NEW OPPORTUNITIES FOR SOFTWARE PRODUCTIVITY IMPROVEMENT

INPUT

About INPUT

INPUT provides planning information, analysis, and recommendations to managers and executives in the information processing industries. Through market research, technology forecasting, and competitive analysis, INPUT supports client management in making informed decisions. Continuing services are provided to users and vendors of computers, communications, and office products and services.

The company carries out continuous and in-depth research. Working closely with clients on important issues, INPUT's staff members analyze and interpret the research data, then develop recommendations and innovative ideas to meet clients' needs.

Clients receive reports, presentations, access to data on which analyses are based, and continuous consulting.

Many of INPUT's professional staff members have nearly 20 years' experience in their areas of specialization. Most have held senior management positions in operations, marketing, or planning. This expertise enables INPUT to supply practical solutions to complex business problems.

Formed in 1974, INPUT has become a leading international planning services firm. Clients include over 100 of the world's largest and most technically advanced companies.

Offices

NORTH AMERICA

Headquarters
1943 Landings Drive
Mountain View, CA
94043
(415) 960-3990
Telex 171407

Detroit
220 East Huron
Suite 209
Ann Arbor, MI 48104
(313) 971-0667

New York
Park 80 Plaza West-1
Saddle Brook, NJ 07662
(201) 368-9471
Telex 134630

Washington, D.C.
11820 Parklawn Drive
Suite 201
Rockville, MD 20852
(301) 231-7350

EUROPE

United Kingdom
INPUT, Ltd.
Airwork House
35 Piccadilly
London, W1V 9PB
England
01-439-8985
Telex 23116

France
La Nacelle
Procedure d'abonnement 1-74
2, rue Campagne Premiere
75014 Paris
France
322.56.46
Telex 220064 X5533

Italy
PGP Sistema SRL
20127 Milano
Via Soperga 36
Italy
Milan 284-2850
Telex 310352

Sweden
Athena Konsult AB
Box 22232
S-104 22 Stockholm
Sweden
08-542025
Telex 17041

ASIA/AUSTRALIA

Japan
ODS Corporation
Shugetsu Building
No. 12-7 Kita Aoyama
3-Chome Minato-ku
Tokyo, 107
Japan
(03) 400-7090
Telex 26487

K.K. Ashisuto
Daini-Suzumaru Bldg., 6th Floor
8-1, Nishi Shimbashi
3-Chome Minato-ku
Tokyo, 105, Japan
(03) 437-0654
Telex 781 26196

Singapore
Cyberware Consultants (PTE) Ltd.
2902 Pangkor
Ardmore Park
Singapore 1025
734-8142

INPUT
Planning Services For Management

NEW OPPORTUNITIES FOR SOFTWARE PRODUCTIVITY IMPROVEMENT

NEW OPPORTUNITIES FOR SOFTWARE PRODUCTIVITY IMPROVEMENT

ABSTRACT

This study of productivity in the implementation of software systems extends a substantial base of past INPUT research on productivity. It concludes that the new development environment is being created through 1) the establishment of information and development centers, 2) the increased use of systems prototyping, and 3) the connection of personal computers to mainframes. This environment creates substantial threats to systems quality. Specifically, there appear to be problems associated with data and information quality, security and protection, and in systems performance at various levels in the information network.

This report analyzes these quality considerations in detail and recommends a course of action to avoid what INPUT believes to be serious threats to data bases and information flow. In addition, tools and aids required to control these problems are summarized.

This report contains 158 pages, including 33 exhibits.

U-SSP-197

NEW OPPORTUNITIES FOR SOFTWARE PRODUCTIVITY IMPROVEMENT

CONTENTS

NEW OPPORTUNITIES FOR SOFTWARE
PRODUCTIVITY IMPROVEMENT

EXHIBITS

I INTRODUCTION

I INTRODUCTION

- Definitions of productivity in software implementation and measures of programmer productivity vary tremendously, and none is satisfactory. However, the symptoms of a productivity problem are clear: continued demand for analysts and programmers, increasing backlogs of requested applications systems, reported user dissatisfaction with "responsiveness" of the information systems department, and even the proliferation of "solutions" to the productivity problem can be taken as evidence of its existence. From INPUT's perspective, the continued interest of clients in software productivity improvement has been sufficient reason to conduct substantial research in this area over the past eight years.

- The subject of software productivity improvement received high priority from our clients again this year. Three separate reports on the subject were scheduled as part of INPUT's 1984 program. As the content of these reports was being defined through client polls, it became apparent that the current shift toward more intensive end-user involvement in the system development process was of primary importance to both information systems planners and vendors of productivity improvement products and services. Therefore, it was decided to plan two of the reports so they could share a common and expanded research base.

- This, in turn, lead to two complementary reports that will directly translate user needs into more detailed vendor product definitions. Although INPUT has always stressed feedback loops among its various programs, this is the third time this year that reports have been closely interrelated. The companion to this report will be Impact of New Software Productivity Techniques (released as part of the Market Analysis and Planning Service).

- The research approach taken has also been toward close integration with other studies. The basic data base used for this report was derived from the following sources:

  - In 1979 and 1980, INPUT conducted a major multiclient study on improving productivity of systems and software implementation. Over fifty companies were visited and multiple on-site interviews were conducted; with the addition of telephone interviews, nearly 100 companies and over 200 individuals contributed data to the research base. In addition, 1,300 mailed surveys were conducted to provide a statistical base for productivity problem definition. This extensive data base (and subsequent updates) provide the foundation for current research.

  - On-site interviews with software vendors and major industry users in 1981 provided detailed data concerning specific productivity tools and aids, and market acceptance of various products and services. This study emphasized IBM's approaches to productivity improvement as a means of establishing the general environment in which specific tools, aids, and approaches to productivity problem solving would have to compete (or exist).

  - During the course of all of INPUT's software productivity studies, emphasis has been placed on personal contacts with experts in productivity improvement and with people whom we have described as "living legends" in the history of systems software development. This highly personalized research has proved to be extremely valuable in putting current hardware/software technological trends in proper perspective.

- In connection with all of these research activities, INPUT has accumulated a substantial library of software productivity information.

- The extensive information base described above is the research foundation for this study. This was supplemented by over 50 carefully selected telephone interviews with individuals who had contributed significantly to our past research efforts. These interviews were distributed as follows:

  - Thirty companies which were part of the multiclient productivity study were interviewed to update and extend the information that had previously been obtained.

  - Seven Information Systems Directors were interviewed (public utility, university, diversified manufacturer, insurance company, interstate bank, transportation company, and leading publishing and information service). The particular companies interviewed were selected based on detailed knowledge of past activities in software productivity improvement.

  - Ten computer service companies who specialized in productivity tools and aids, or services, were interviewed. They were selected based on past research and recommendations arising from current research.

  - Ten individuals prominent because of their efforts on productivity improvement were interviewed. They were selected based upon past contributions to productivity improvement and continued involvement in the productivity problems associated with today's hardware/software technological environment.

- The hardware/software technological environment that INPUT feels is of most importance today can best be characterized as follows:

- It is an IBM, SNA-oriented environment in which intelligent work-station and personal computers are being integrated with large main-frames. The purpose of such integration (linkage) is the interchange of data and information.

- End users are becoming more involved in the development of computer/communications systems because of such integration, and because of current emphasis upon information centers and systems prototyping. INPUT refers to this trend as Distributed System Development (DSD).

• The focus of this study will be on the tools and aids needed to facilitate and control systems development in such an environment.

II EXECUTIVE SUMMARY

II EXECUTIVE SUMMARY

- This chapter summarizes key forecasts, issues, and trends which are discussed in more detail in the remainder of the report.

- This Executive Summary is prepared in a presentation format; i.e., the exhibits are set in larger type for ease of use with an overhead projector and the text is in script form. The script for each exhibit is contained on the left-hand page opposite the exhibit.

A. THE BATTLE IS OVER

- A few years ago, INPUT depicted the data processing "fortress" under seige from end users. The battle concerned the lack of productivity (or responsiveness) on the part of the central data processing facility to user demands. Since then, the walls have been breeched and the drawbridge has been lowered. Users are either demanding access to the corporate treasures (data and information) or have already plundered it.

- In fact, the successor to the "data processing" department, the Information Systems (IS) department, is cooperating with users. This new spirit of cooperation is apparent through:

  - The trend towards information centers.

  - The mutual involvement of IS and end-users in system prototyping.

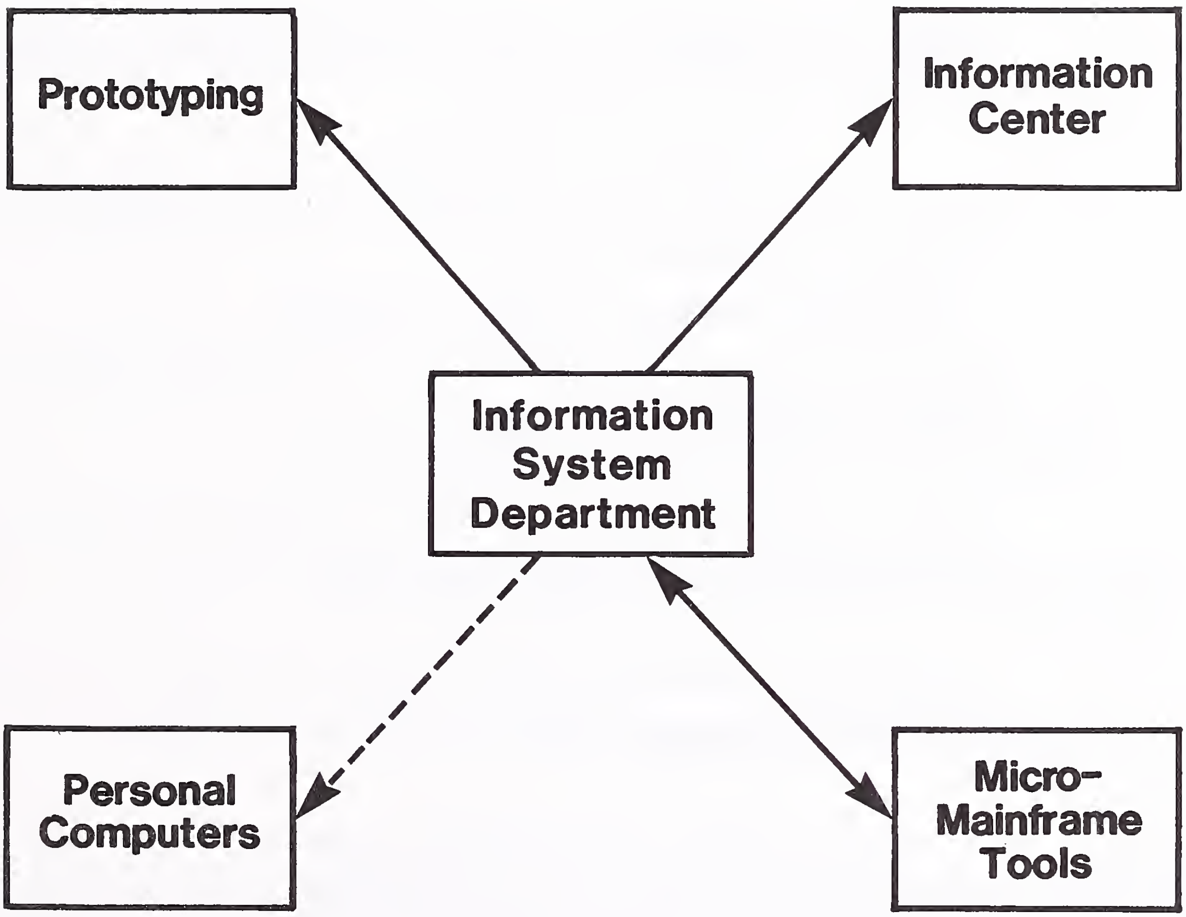
  - The linkage of micro and microcomputers to mainframes to expedite data and information flow.

  - The standalone personal computer's continued existence is a reminder of the primary user weapon.

- Now end users are actively involved in system development in a new, open environment. INPUT refers to this environment as Distributed Systems Development (DSD).

EXHIBIT II-1

THE BATTLE IS OVER



B. CONFLICTS IN THE DSD ENVIRONMENT

- However, there remain fundamentally opposed forces in the DSD environment and there appear to be inevitable conflicts:

  - Top-down systems design does not necessarily interface with bottom-up systems development.

  - Access to corporate data creates security problems, and requirements for security create access problems.

  - Ease-of-use is not always compatible with the increased functional capability of integrated systems.

  - The uncertainty of data and its accuracy is increased substantially in a distributed data base environment.

  - Micro processing demands can overload mainframes, and mainframe off-loading can cripple personal computers.

  - Management reorts from various sources in the DSD environment can be in conflict with each other.

