it classified or sensitive. Also, overall software requirements seem similar, although certain details may differ in the privacy situation because of communication matters or difference in user discipline.

(4) The file access and protection problem is similar under both circumstances. Not all users of a shared computer-private system will be authorized access to all files in the system, just as not all users of a secure computer system will be authorized access to all files. Hence, there must be some combination of hardware and software features which controls access to the on-line classified files in conformance with security levels and need-to-know restrictions and in conformance with corresponding attributes in the privacy situation. As mentioned earlier, there may be a minor difference relative to volume. In classified files, denial of access must be absolute, whereas in private files access to a small quantity of sensitive information might be an acceptable risk.

(5) The philosophy of the overall system organization will probably have to be different in the privacy situation. In the classified defense environment, users are indoctrinated in security measures and their personal responsibility can be considered as part of the system design. Just as the individual who finds a classified document in a hallway is expected to return it, so the man who accidentally receives classified information at his console is expected to report it. The users in a classified system are subject to the regulations, authority, and discipline of a governmental agency. Similar restrictions may not prevail in a commercial or industrial resource-sharing computer network, nor in government agencies that do not operate within the framework of government classification. In general, it would appear that one cannot exploit the good will of users as part of a privacy system's design. On the other hand, the co-operation of users may be part of the design philosophy if it proves possible to impose a uniform code of ethics, authority, and discipline within a multi-access system. Uniform rules of behavior might be possible if all users are members of the same organization, but quite difficult or impossible if the users are from many companies or agencies.

(6) The certifying authority is certainly different in the two situations. It is easy to demonstrate that the total number of internal states of a computer is so enormous that some of them will never prevail in the lifetime of the machine. It is equally easy to demonstrate that large computer programs have a large number of internal paths, which implies the potential existence of error conditions which may appear rarely or even only once. Monitor programs

governing the internal scheduling and operation of multi-programmed, time-sharing or batch-operated machines are likely to be extensive and complex; and if security or privacy is to be guaranteed, some authority must certify that the monitor is properly programmed and checked out. Similarly, the hardware must also be certified to possess appropriate protective devices.

In a security situation, a security officer is responsible for establishing and implementing measures for the control of classified information. Granted that he may have to take the word of computer experts or become a computer expert himself, and granted that of itself his presence does not solve the computer security problem, there is nonetheless at least an assigned, identifiable responsible authority. In the case of the commercial or industrial system, who is the authority? Must the businessman take the word of the computer manufacturer who supplied the software? If so, how does he assure himself that the manufacturer hasn't provided "ins" to the system that only he, the manufacturer, knows about? Must the businessman create his own analog of defense security practices?

(7) Privacy and security situations are certainly similar in that deliberate penetrations must be anticipated, if not expected; but industrial espionage against computers may be less serious. On the other hand, industrial penetrations against computers could be very profitable and perhaps safer from a legal viewpoint.

It would probably be difficult for a potential penetrator to mount the magnitude of effort against an industrial resource-sharing computer system that foreign agents are presumed to mount against secrecy systems of other governments. To protect against large-scale efforts, an industry-established agency could keep track of major computing installations and know where penetration efforts requiring heavy computer support might originate. On the other hand, the resourceful and insightful individual can be as great a threat to the privacy of a system. If one can estimate the nature and extent of the penetration effort expected against an industrial system, perhaps it can be used as a design parameter to establish the level of protection for sensitive information.

(8) The security and privacy situations are certainly similar in that each demands secure communication circuits. For the most part, methods for assuring the security of communication channels have been the exclusive domain of the military and government. What about the non-government user? Could the specifications levied on common carriers in their

implied warranty of a private circuit be extended? Does the problem become one for the common carriers? Must they develop communication security equipment? If the problem is left to the users, does each do as he pleases? Might it be feasible to use the central computer itself to encode information prior to transmission? If so, the console will require special equipment for decoding the messages.