  - The parallel trends of centralization, integration, differentiation and mechanization inherent in the DSD environment make hardware/software planning exceptionally complex.

- These conflicts will severely affect systems quality in terms of both data/information quality and systems performance, each of which can result in decreased productivity.

EXHIBIT II-2

CONFLICTS IN THE DSD ENVIRONMENT

Top-Down Design	vs.	Bottom-Up Design
Security	vs.	Access
Ease-of-Use	vs.	Added Function
Data Quality	vs.	Distributed Data Bases
Micro Demands on Mainframes	vs.	Off-Loading of Mainframes
Management Reports	vs.	Management Reports
Centralization, Integration, Differentiation, and Mechanization	vs.	Hardware/Software Planning

C. HOW USERS RATE DSD PROBLEMS

- Problems perceived by at least 75% of the respondents to this survey can be grouped into three main categories: data-base-related, information-related, and performance-related.

  - Data-base-related problems were believed to be "very serious" by at least 45% of the respondents; less than 20% stated that no data-base-related problems existed.

  - The primary concerns were: data base integrity, data base sychronization, and data security and protection.

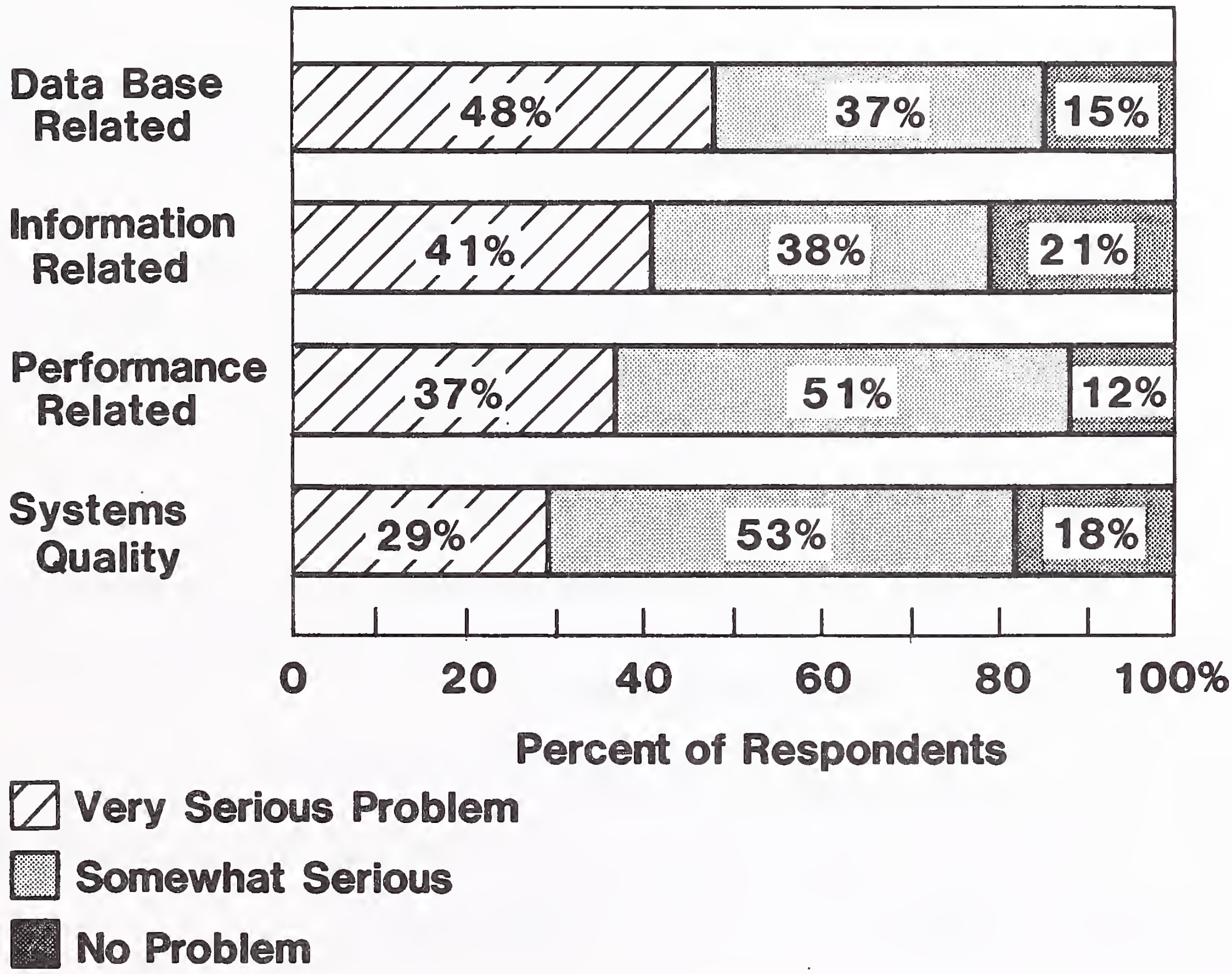
- Over 40% of respondents felt that user understanding of corporate data and conflicting reports to management will be very serious problems. In other words, end users may not understand the data they are working with at intelligent workstations, and management will receive conflicting reports in support of decision making.

- Ninety percent of respondents felt that demands for corporate data will have an adverse performance impact on mainframes. Not surprisingly, approximately the same percentage felt this will present problems in mainframe capacity planning. Over 35% of respondents felt both problems will be very serious.

- Although over 80% of the respondents felt overall systems quality would be a problem, the impact on the overall system was not considered to be as severe as the problem categories grouped above. INPUT believes this perception is erroneous and that quality of the overall system will suffer as a result of the other perceived problems and will therefore be more severe than any of the contributing factors.

EXHIBIT II-3

HOW USERS RATE DSD PROBLEMS



D. PRODUCTIVITY PRIORITIES ARE NOT CORRECT

- INPUT's exhaustive multiclient study of productivity improvement developed a productivity pyramid that emphasized that a comprehensive program of improvement must be built from the bottom up:

  - The highest priority is a commitment to quality as the base.

  - End users must be involved with IS in assuring that quality systems are developed.

  - Management at all levels must understand the commitment to improve both productivity and quality.

  - Once this plan and program of improvement has been established, effective personnel must be recruited, motivated, and retained.

  - Then the proper tools to aid productivity can be selected and introduced to assure a productive environment.

- A productivity study conducted by INPUT in 1981 discovered that commitment to quality was only rated fourth in priority for productivity improvement among the users and vendors interviewed.

- This study reveals an even more serious distortion of priorities in the DSD environment. There is undue emphasis upon tools and aids, and in the rush quality is currently receiving the lowest priority.

- It is INPUT's opinion that true productivity cannot be improved by developing systems that have the potential to lower the quality of data and information, and have unpredictable performance impacts throughout the computer/communication network.

EXHIBIT II-4

PRODUCTIVITY PRIORITIES ARE NOT CORRECT

Recommended	INPUT'S Productivity Pyramid	1982 Study	This Study
5	The Right Tools	5	2
4	Effective Personnel	3	4
3	Broad-Based Management	2	3
2	User Involvement	1	1
1	Commitment to Quality	4	5
	1979-1980	1982	1984

1 = Most Important

5 = Least Important

E. TOOLS AND AIDS ARE NEEDED TO CONTROL DSD

- There are literally hundred of tools and aids available to facilitate DSD. The primary ones being used are: fourth-generation languages, application generators, relational data base systems, and integrated PC software. These are effective, but address primarily the programming phase of systems development in the total systems life cycle.

- The wide variety of "solutions" available for information centers, prototyping, and intelligent workstation support is in itself a problem. With literally hundreds available, selection becomes a problem. To the degree that the tools supporting the DSD environment cause the potential quality problems users have identified, these tools become part of the problem.

- When users were asked what tools and aids they used for control of DSD, they were uncertain. When asked about the tools they needed for control many did not respond. As one respondent stated: "That is a good question."

- INPUT has determined that there is a primary shift in perspective required--away from concern about discrete computer systems and toward data/information flow. If information flow is to assist in the decision-making process, new and complex analysis tools from operations research (OR) and artificial intelligence (AI) must also be employed.

- To control data/information flow quality, new tools and aids are required. These tools and aids must be available in order to assure that a serious commitment to quality can even be made in a DSD environment.

EXHIBIT II-5

TOOLS AND AIDS ARE NEEDED TO CONTROL DSD

Tools and Aids for Facilitating DSD

- Fourth-Generation Languages
- Applications Generators
- Relational Data Base Systems
- Integrated PC Software

Tools and Aids to Control

?

DSD Leads to a Process

- Data/Information Flow
- Complex Analysis Tools
    - Operating Research
    - Artificial Intelligence
- New Tools for Process Quality Control Needed!

F. CATEGORIES OF TOOLS, AIDS, AND APPROACHES NEEDED

- Expanded dictionaries and directories at various levels of detail are necessary to ensure communication between data base administrators and end users in a DSD environment.

- Languages are going to proliferate but the "Tower of Babel" must be controlled if quality systems are to be developed. An understandable internal language must be developed in the DSD environment.

- The impact of data requests from intelligent workstations and data transmissions from mainframes must be monitored in order to predict impact in both directions.

- Security of information flow requires a great deal of research, but "data bank access" is no longer sufficient. Statistical analysis of authorized data use can reveal sensitive information. Attention must be given to the process rather than merely to the data base.

- In addition to data processing languages, a communications command language to facilitate control of encoded data, images, paper documents, and audio-visual information is required.

- Performance prediction and monitoring must be refined as both data and programs (processing requirements) flow through the networks.

- If unworkable systems are going to be avoided, we must develop tools to predict and analyze the performance and results of OR and AI tools themselves.

- An integrated paper and electronic document storage and control system is required to control current information flow and to prepare for future electronic document storage.

EXHIBIT II-6

CATEGORIES OF TOOLS, AIDS, AND APPROACHES NEEDED

- Dictionaries, Directories, Encyclopedias, and Glossaries
- Meta Languages - Programming and Data
- Data Flow Performance Monitors
- Integrated Security - Access Control, Information Flow Control, Data Base Certification
- Communications Command Language - For Moving and Controlling Data and Information Structures
- Processing Performance Monitors
- Tools to Analyze Tools
- Integrated Document Storage and Control System

III DISTRIBUTED SYSTEMS DEVELOPMENT

III DISTRIBUTED SYSTEMS DEVELOPMENT

A. HISTORICAL PERSPECTIVE

- Systems software is designed to help people use computers. Since we are going to focus on an IBM hardware/software environment, it is helpful to understand a little about its evolution.

  - In 1963, in preparation for the announcement of System/360, IBM surveyed all of its major customers in the United States concerning the relative importance of various attributes of systems software. The results are summarized in Exhibit III-1, and several things are clear, despite some terminology that may be unfamiliar.

    - "Ease of use" (programming) was the top-ranked attribute for all programming languages, report generators, and I/O systems (access methods and file handling).

    - "Ease of use" (operational) was the top-ranked attribute for loaders and monitors (operating systems), and for sorts was tied for first with "speed of operation."

    - Thus, "ease of use" on a combined basis was ranked number one on all of the systems software components that were evaluated, and "ease of use" (programming) had the highest mean ranking of any of the attributes (2.6).

EXHIBIT III-1

RANKINGS OF SYSTEM SOFTWARE ATTRIBUTES

(IBM Study - 1963)

ATTRIBUTE	ALGEBRAIC COMPILERS	ENGLISH COMPILER	ASSEMBLERS	I/O SYSTEMS	LOADERS	MONITORS	RPGs	SORTS	MEAN RANKING
Ease of Use (Programming)	1	1	1	1	5	5	1	6	2.6
Object Efficiency	2	2	4	4	7	8	3	7	4.6
Speed of Operation	3	3	3	3	2	3	4	1	2.8
Documentation	4	4	2	2	4	4	2	3	3.1
Debugging Aids	5	5	5	10	10	11	6	11	7.9
Diagnostics	6	7	7	11	8	10	7	9	8.1
Ease of Use (Operational)	7	8	6	6	1	1	5	1	4.4
Compatibility (Across Machines)	8	6	11	12	11	12	11	12	10.4
Ease of Modifications	9	10	8	7	6	6	8	5	7.4
Main Memory Usage (Minimize)	10	9	9	5	3	2	9	8	6.9
Modularity, of Construction	11	11	10	8	9	7	10	10	9.5
Check Point & Restart	12	12	12	9	12	9	12	4	10.3
Number of Installations Ranking	237	256	450	433	281	329	218	417	

- Although "speed of operation" was ranked first in importance only on sorts (where it tied with "ease of use" (operational)), its mean ranking of 2.8 placed it second among the attributes in order of importance for systems software.

- The term "operating system" came into vogue when System/360 was announced, and over the last two decades it has evolved from OS/360 to MVS/XA. Based on the survey "ease of use" was established as the primary design point for OS/360. This prompts the following comments:

  - At the time it was released, OS/360 was the slowest, most-difficult-to-use system yet developed.

  - It (OS/360) required the isolation of a separate set of systems programmers to attend to its generation and maintenance, and to assist the applications programmers with JCL (to say nothing of memory dumps).

  - OS/360 has evolved to MVS/XA, which is not easy to use, not fast, and whose very complexity contributes to the software development problems we are currently attempting to solve.

  - The industry is still in pursuit of "ease of use" (or "user friendly systems") and as systems users become "more human" the problem is not getting any simplier.

- There is a paradox associated with the quest for "ease of use"--as more function is added, the resulting complexity proves self-defeating. However, the poor performance (operating speed) associated with complexity is justified based on the priority given "ease of use." It is important to remember this as human interfaces are designed in today's environment.

- For those wondering about interactive computing and data base systems, IBM felt that QTAM (under the I/O System) was adequate to permit users to develop their own terminal systems, and ISAM (also under the I/O System) made a data base system unnecessary.

- The 1963 IBM study also asked for a breakdown of how time was spent during the systems development process. During INPUT's 1979-80 productivity study, one of the clients had just completed an in-depth time analysis of these systems development activities. (The client had a highly advanced development environment--terminals for all IS employees, a highly respected interactive system development support system, an internally developed DBMS of high quality, etc.) It was decided to compare the IBM results (from 357 large "commercial" installations in 1963) against the clients 1980 time distribution, as shown in Exhibit III-2.

  - The development process has remained remarkably similar in terms of time distribution. This was especially surprising because:

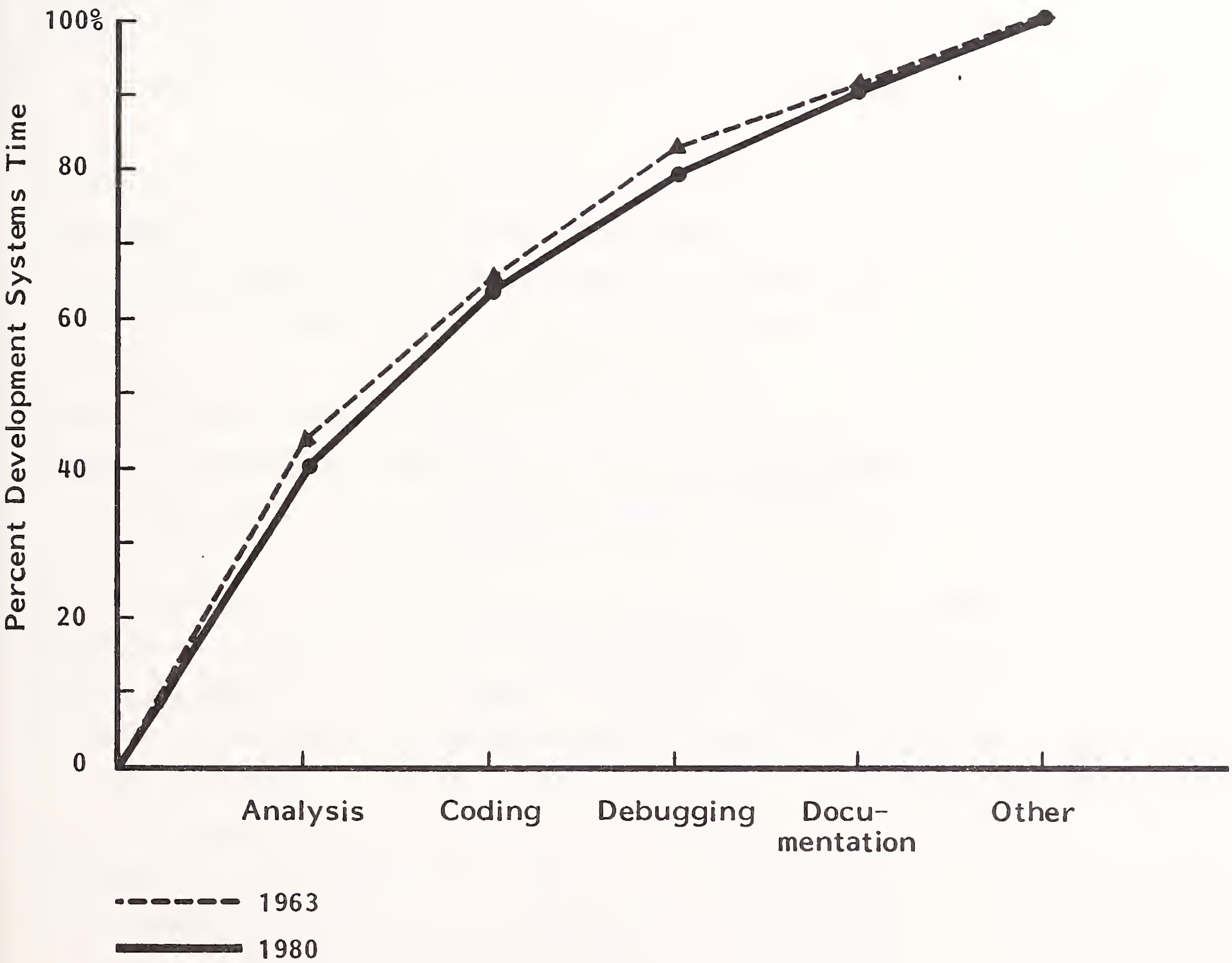
    - A leading-edge development environment has been established by the client (many of the tools and aids that have been developed for internal use have received wide external distribution).

    - The development language used in 1963 was Autocoder (assembly language), but the predominant language used by the client was COBOL (with assistance from a fourth-generation data base language).

    - The heavy concentration of tools and aids on the coding (and to a lesser extent the debugging) phase of the development process has had remarkably little impact on time distribution of the overall systems development process. (In other words, if time is being saved on coding and debugging, it is not being applied to doing a more thorough job of analysis.)

EXHIBIT III-2

TIME USAGE IN SYSTEMS DEVELOPMENT



Note: Curve is Cumulative

- However, the 1963 research also disclosed that less than 15% of total system cost was applied (or budgeted) for maintenance, and by 1980 maintenance represented over 60% of the cost (primarily personnel) over the systems life cycle. The relative costs over the systems life cycle are depicted in Exhibit III-3, and the impact of most tools and aids clearly address only a small part of the productivity problem. Some additional comments on the maintenance problem are necessary.

  . The relatively modest portion of effort budgeted for maintenance in 1963 may be partially attributed to the naive attitude that once something works it will run forever, and therefore maintenance was substantially understated.

  . However, it is also probable that the enormous increase in maintenance costs can be partially attributed to the increased effort required to "take advantage of" the latest systems software (conversion to various operating systems releases).

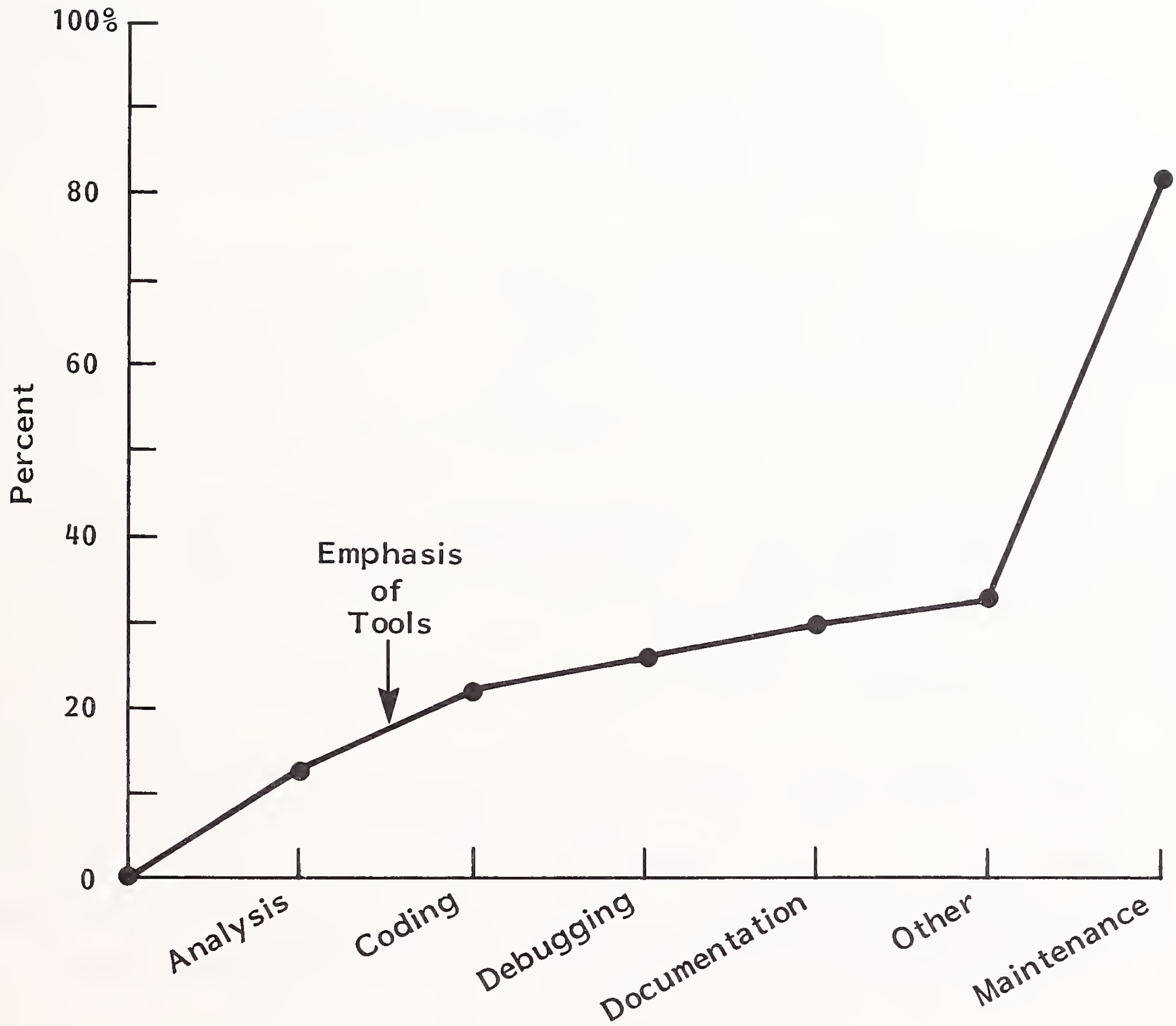
  . The distinction between development and maintenance has never been clear, and the trend toward Distributed Systems Development (DSD) may make it utterly meaningless in the future.

- There are other important elements of cost (time) distribution that were not measured in 1963 but have increased astronomically over the last two decades. These are the support functions, which have become attached to the systems development process. Specifically:

  . A special breed of systems programmers is necessary to install and maintain system software (including various productivity tools and aids), and to instruct analysts and programmers in how to function (or provide service) in the hardware/software environment.

EXHIBIT III-3

RELATIVE COSTS OVER SYSTEM LIFE



- Data base administrators are needed to develop, maintain, and document common data bases, and support systems developers.

- Miscellaneous support and planning personnel are necessary to evaluate hardware/software technology and to establish the development environment (capacity planning; tool and aid evaluation and selection; etc.)

- Although it must be assumed that these functions are necessary and contribute to the effective operation of the IS department, they do represent overhead in terms of the true cost of software implementation.

- It becomes apparent that relatively little I/S personnel time is spend actually implementing new applications. To the degree that productivity is measured by responsiveness to requests (or demands) for new applications development, the IS organization must appear to be extremely sluggish to the external observer. In many ways the typical IS organization has become comparable to the United States Army--a tremendous support organization is required for each "productive" worker (in the army's case a combat infantryman).

- There is one other disturbing similarity in the Army-IS analogy. It has been discovered that only a small percentage of infantrymen actually fire their weapons at the enemy while in combat, and INPUT has determined that the productivity of individual programmers/analysts varies by a ratio of up to 25-1. (In fact, some respondents to past research have stated that the range is infinite because "some problems would never get solved by some people.")

B. END-USER INVOLVEMENT

- INPUT's previous productivity study concluded that the entire organization and not just the "foot soldiers" must be valued in any effective strategy to improve productivity in the systems development process. This resulted in the construction of a "productivity pyramid," which was designed to depict the importance of building the strategy from the base up, as shown in Exhibit III-4.

  - A commitment to quality was established as the foundation of the pyramid, with architectural stability being of particular importance.

  - User involvement was next emphasized so that users would become: 1) involved in systems development and operation, 2) informed of what IS could or could not do for them, and 3) aware of how their needs fit into larger company requirements.

  - Broad based IS management placed emphasis on the education of both top management and users in "nontechnical IS fundamentals." (In addition, the entire study emphasized two-way communication and equal emphasis was placed on reciprocal education of IS management.)