(9) Levels of protection for communications are possibly different in the two situations. If one believes that a massive effort at penetration could not be mounted against a commercial private network, a relatively low-quality protection for communication would be sufficient. On the other hand, computer networks will inevitably go international. Then what? A foreign industry might find it advantageous to tap the traffic of U.S. companies operating an international and presumably private computer network. Might it be that for reasons of national interest we will someday find the professional cryptoanalytic effort of a foreign government focused on the privacy-protecting measures of a computer network?

If control of international trade were to become an important instrument of government policy, then any international communications network involved with industrial or commercial computer-private systems will need the best protection that can be provided.

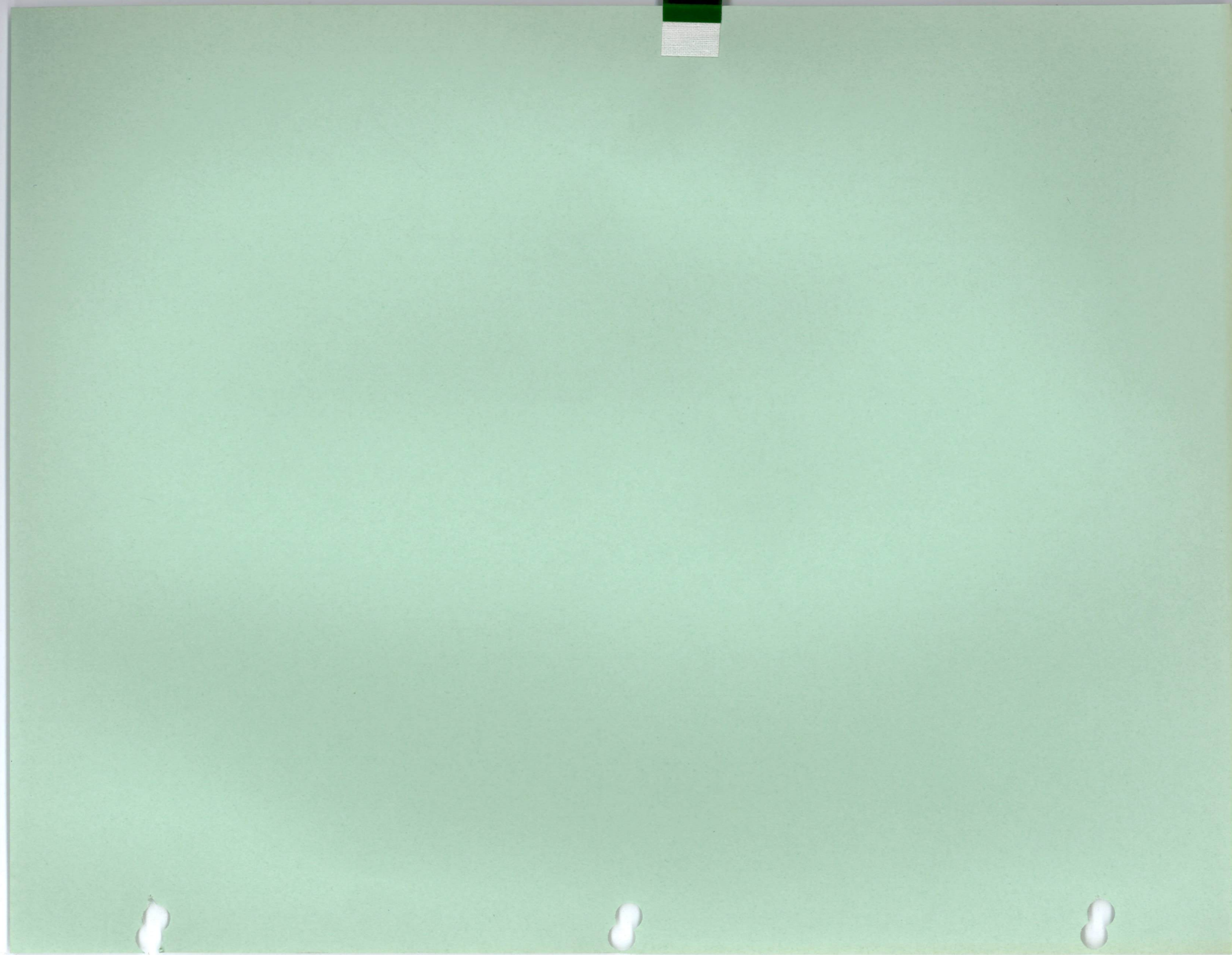
This paper has attempted to identify and briefly discuss the differences and similarities between computer systems operating with classified military information and computer systems handling private or sensitive information. Similar hardware and software and systems precautions must be taken. In most respects, the differences between the two situations are only of degree. However, there are a few aspects in which the two situations genuinely differ in kind, and on these points designers of a system must take special note. The essential differences between the two situations appear to be the following:

(1) Legal foundations for protecting classified information are well established, whereas in

the privacy situation a uniform authority over users and a penalty structure for infractions are lacking. We may not be able to count on the good will and disciplined behavior of users as part of the protective measures.

- (2) While penetrations can be expected against both classified and sensitive information, the worth of the material at risk in the two situations can be quite different, not only to the owner of the data but also to other parties and to society.
- (3) The magnitude of the resources available for protection and for penetration are markedly smaller in the privacy situation.
- (4) While secure communications are required in both situations, there are significant differences in details. In the defense environment, protected communications are the responsibility of a government agency, appropriate equipment is available, and the importance of protection over-rides economic considerations. In the privacy circumstance, secure satisfactory communication equipment is generally not available, and the economics of protecting communications is likely to be more carefully assessed.
- (5) Some software details have to be handled differently in the privacy situation to accommodate differences in the security of communications.

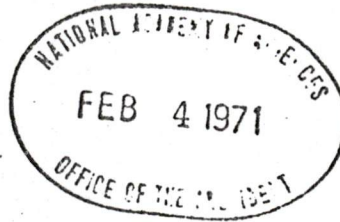
It must be remembered that since the Federal authority and regulations for handling classified military information do not function for private or sensitive information, it does not automatically follow that a computer network designed to safely protect classified information will equally well protect sensitive information. The all important difference is that the users of a computer-private network may not be subject to a common authority and discipline. But even if they are, the strength of the authority may not be adequate to deter deliberate attempts at penetration.





NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICE OF THE ADMINISTRATOR



14007
FEB 5 REC'D

FEB -2 1971

Dr. Philip Handler
President
National Academy of Sciences
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

Dear Dr. Handler:

On January 29, 1971, NASA announced the signing of a NASA-ARPA agreement whereby the Ames Research Center will act as the host site for a powerful new computer, Illiac IV, developed by the University of Illinois under contract to ARPA. The computer, unique in its capability to accomplish parallel array processing, will be used in support of ARPA sponsored research, and by Ames in the field of computational fluid dynamics.

A copy of the Agreement is enclosed for your information,

Sincerely yours,

Homer E. Newell
Associate Administrator

Enclosure

Copy: Mr. Warren House ✓
Dr. Hugh Odishaw

MEMORANDUM OF UNDERSTANDING
BETWEEN
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AND
ADVANCED RESEARCH PROJECTS AGENCY
CONCERNING
THE ILLIAC IV COMPUTER SYSTEM

I. Background and Purpose

The Advanced Research Projects Agency (ARPA) of the Department of Defense conducts research in information processing technology. A specific product of this research is an advanced prototype computer, ILLIAC IV, utilizing parallel processing as a computational technique. ILLIAC IV is in final stages of assembly by Burroughs Corporation under contracts sponsored by ARPA.

The objectives of the ILLIAC IV development program are these:

1. To successfully demonstrate the efficiency and versatility of parallel array processing.
2. To make this demonstration utilizing a sufficiently powerful hardware/software system such that the cost effectiveness and importance of array processing is adequately visible.
3. To permit a variety of DoD, NASA and private sector activities to utilize the initial system sufficiently to develop and test software, evaluate the usefulness of array processing for their needs, and to solve a series of practical problems beyond the capabilities of other machines.