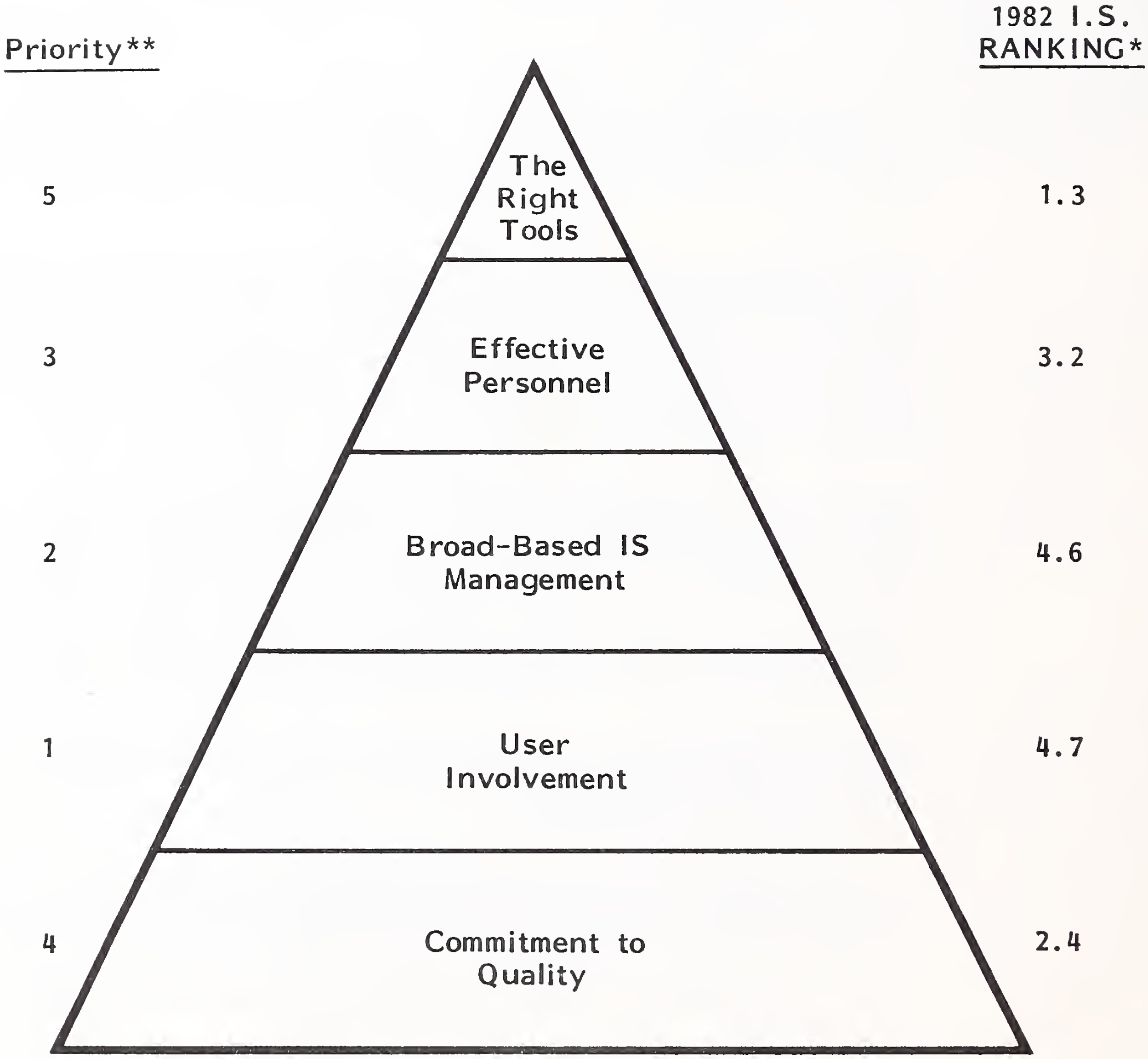
  - Effective personnel emphasized employee selection, retention, motivation, education, and training.

  - The right tools were described as a means of achieving "micro-productivity," but were considered to be relatively useless in achieving "macro-productivity," unless the other layers of the pyramid were in place.

- An INPUT research study completed in late 1982 revealed that IS management, when asked to rank the five strategic factors (five being most impor-

EXHIBIT III-4

THE PRODUCTIVITY PYRAMID



* Mean of Responses

** 1 = Most Important, 5 = Least Important

tant, and one being least important), placed the levels in proper pyramid order with one important exception: Commitment to quality was ranked only fourth in importance (see Exhibit III-4). User involvement was considered the most important factor in productivity improvement, and certainly every indication in the industry points in that direction.

- It really does not make any difference whether IS management has decided it is really important to get users involved or whether users (with their PCs and Lotus 1-2-3s) have seized the initiative themselves--the primary productivity improvement stragegy in the last five years has been to distribute systems development responsibility to end users. INPUT refers to this as "distributed systems development" (DSD), and its primary tactical manifestations are: information centers, prototyping, and micro-mainframe links.

- There are profound ramifications of this lack of commitment to quality that will be discussed later in this report (in fact, the primary emphasis will be quality control), but it is important to mention one now: top-down design made good systems sense before the term "structured programming" was even coined, and distributed systems development (DSD) promotes bottom-up design. Architectural stability in such an environment will obviously be extremely difficult even if there is commitment to quality, but awareness of the threat to information quality is absolutely essential. We will now turn to IS management's current perceptions of DSD.

C. INFORMATION CENTERS

- Three years ago, when information centers were in their infancy, a consultant in productivity improvement stated: "Information centers will raise enough questions and problems to keep us all busy for the next five years." When interviewed for this study, the same consultant stated, "I was wrong! Information centers will keep us all busy for the rest of our lives." If the recent

Information Center Conference and Exposition is any indication, we may not live to see the term defined. Information centers can be anything you choose to make them:

- A simple extension of the traditional data processing function to educate users concerning the central facility (and, hopefully, to do a little PR work).

- A separate computer facility with substantial user-oriented data bases and a rich array of user friendly tools.

- A computer store with supporting training facilities.

- An end-user computing department encompassing office automation, personal computing, and time-sharing groups--all of which were previously separate.

- The information center concept is vague in implementation, and has evidently met with opposition from data processing, corporate management, and even end users depending upon the particular circumstances.

- Information centers may be either an appendage to the DP department, a separate organization, or even distributed to particular end-user organizations.

- Nevertheless, 17 respondents to this survey state that they have information centers currently installed, four have a definite plan for implementation, and only nine state that they have no current plan. (With the vague definition that was used, it is probable that even the nine without a plan have something resembling an information center already in place.) Respondent evaluation of information centers is contained in Exhibit III-5.

EXHIBIT III-5

RESPONDENT EVALUATION OF INFORMATION CENTERS
(Installed: 17, Planned: 4, No Plan: 9)

	NUMBER OF RESPONDENTS
Advantages:	
User Exposure to DP Services, Concepts, and Problems	13
Quick Response to Simple Requests	8
End-User Education and Training	6
Single Information Source	4
Flexibility (Alternate Solution)	2
Disadvantages:	
Excess Resource Use (Human and Systems)	11
Standards, Control, and Security	5
User Expectations (Over Sold)	3
Miscellaneous	8
None	4

- The advantages of information centers from the point of view of IS management can be summarized as making their function more accessible and responsive to end users. In the process of doing this, it is anticipated (or hoped) that end users will become more involved in solving their own problems and more sympathetic toward the IS function.

- The disadvantages center around resource availability and cost; lack of standards, control, and security; and excessive user expectations. However, there were a variety of other problems mentioned:

  . Users want to do everything through the information center.

  . Users resist using the information center.

  . The DP department cannot provide the necessary resources since it does not know what is going on.

  . There is conflict between DP and the information center.

  . Users exceed their abilities and "inefficient programs" are developed.

  . Data and skilled personnel (communications) are not available to support the center.

- The vendors and experts interviewed had a variety of opinions depending upon their relationships with information centers. The most prevalent attitudes can be paraphrased as follows:

  - "It is difficult to argue with improving communications between the IS department and users."

- "It's just the latest buzz word--many companies are already providing such services."

- "Information centers started as IBM's answer to the control of personal computers, and (information centers) are evolving into local points for the sale of IBM products."

- "Information centers are an external diversion to distract the source of the unrest."

- "Five to six years ago it was structured methodologies, then there was prototyping, and then fourth-generation languages--none of these solved the problem so it was decided to try something new (information centers). All of the above are aids, not solutions, to the productivity problem."

D. PROTOTYPING

- Although the concept of prototyping is less general than information centers, there are nevertheless less several different views of the process.

  - In its simplest form, the prototype is considered a quick-and-direct throwaway.

  - However, there are those who prefer to view the prototype as re-cyclable in the sense that there is recoverable scrap value (code) that can be applied to the next system.

  - Thus, there are some who plan (or assume) that the prototype will be retainable as part of the eventual system.

- Even the term prototype has acquired an unpleasant, or wasteful, connotation for some and iterative systems development is beginning to be applied to the process.

- During the research for this report, one of the expert respondents stated: "Perhaps we should refer to it (prototyping) as eternal systems design."

• Regardless of how it is reviewed, some form of prototyping is currently being used by 14 respondents, five plan to use it in the near future, and only 11 have no current plans to try it. (Once again, with the vague definition, it is probable that even those who have no plans to use prototyping on a formal basis will probably find themselves approaching it in actual fact--current technology and tools encourage it.) The respondents' evaluation of prototyping is contained in Exhibit III-6.

  - The primary advantages of prototyping are related to getting end users involved at an early stage in the development cycle by showing them the specific information they will receive. There is general recognition that one picture (screen of information) is worth a thousand words (written specification of output). In addition, five of the respondents felt that the quality of the resulting system was improved because it truly gave the users what they wanted. However, two users stated there were no advantages to prototyping (an unusually strong response to an open-ended question).

  - The primary disadvantages centered around the excess human resources required to develop the prototype and the obvious "waste" of having to throw away at least some of the code. In addition, respondents reported that there was a natural tendency for users to say: "Well that was easy, I'll start using it tomorrow morning."

EXHIBIT III-6

RESPONDENT EVALUATION OF PROTOTYPING
(Currently Using: 14, Plan to Use: 5, No Plan: 11)

	NUMBER OF RESPONDENTS
Advantages:	
User Involvement (Clear Picture of Output)	12
Better Quality System (What User Wants)	5
Accessibility (Data and New Technology)	3
Determine Systems' Impact on Resources	2
None	2
Disadvantages:	
Excess Resource Use (Including Waste)	10
User Expectations (Including Prototype to Production)	4
Miscellaneous	5
None	5

- Miscellaneous disadvantages were concerned with: the opinion that only small systems could be prototyped, that poor quality systems got into production, that data could not be "prototyped", and that the results were "too timely" (the system got done before the supporting data were available).

- The vendor and expert opinions concerning prototyping begin to strike some of the central issues of distributed systems development:

  - Properly used prototyping can help users identify their needs by having a "physical image of their thoughts," and the human factors problems can be addressed early. Computer power becomes readily available and results are immediately available. In addition, under ideal circumstances:

    - Once user needs are identified (by the image selected), they can be verified using prototyping.

    - Analyst/programmers can use the prototyping experience to determine how to build the operational system.

    - Data base structures can be prototyped during the process.

    - Better systems should result from the use of prototyping.

  - However, as one expert stated, "the solution to the problem changes the problem"; there will be inevitable abuses of the tools and aids associated with prototyping.

    - The ease of use and flexibility demonstrated by prototyping will result in a continuing search for the perfect system and the system will never be completed (or become productive).

- "Eternal systems development" will result in planning and control information that is impossible to reconcile, audit, or (eventually) even use.

- Systems costs will be impossible to budget, monitor, control, or even determine.

- One expert respondent used the example of the "mad optometrist" who became so absorbed in the question "is it better this way or that way?" that he completely ignored the following:

  - Which line on the chart the patient was focusing on.

  - Whether the glasses were to be used for reading or driving a car.

  - What the underlying reason for the deficiency in vision might be.

  - The quality of the end product in terms of qualifying the person to drive a car or read without headaches.

E. STANDALONE PERSONAL COMPUTERS

- The standalone personal computer has created the current credibility gap between the IS departments and end users. End users armed with "cheap" desktop computers and spreadsheet software packages could get results more rapidly and more cheaply than they could through the IS department and central data processing facility. Hence this should be the future DSD direction. This view of standalone PCs, and the questions it raises concerning the productivity and cost-effectiveness of large-scale hardware/software systems has given impetus to the DSD environment analyzed in this report.

- The IS department respondents to this study are theoretically an endangered species if standalone PCs are the answer to the software productivity problem. Respondents' evaluations of standalone PCs are presented in Exhibit III-7.

- Twenty-one of the thirty respondents to this survey have standalone PCs installed and four of the remaining nine have definite plans for installations. In all probability the remaining five either have PCs installed or will have them regardless of whether the IS department knows or is trying to contain proliferation. (It should be pointed out that even if PCs are forbidden in the workplace, their use at home can create "noise" in corporate information flow.) From the respondents' point of view:

  - The advantages of standalone PCs center around giving end users total responsibility for producing the simple reports they need in their routine work, and it is assumed this will make end users more productive. (Implied in the responses was a general attitude that standalone PCs serve to keep end users busy and away from the IS department with unreasonable requests for "stupid" reports.) Some also felt that PCs were cost-effective for these simple reports and actually tended to off-load the mainframe (although it is doubtful that measurable off-loading is the basis for this advantage). Reported miscellaneous "advantages" of standalone PCs were as follows:

    - "Users can continue to work when the mainframe is down." (This is hardly a solution to mainframe reliability and availability problems.)

    - "Security." (The implication is that users can feel secure in knowing their personal files are inaccessible to others.)

    - "DP problems will now be understood by users." (A clear statement of a perceived general attitude underlying giving end-user responsibility for their own destinies.)

EXHIBIT III-7

RESPONDENT EVALUATION OF STANDALONE PCs
(Installed: 21, Planned: 4, No Plan: 5)

	NUMBER OF RESPONDENTS
Advantages:	
Give Users Responsibility for Own Destinies	11
Production of Simple Reports	5
Productivity Improvement	5
Off-Load Mainframe and Cost-Effective	5
Miscellaneous	5
Disadvantages:	
Not Integrated (Data Deficiencies)	11
Uncontrolled Growth	7
Limited Capability and "Improper" Use	5
Limited Resources	3
Miscellaneous	5
None	1

- The perceived disadvantages of standalone PCs reflect the primary concerns that have prompted the predominant hardware/software trends in the industry today (micro-mainframe links and all of their ramifications). Standalone PCs are viewed as being out of control, both in a systems sense, where they are not integrated with central processing and data facilities, and from a management perspective, where they are being acquired outside the normal budget and IS planning process. In addition, PCs are viewed as having limited capability and a high potential for being used "improperly." Reported miscellaneous disadvantages included:

  - The expense of PCs.

  - Security problems.

  - Transfer of "power" to users.

  - General problems of coordination and communication.

- The vendors and experts generally agree on the advantages and disadvantages of standalone PCs expressed by IS management and are leading the rush to link micros to mainframes. Over a year ago (March, 1983); Don Estridge, at that time Vice President of IBM's Entry System Division, stated: "The IBM PC is communications oriented. The day of the standalone is over." IBM's obvious direction since that time clearly supports this statement, and theoretically addresses the "evils" of standalone PCs.

- However, some of the experts interviewed during this study identified another significant potential problem with standalone PCs. They drew the possible parallel between the emerging supermicros and some of the attempts at decentralized data processing with minicomputers over the years. The problem is defined as follows:

- There is a tendency for "pockets of data base" to develop.

- Since information represents power within the corporation, these intra-organizational data bases are jealously guarded and used as weapons in internal political wars.

- The results are competitive data bases and alternate information sources. The quality of information will inevitably degenerate in such an environment.

- To the degree that micro-mainframe links facilitate the development of competitive data bases, they will exacerbate the problems associated with decentralized information sources. The "solution" to standalone PC problems may result in a more critical problem set.

F. MICRO-MAINFRAME LINKS

- A micro-mainframe link is the latest computer industry term looking for a concept to which it can be attached. Past examples of jargon that confused even data-processing insiders are too numerous to mention. At present, it is sufficient to state that micro-mainframe links can be anything--from something that makes a personal computer look like a dumb terminal to a theoretical, integrated, secure, distributed data base network that has yet to prove practical in terms of implementation. However, as previously discussed, micro-mainframe links will supposedly address many of the disadvantages of standalone PCs.

- Eighteen of the IS department respondents reported they already had micro-mainframe links established, seven reported that they had definite plans to connect micros to mainframes, and only five did not have any plans for such links, as shown in Exhibit III-8. The total of 25 respondents (out of 30) who will have micro-mainframe links is exactly the same number who will have PCs installed--in other words "the day of the standalone is (truly) over." The advantages and disadvantages of micro-mainframe links are reported to be as follows:

- The overwhelming advantage of micro-mainframe links was reported to be in giving personal computers ready access to organization data (corporate or centralized data bases). Twenty respondents reported some form of data accessibility as being the primary impetus for such networking; although this is not surprising, there was a vagueness concerning the specific expectations of micro-mainframe links that cannot help but make one uneasy.

- A few respondents (five) felt that micro-mainframe links could significantly off-load mainframes by transferring processing to the intelligent workstations. The miscellaneous advantages mentioned were as follows:

    - Small systems could be developed and exchanged between micro and mainframe.

    - There would be improved productivity for end users.

    - A true multifunction workstation would be possible.

    - The workstation could continue to function when the mainframe was down.

- The disadvantages attributed to micro-mainframe links centered around the potential problems of security, protection, and integrity of the central data base (and the distributed data), and the problems of integrating the distributed data bases with the central data base. Only

EXHIBIT III-8

RESPONDENT EVALUATION OF MICRO-MAINFRAME LINKS
(Installed: 18, Planned: 7, No Plan: 5)

	NUMBER OF RESPONDENTS
Advantages:	
Personal Computer Access to Organized Data	20
Off-Load Mainframe	5
Miscellaneous	4
No Proven Advantage	1
Disadvantage:	
Security, Protection, Integrity, and Integration (Data Base)	8
"A Base" Mainframe Capacity	5
Invalid Reports	3
Difficult to Use	2
Miscellaneous	6
None	7

a few (three) respondents mentioned "invalid" reports as a possible result of the perceived problems of quality control associated with the dymanics of data exchange between (and among) central and distributed data bases. Five respondents felt that managing this exchange of data might result in "abuse" of mainframe capacity. (An exact off-set to the five respondents who felt off-loading the mainframe would be an advantage.) Miscellaneous disadvantage included:

- Potential impact on ease-of-use (the links would be difficult to use).

- The IBM PC was "slower" than a conventional terminal when linked to the mainframe.

- Paper would still have to be handled (report generation at local level would not diminish paper flow).

- File transfer would prove expensive.

- Programmers should be involved in the development of systems.

- There would be no standards for systems development.

- Vendors and consultants shared some of the same feelings expressed by IS respondents. This is not too surprising because both vendors and consultants have contributed to the popularity of micro-mainframe links. However, there were some different perceptions of the significance of micro-mainframe links:

  - IBM's implementation of micro-mainframe links under SNA could be used as a multifunction weapon to combat:

    - Competitive personal computers (and software).

- Minicomputers and UNIX in the network processing hierarchy.

- Effective off-loading of mainframes.

- Alternative data base strategies.

- Competition in general (by adding sufficient complexity to make the exercise of account control easier).

- Most vendors and all experts interviewed were in general agreement that the technical problems of distributed data bases are nontrivial and that a rush to DSD through the implementations of micro-mainframe links would result in a serious threat to the quality of management information.

- All of the experts agreed that micro-mainframe links were going to be expensive in terms of the hardware/software required for implementation. (However, the degree of concern for the expense varied from "you have to pay for progress" to "it could be a disaster for the total information systems budget.") It was agreed that such expense would have to be justified through improved end-user productivity (rather than through savings in the traditional IS budgets).

- One expert commenting on micro-mainframe links as a "solution" to the problems and disadvantages of standalone personal computers stated: "Once you disconnect, you have a standalone, and once you have the data (from the central data base) you can process at home or on a plane. If the IS function thinks micro-mainframe links are going to give them control of PCs, they are crazy."

G. DSD IMPACT ON SYSTEMS DEVELOPMENT AND MAINTENANCE

I. I.S. MANAGEMENT'S VIEW OF IMPACT OF DSD

- The IS respondents questionnaire that was used in this research (see Appendix A) used open-ended questions to solicit the advantages and disadvantages of DSD. Just prior to those questions they were asked for some general questions concerning the potential impact of DSD. The simple yes/no answers are presented in Exhibit III-9.

  - Generally speaking it appears that DSD has helped communications and relations between the IS department and end users (at least in the minds of the IS respondents).

  - However, the actual impact of DSD on productivity measures is not conclusive:

    - Fifty-nine percent stated the backlog had been reduced, but 41% disagreed.

    - Fifty-seven percent stated systems were developed more rapidly in a DSD environment, but 43% disagreed.

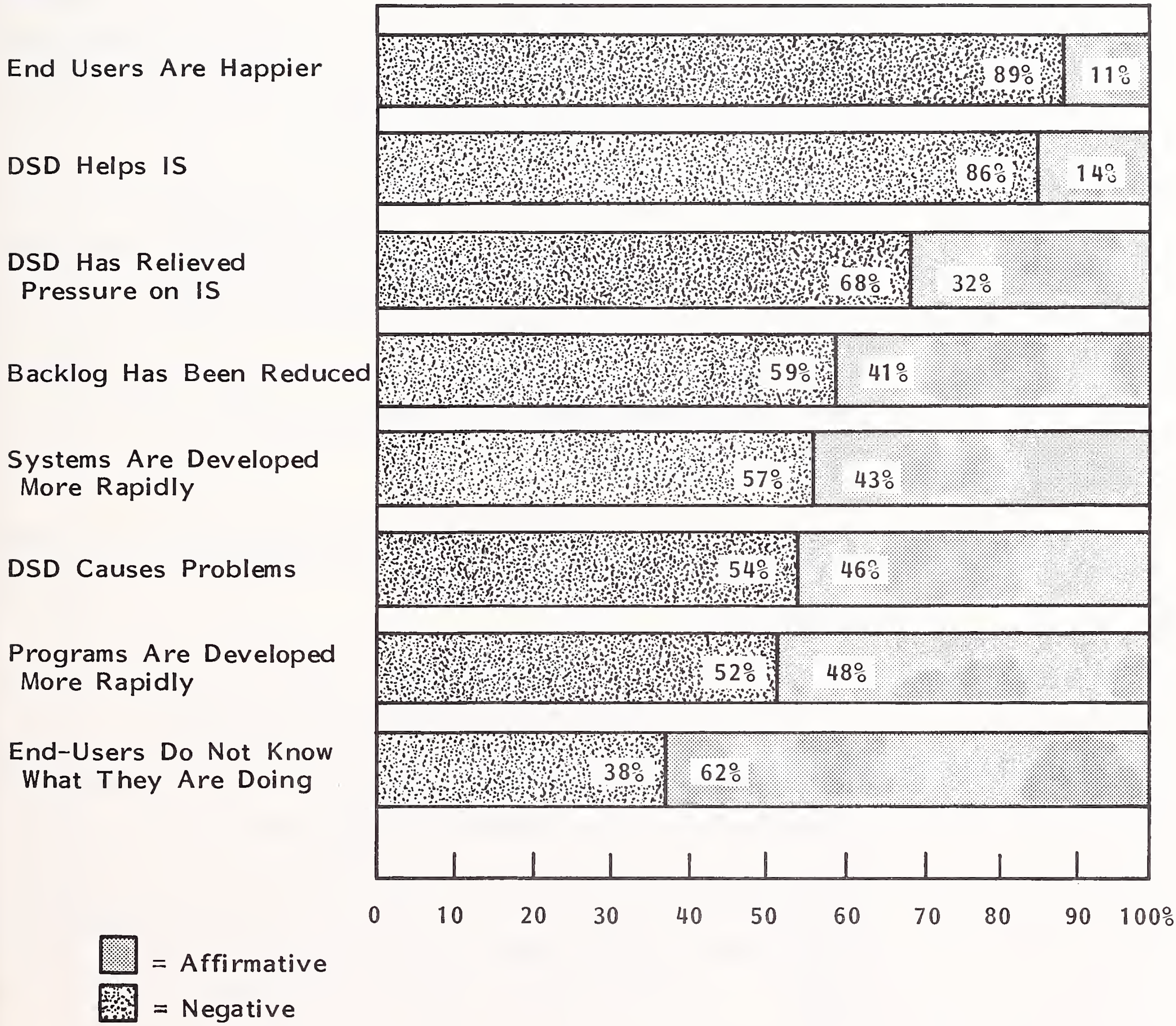
    - Fifty-two percent felt programs got written faster in a DSD environment, but 48% disagreed.

    - Against these rough measures of productivity no conclusion can be drawn concerning the impact of DSD on systems development productivity.

  - Only a slight majority of the IS respondent (54%) felt that DSD would create problems.

EXHIBIT III-9

GENERAL IMPACTS OF DSD



- Users were told that past research had revealed that maintenance was the most costly part of the systems life cycle, and were then asked what impact they thought DSD would have on maintenance. The responses are categorized in Exhibit III-10.