ILLIAC IV will be operated as a continuation of the ARPA computer research and development program with its primary goal being to define the operating envelope of the machine. Problems to be studied on the prototype machine will include global atmosphere modeling, weather prediction, fluid dynamic problems, radar signal processing and other problems amenable to parallel processing and which further the objectives of the research and development program.

ARPA has requested the assistance of the National Aeronautics and Space Administration (NASA), as provided for in paragraphs II and III below, in the completion, installation and operation of the ILLIAC IV Computer System. It is the understanding and agreement of the parties that the assistance referred to herein will be furnished by the NASA-Ames Research Center, Moffett Field, California, in accordance with the attached NASA proposal dated October 30, 1970.

II. Responsibilities

A. NASA will:

1. Provide facilities at Ames Research Center to house the ILLIAC IV Computer System, as outlined in the attached NASA proposal, subject to availability of funds.
2. Provide technical and other services relative to the ILLIAC IV Computer System, as outlined in the attached NASA proposal.
3. Inform ARPA, on a quarterly basis, of all costs incurred under this memorandum and chargeable to ARPA in accordance with Section III-B below. Reports will be rendered by the Research Support Directorate, Ames Research Center.

- B. ARPA will provide overall technical guidance relative to the completion and installation of the ILLIAC IV Computer System.
- C. ARPA and NASA jointly shall establish all policies and procedures relative to the acceptance, management, use and operation of the ILLIAC IV Computer System.

III. Funding

- A. NASA will fund the facilities referred to in Section II-A-1 above, including special construction and equipment items, and will provide the associated utilities required. NASA will gain right to 18% of available user time on ILLIAC IV based on the following investment items totalling \$2,850,000:
 - 1. Contribution of \$2 million to ARPA which represents an investment as a user in the hardware costs of the ILLIAC IV Computer System.
 - 2. Interactive graphics equipment or other peripheral hardware, as agreed upon by ARPA and Ames Research Center, not to exceed \$400,000 in cost.
 - 3. Special construction and equipment items, totalling approximately \$450,000 referred to in III-A above and included in the facility to house the ILLIAC IV, e.g., computer air conditioning equipment, the computer floor, and fire protection equipment.
- B. Except for those costs to be funded by NASA in accordance with Section III-A, above, ARPA will be responsible for all costs, including but not limited to the following:

1. Costs arising from the current contracts with University of Illinois (and subcontracts) for the development of hardware and software systems of ILLIAC IV.
2. Costs (exclusive of civil service salaries and utilities) incurred by the host installation (NASA-Ames Research Center) in carrying out jointly-approved programs for the future development of hardware and software systems of ILLIAC IV.
3. Costs (exclusive of civil service salaries and utilities) incurred by the host installation associated with completion, delivery, installation, maintenance, and operations (including user services) of ILLIAC IV.

IV. General

- A. All assistance to be provided by NASA under this memorandum will be performed in accordance with the provisions of the attached NASA proposal.
- B. Each party assumes responsibility, when physical possession is taken, for safeguarding classified information and material received from the other party. Such safeguarding will be in accordance with the regulations of the receiving party.
- C. This Memorandum of Understanding will remain in force and effect for five years, unless terminated by joint agreement.

- D. With respect to administration of this memorandum, including responsibilities in paragraph IIC, the point of contact in ARPA will be the Director, Information Processing Techniques, and in NASA, the Director, Research Support, Ames Research Center.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Hans Mark

Dr. Hans M. Mark, Director
NASA, Ames Research Center

Date: January 26, 1971

APPROVED: *Jacob E. Smart*

Jacob E. Smart
Assistant Administrator for DOD and
Interagency Affairs, NASA Headquarters

Date: January 29, 1971

S. J. Lukasik

S. J. Lukasik
Acting Director
Advanced Research Projects Agency

Date: 29 January 1971