    - Nearly half of the respondents (14) felt that software maintenance costs would increase. The reasons for this opinion centered around poor documentation, and around user involvement that would add cost even if hidden in the user's departmental budget.

    - Another seven respondents stated that they really had not given it very much thought and really didn't know (nor care?) what the impact would be. (One of the reasons commitment to quality is so important is that problems are easiest to correct early in the systems life cycle and become more expensive during the maintenance phase--not thinking about maintenance has been a substantial contributor to the software productivity problem.)

    - Only four respondents stated maintenance costs would decrease, and only one stated this decrease in cost would be because of improved system quality.

    - Two respondents stated that vendors would be responsible for any increased maintenance costs. This is somewhat difficult to understand unless it means that applications packages from vendors would have to be integrated into a new technological environment. (It is a fact that in the changing technological environment both hardware and software have contributed substantially to the software maintenance burden, but it is inevitable that any vendor increased costs will be passed on to the users.)

EXHIBIT III-10

IMPACT OF DSD ON MAINTENANCE

	NUMBER OF RESPONDENTS
• Maintenance Costs Will Increase	14
- Poor Documentation Under DSD	
- Costs Transferred to User	
- Could be a Mess	
- Nightmare on Distributed Systems	
- More Personnel Required	
• Don't Know	7
- Can't Even Measure Maintenance Costs Now	
- No Idea	
- Haven't Given it Any Thought	
- Good Question	
• Costs Will be Lower	4
- Better Design Will Lower Cost	
- Users Will Be Involved (and Be More Reasonable	
• Costs Will Be Increased For Vendors	2
• No Change	2
• New Applications Rather Than Maintenance	1

- Only one respondent stated that maintenance would decrease because new applications would be written rather than the old ones maintained (and the impact on cost was not indicated). This observation is more in line with the concepts of "iterative systems development" and "eternal systems development" put forth by some of the expert respondents. If systems are never completed you never have to maintain them.

2. INPUT'S ANALYSIS OF DSD IMPACT

- INPUT's analysis of the impact of DSD can best be depicted by the reconstruction of the productivity pyramid presented earlier (see Exhibit III-4). The restructured "pyramid" is presented in Exhibit III-11. This unstable structure seems to be dedicated to the following set of priorities and assumptions:

  - Get end-users involved at any cost--it will keep them busy and improve IS-user relations.

  - Find the "magic bullet" or "Swiss army knife" that will permit some results to be achieved as soon as possible.

  - Recruit or develop personnel effective in the use of the tools.

  - Assume that management will be satisfied with the information produced by DSD and will be willing to pay the cost.

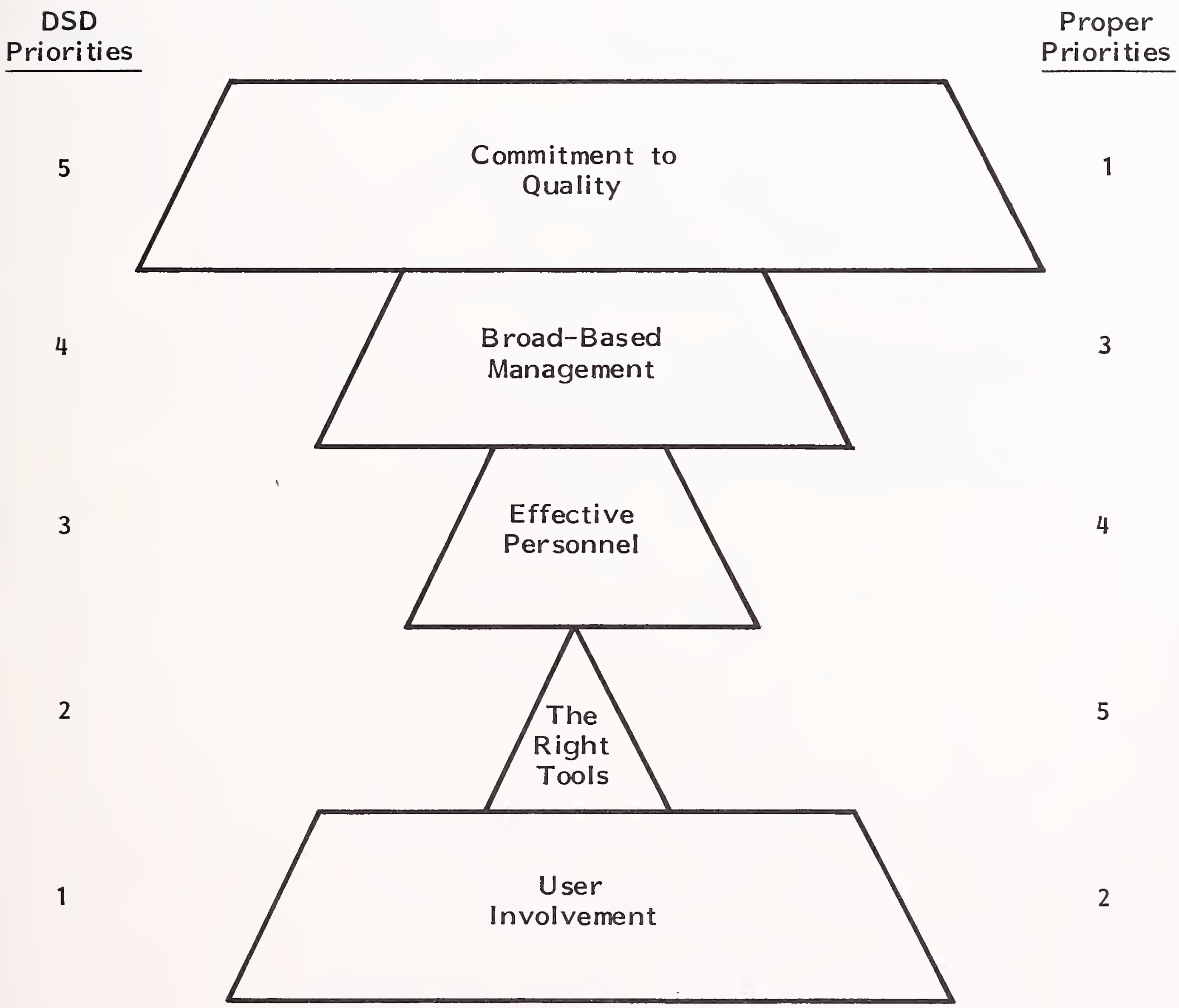
  - Conceptual systems design and data quality will take care of themselves after systems are developed.

- It is INPUT's opinion that productivity must be measured by both cost and systems quality. DSD seems to have the potential for increasing cost and decreasing quality. This combination would mean drastically reduced productivity of the IS function. The DSD potential for increased cost seems to be recognized, but there does not seem to be a similar understanding or concern about quality.

EXHIBIT III-11

MISPLACED PRIORITIES IN THE DSD ENVIRONMENT

(The Restructured Productivity "Pyramid")



- The fundamental difference between the emerging DSD hardware/software environment and the traditional systems development environment is essentially that of process versus discrete manufacturing.

  - There will be more concern with information flow than with data bases.

  - "Programs" will direct and control information flow rather than accept discrete inputs (parts/data) and produce specific outputs (products/documents).

  - All components of the system, including the human being developing and using it, will tend to become more interdependent. (Emerging expert and knowledge-based systems are examples of this.)

- As usual, there is nothing wrong with the DSD concept. In fact, it is probably inevitable. The problems will arise from the turbulence created by conflicting approaches and strategies during implementation.

IV TOOLS, AIDS, AND APPROACHES TO FACILITATE AND CONTROL DSD

IV TOOLS, AIDS, AND APPROACHES TO FACILITATE AND CONTROL DSD

A. PROBLEMS IN THE DSD ENVIRONMENT

- Respondents were asked whether certain potential problems in a micro-mainframe-linked DSD environment were "very serious," "somewhat serious," or "not a problem." The results are presented in Exhibit IV-1.

  - More than one-third of the respondents ranked the following problems "very serious": mainframe performance impact, data base integrity, data security/protection, mainframe capacity planning, data base synchronization, conflicting reports to management, and user understanding of data. These specific problems can be categorized into these serious problem areas:

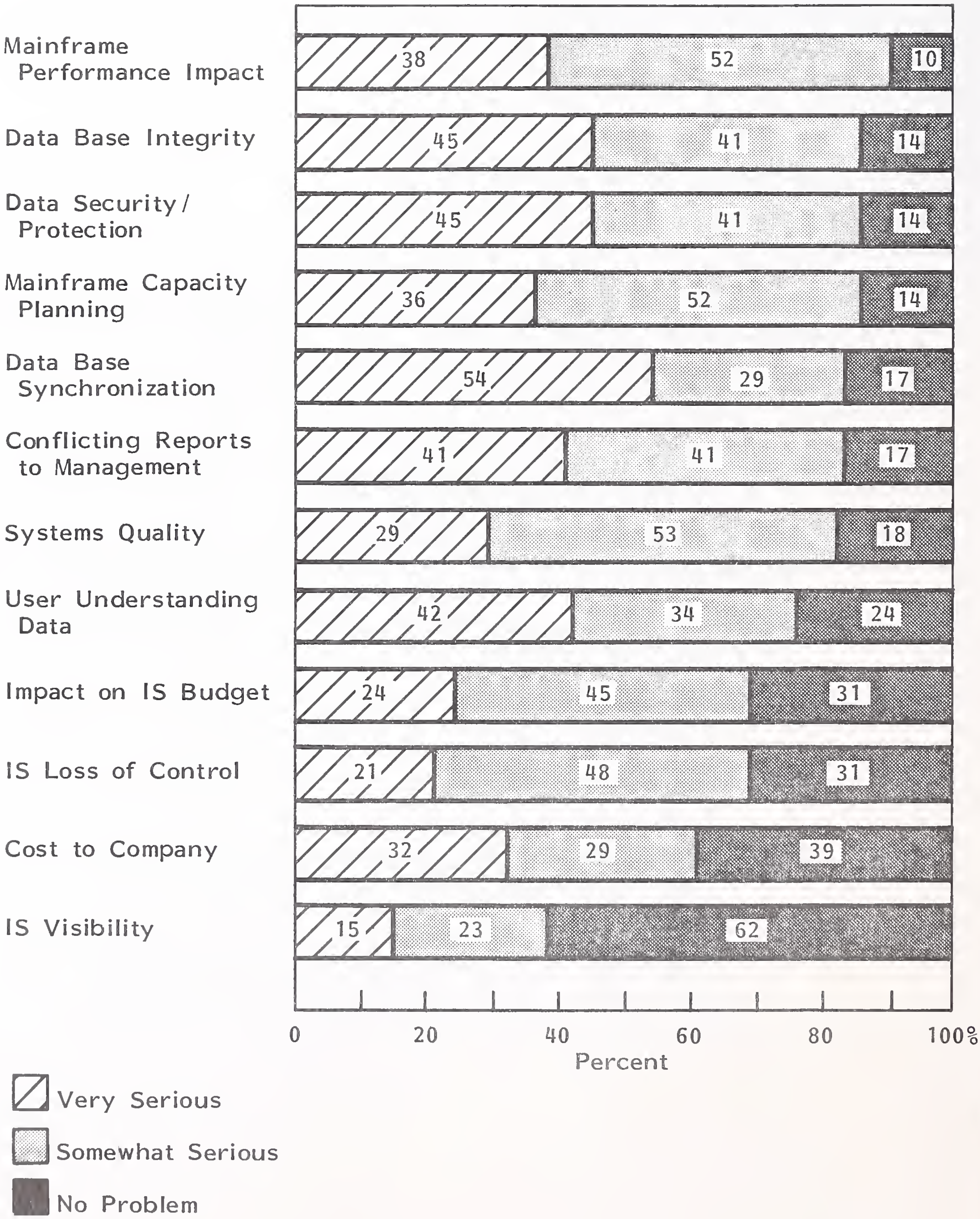
    - Data base management in a distributed development environment.

    - Severe, but unpredictable impact on mainframe performance (probably related to mainframe DBMS burden).

    - Information flow problems (misunderstood data does not convey information, nor do conflicting reports to management).

EXHIBIT IV-1

PROBLEMS IN THE DSD ENVIRONMENT



- It is INPUT's opinion that these are proper areas for concern, but that the magnitude of the problems is not fully understood by the respondents.

- The only category respondents reject as a problem area is IS visibility; 62% state that this is not a problem. This category was included with the thought that having end users be responsible for generating management information might result in a lowering of recognition for the role IS plays in the corporation. Either the question was misinterpreted or IS management feels it already has too much visability.

- Both vendors and experts agreed with the IS respondents in identifying distributed data base management, mainframe performance/capacity, and information flow as very serious problems in a DSD environment. The vendors' concerns about these problem areas closely paralleled the concerns of the IS respondents, but the experts expressed deeper concerns and elaborated on problem areas that were presented. Among the observations made were:

  - "Both internal and external auditors have a lot of problems with prototyping--and they should."

  - "IBM is still trying to sell large mainframes, and micro-mainframe links are going to sell a lot of mainframes."

  - "Problems of security/protection have been talked to death but they have not been addressed on an overall basis."

  - "Most systems analysts and programmers do not have a good understanding of data base management problems--structures, integrity, syncronization, security, and performance impact of various approaches--users certainly aren't going to improve on the situation."

  - "The technical experts aren't expert anymore."

- "If management recognizes conflicting information, the problem might get solved--the real problem is conflicting action based on the conflicting information."

- "Systems quality is going to suffer because you can't separate data problems from systems quality."

- "Information flow, and all of its ramifications, is not understood--period."

- INPUT finds it convenient to think of the problems associated with DSD as being comparable to the interference patterns recorded in holograms, and to think of the solutions (tools, aids, and approaches) to the problems as being the coherent light sources necessary to display the holographic information.

B. THE DSD HOLOGRAPHIC MODEL

- There is currently a great deal of fascination with holographic paradigms of everything from the universe to the human brain. At the risk of being trendy, INPUT feels a holographic model is especially appropriate for describing both problems and solutions in the DSD environment. The following description of the fundamental principle of holograms (by biologist Lyall Watson) should demonstrate why INPUT feels that way.

    - "If you drop a pebble into a pond, it will produce a series of regular waves that travel outward in concentric circles. Drop two identical pebbles into the pond at different points and you will get two sets of similar waves that move towards each other. Where the waves meet, they will interfere. If the crest of one hits the crest of the other, they will work together and produce a reinforced wave of twice the normal height. If the crest of one coincides with the trough of another, they will cancel each other out and produce an isolated patch of calm water. In fact, all possible combinations of the two will occur, and the final result is a complex arrangement of ripples known as an interference pattern.

- Light waves behave in exactly the same way, and a hologram is the extremely complex pattern recorded by two laser beams on a photographic plate (one after being reflected off an object). The hologram itself does not seem to contain any information concerning the object, but a coherent light source reveals a three-dimensional image of the object. Moreover, even a portion of the photographic plate is sufficient to reconstruct the entire image." (This property has some disturbing characteristics for security in a distributed data base environment.)

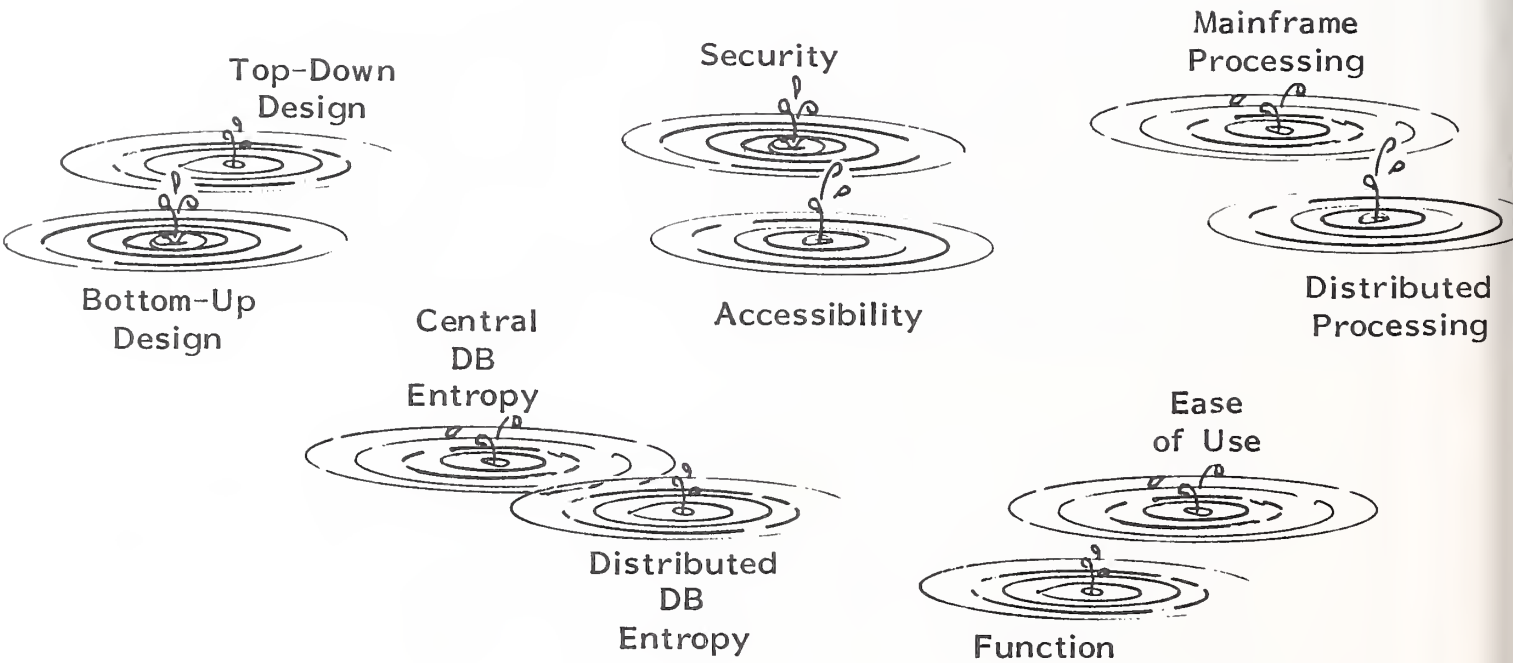
- There are a lot of identical pebbles being dropped in the information systems pond in a DSD environment, and the result is going to be some exceptionally complex interference patterns. Exhibit IV-2 depicts a few of the matched pairs of pebbles, and a major source of turbulence represented by conflicting management reports.

    - The potential problems of top-down and bottom-up design have been mentioned previously. Ideally, if the crests match, superior systems will emerge. On the other hand, if crest and trough meet they may cancel each other out and no workable system will emerge. (This results in negative productivity since the system must be started over from scratch.) The most likely result will be an interference pattern that will require a source of coherence if quality information is to be reconstructed.

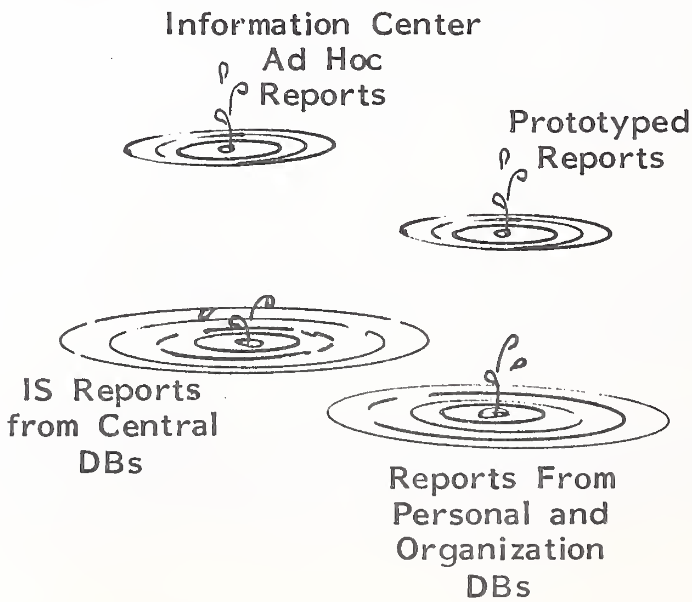
    - Providing easy accessibility and establishing a secure central data base represents a classic case of interference between objectives—and if

EXHIBIT IV-2

POTENTIAL DSD HOLOGRAMS
(Interference Patterns)



CONFLICTING MANAGEMENT REPORTS



DSD does nothing else it will certainly force IS management to face these problems. Unfortunately, security problems that might have been solved for a "data bank" are not so easily solved in a network environment. The problem becomes something of a paradox.

- The nature of the problem relates to the fact that data open to legitimate query (or transfer to personal data bases) may reveal other information when subjected to statistical (or other) analysis.

- In other words, a "meaningless" interference pattern of data from various sources may reveal information that should be protected if someone is smart enough to develop the proper "decoder."

- Computer scientists are still wrestling with theoretical solutions to this problem.

- Speaking of paradoxes, the ease-of-use and added-function problem that goes back to mainframe operating systems days is currently being repeated with integrated software packages for micros. (For those wondering why the recently announced IBM PC AT can have up to three megabytes of RAM, just wait until IBM releases its long-awaited multi-user operating system.) Something simple linked to something complex can lose its user friendliness in a hurry.

- INPUT issued an "Executive Bulletin" (Micro-Mainframe Links, The Challenge), and "Supplement" (Understanding Entropy in Micro-Mainframe Networks) on the potential, high entropy of both data and information in the micro-mainframe environment. The natural tendency to disorder of data in the DSD environment increases exponentially.

- This means more processing power at all levels merely to maintain quality. According to information theory, the communication of data from one location to another is also subject to entropy. Since publishing the above Executive Bulletins, INPUT has discovered that IBM has become aware of data entropy and can actually measure it at various levels (from heat in the machine room down to gate level).

- Balancing of processing between mainframes and micros is going to be a nontrivial problem, and processing suitable for a micro may bring a mainframe to its knees. The example INPUT has used repeatedly is a JOIN of relational tables in a personal data base, as opposed to a JOIN initiated from the intelligent workstation against large relational tables on mainframes. Processing performance problems can be transmitted in either directions over micro mainframe links. The resulting interference pattern could result in both micro and mainframe throwing so much work at each other that neither could get anything done (shades of the virtual storage thrashing problem).

- The interference patterns for management, when receiving "information" from multiple sources (ad hoc reports, reports from personal data bases, systems being prototyped, etc.), could prove to be disastrous for the decision-making process. Some of the experts referred to conflicting management reports as a potential nightmare. It is important to recognize that "productivity" of all of the information sources may be "improved" if productivity is measured by the ability to throw pebbles into the pond. Unfortunately, management may be confused by the ripples, and the corporate row boat may sink under the burden (expense). Software productivity must be related to quality, not quantity.

• Based on the DSD problems identified by the research for this report and probable interference patterns associated with distributed data bases, distributed processing, and information flow, it seems imperative to concentrate on restoring commitment to quality at the base of the productivity pyramid. Tools, aids, and approaches to assure and control quality are essential in the DSD environment if software productivity is to be maintained--much less improved.

C. A COHERENT STRATEGY FOR MAINTAINING QUALITY

- The trend toward distributed systems development encompasses a number of other strategy trends, as shown in Exhibit IV-3. Recognition of these trends is necessary in order to identify the tools, aids, and approaches needed to implement a coherent strategy to maintain systems quality while improving productivity in implementing systems.

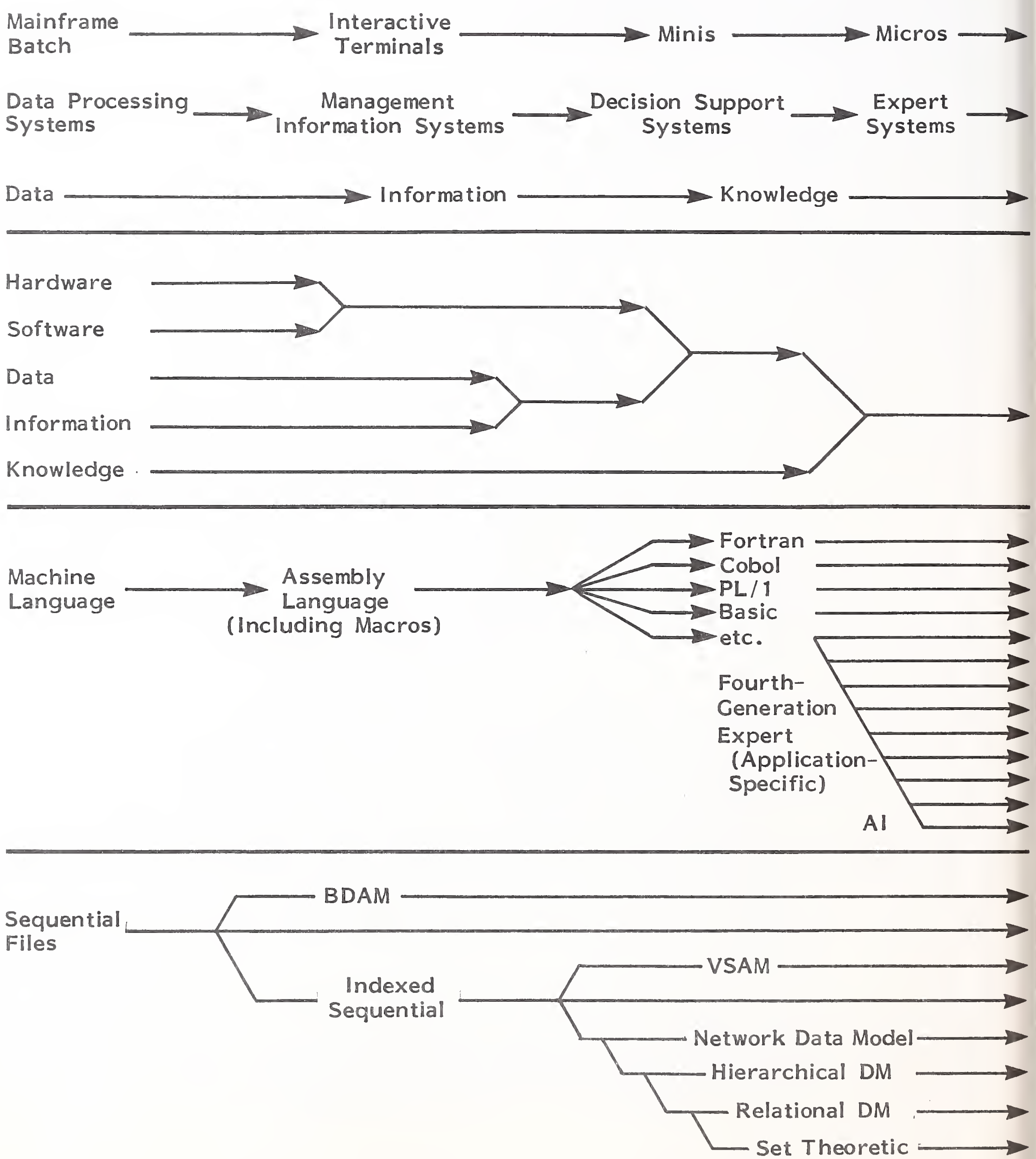
  - The general hardware trend--from batch-oriented mainframes, to interactive timesharing systems, to distributed processing on minicomputers, to the current micro-mainframe links--is quite clear. All possible hardware arrangements are still possible and appropriate in today's environment (batch processing and minicomputers are not dead).

  - The trend (from data processing systems, to management information systems, to decision support systems, and now, to expert systems) represents more of a change of terminology than of substance (although distinctions can be made). There is disturbing evidence that each new "concept" merely promises to deliver what was promised by the previous solution. As one expert stated when asked about expert systems: "(this is) one more attempt to divert attention from the problem (poor conceptual systems design)."

  - Data, information, and knowledge are becoming combined in computer/communications networks and this is important. From a systems

EXHIBIT IV-3

STRATEGIC TRENDS



point of view, it is convenient to think of data as being encoded; information as including paper documents, audio-visual recordings on other media (magnetic or optical), books, periodicals, etc.; and knowledge as more highly organized information (and data) supporting what is known about a particular subject.

- There is a trend toward increasing integration of hardware, software, data, information, and knowledge.

    - Hardware and systems software have become inseparable in most users' minds. (Hardware, from IBM 3084s down to PCs, is seldom sold without an appropriate operating system.)

    - We are slowly beginning to learn that applications software is of little use unless necessary data are available. (It is also true that enormous central data bases are of little value unless they can be readily and economically accessed.)

    - An expert system must include both the knowledge base and the necessary algorithms (programs) or neither is of any value. When interacting with the expert, all three become interdependent and each causes changes in the other (hardware/software/experts).

- Languages are proliferating and will continue to do so. In addition, dialects and "slang" develop in individual languages. This will continue despite the best intentions of various standardization efforts.

- The quest for a single data model is also doomed because even set theoretic principles break down under the strange information views required in personal filing systems.

- With the major trends as environmental assumptions, we have effectively ruled out the simplest solution to many of the problems--standards. This is not to say that when IS management is firmly in control, standards should not be applied--we are merely assuming that the trend to DSD (combined with the strategic trends) poses a significant threat to standards efforts. (In fact, the one traditional standard of the "IBM solution" is no longer available since IBM seems intent on tossing as many pebbles in the DSD pond as possible--more on that later.) Therefore, we must bring a coherent strategy to the holographic interference patterns of the DSD environment without the benefit of not permitting the first stones to be thrown.

I. TOP-DOWN VERSUS BOTTOM-UP DESIGN

- The interference pattern associated with top-down versus bottom-up design presents problems of program integration, data availability (and understanding), and performance of the integrated system. It is assumed that the top-down designer (IS department) will be responsible for the integration. In order to do that there must be understanding of the following:

  - The data required and how it will be used (processed).

  - Newly generated information and data and their distribution.

  - The potential performance impact on mainframes and communications networks.

- The most important approach to be taken in facilitating this understanding is to be sure that bottom-up "design" does not consist merely of screen painting. (As one respondent stated, "workable systems are not developed by artists.") A data flow prototype should be required and this in turn should be used to project a cost estimate for the data. Simply stated:

  - Get the data base administration function involved.

- Walk through the data flow prototype--early! (Gopal Kapur, one of the experts we talked with, passes out signs stating "In God we trust--everything else, we walk through." It's sound advice but is seldom taken.)

- Tools and aids needed to support this approach and reconstruct the top-down versus bottom-up interference pattern are as follows:

  - Data dictionaries, glossaries, and encyclopedias that would have the following characteristics.

    - Dictionaries would come in a variety of forms: abridged/unabridged, pocket, and specialized.

    - Glossaries would support particular projects in data flow prototyping and should permit comparison against appropriate dictionaries.

    - Data encyclopedias would describe major data flows within the organization.

  - A meta language (common language), and appropriate translations for the emerging Tower of Babel, which would serve the following purposes:

    - Facilitate integration with central systems and data structures.

    - Facilitate walk-throughs and understanding.

    - Serve to improve performance provided the right language is developed (or selected) and translators are properly positioned in the network.

- Data flow performance monitors and estimators with the following characteristics:

  - Estimates of processing requirements against various data models and structures, based on volume and the algebra and calculus being applied.

  - The ability to secure and reject "unreasonable requests" (or at least signal high cost before proceeding).

  - Determine and establish limits of transcomputability (an algorithm is transcomputable if its computational cost exceeds all bounds that govern the physical implementation of algorithms--in other words, you cannot build a machine to process the solution) of analysis tools.

  - Provide automatic scheduling and performance tuning.

2. SECURITY VERSUS ACCESSIBILITY

- The assumption that data are available for systems developed in a DSD environment presents one set of problems with which we are all familiar. The problem of access to available data is another question, and the interference pattern created by accessibility and security can be extremely complex and disruptive. Unreasonable security measures can be used as a means of denying access for legitimate reasons, and authorized access can compromise security.

- The whole security-protection issue has been talked to death and very little has been down in most companies--it is complex work. INPUT believes the DSD environment will force attention to be focused on the problem. As one respondent stated: "DSD only complicates a problem that has not been solved--perhaps this will at least force some understanding of how serious it really is." At the very least, security problems associated with access control, information flow control, and certification should be understood; classifications, authorization, and procedures should be reviewed on a continuing bases. At the very least:

- All systems designers and authorized users should be trained in both the handling of classified information, and the problems associated with recruiting of shared systems.

- A true security assessment should be made of central systems, and users should be informed of any security risks.

- Users should be continually reminded when they are dealing with classified data, and impressed with their responsibilities for its protection.

- Usage statistics on classified data should be recorded and users should be reminded that their usage is being monitored.

- Measures that would be helpful include the following:

  - Incorporate audit trails of secured data usage in the data flow analysis programs that will also monitor performance (mentioned above).

  - Provide automative security prompts in data base systems.

  - Provide security information in data documentation (dictionaries, glossaries, etc.).

  - Provide encryption protection for removable storage media, with keys secure to the specific system. (In other words removable disks can be tied back to the specific processor and/or program.)

3. EASE OF USE VERSUS FUNCTION

- It is extremely difficult to determine the tradeoffs between ease of use and added function in operating systems, languages, and data base systems. This difficulty is compounded by the various skill levels and functional requirements of both individual users and classes of users. What is easy to use for one person is too complex for another, and a level of functionality which is satisfactory for one user set is inadequate for another. However, the general tendency has been to add function at the expense of ease of use, and this tendency is already apparent in the DSD environment.

  - Personal computer users will find it necessary to interface with multiple operations environments as integrated packages, "windowing," and micro-mainframe links become available.

  - UNIX may be touted as a solution from mainframe to desktop, but it will be subject to the ease-of-use versus function interference pattern.

    - Although UNIX may be considered easy to use by a generation of minicomputer users (especially compared to MVS), it will prove complicated for those accustomed to personal computer operating systems.

    - In addition, as UNIX is extended to cover the DSD environment from mainframe to desktop, functions must be added to satisfy those accustomed to more comprehensive environments and to address very real weaknesses such as protection and security. This can only result in decreased ease of use.

  - Fourth-generation languages are being extended in order to facilitate the development of major applications, and are becoming associated with more complex data structures. The question of whether such languages should be designed for end users or programmers has already begun to surface, and efforts to accommodate both user sets will follow a predictable path and satisfy neither.

- Multiple data models have been endorsed by IBM with its announcement of DB2 for mainframes along with appropriate extract programs for building relational tables from IMS data bases, VSAM data structures, and sequential files. Experienced systems people are already beginning to ask questions about the relational model ("Is it for real or just a solution looking for a problem?" was a question INPUT received), and a book on data base systems for personal computer users has referred to relational systems as being complicated. This illustrates several aspects of the ease of use versus function problem.

  . Giving additional choices complicates the decision-making processing for even experienced systems personnel.

  . Although the inventor of the relational model considers it to be a productivity aid (Dr. Codd received the Turing Award for his paper that emphasized the relational model as a productivity aid), it is still complex for certain user sets.

  . Familiarity with a particular data structure, language, or operating system creates resistance to change regardless of the potential advantages of the new offering.

- The quest for ease of use can also obscure complex functions with the result that users literally do not understand what they are doing. The fascination with the ability to generate different views of data in a cosmetically appealing format all too frequently leads to ignorance of the underlying logic used for data selection and the algorithms used for processing these data. Indeed, many users would not understand the significance of these functions if they were presented in appropriate algebraic statements (or if these statements were described in English). In a DSD environment this can lead to unfortunate consequences.

- Different spreadsheet packages can produce different results without operators (or analysts) being aware of the significance.

- Inappropriate statistical techniques may be readily applied by those who do not understand them.

- Statistical techniques may be applied appropriately against inappropriate data.

- Errors in productivity tools and aids may go undetected by users who do not understand what they are really doing.

- In other words, complex problems are not necessarily simplified by making systems user friendly. The result can be inaccurate information in the decision-making process.

• Emerging expert systems recognize the requirement of explaining inferences made by the system. It is accepted that, even after substantial detailed analysis by "knowledge engineers," the expert must be able to understand what the system is doing and be able to override and correct it. The current DSD environment does not encourage such discipline because of the emphasis on results rather than quality.

  - Automatic mapping and documentation of transformations in structure and content which occur when host data bases are distributed for purposes of prototyping, departmental information centers, and person (or organizational) data bases should be provided from the host system.

  - Test data should be generated on the host and distributed on request from the central facility. The test data themselves should be documented to illustrate the quality control exercised over the host data base and to further define the characteristics of the data being distributed.

- Cautions on use of the data should be provided as appropriate, and protection and security options should be available from the central facility.

- A document control system should make available a list of all reports currently employing the distributed data and require registration of all documents generated and distributed using the data.

4. CENTRAL VERSUS DISTRIBUTED DATA BASE ENTROPY

- It is beyond the scope of this study to describe in detail the potential problems of data base entropy. Readers of this report are referred to INPUT's previous executive bulletins for an explanation of our general concerns on the subject. IS management's concerns about data base integrity and synchronization, and about user understanding of data and information are all intuitive expressions of the general problem. Two examples will be given to illustrate the potential interference patterns of central versus distributed data base problems.

  - Practically every data base administrator has experienced the problems of reorganizing very large data bases. Numerous examples have occurred of IMS data bases that grew to the point where the system would not stay up long enough to reorganize the file (or at least a solid block of computer time was not available on the system). This is an example of the relationship of entropy to data base size. The larger the data base, the greater its tendency to become disorganized (higher entropy) and the more energy (computer power) is required to maintain data quality.

  - Data structures can also influence entropy. The more ways data can be rearranged (the more flexible the structure), the higher the entropy will be. The general understanding that relational data base systems should not be used for very large data bases because of performance problems merely illustrates that the entropy of relational data bases is higher than an alternate structure (hierarchical) and shows that more energy is required to maintain order.

- These two simple facts give focus to some problems:

  - How big do you let central data bases become before the central processor runs out of power?

  - A prototype system developed using a relational DBMS on a personal computer may become impractical when it becomes operational on the large mainframe because of performance problems when used against corporate data bases that have been converted to relational tables using DB2.

- All of this is complicated by the fact that entropy comes into play whenever information is transferred from one location (or human being) to another. It is essential that the entropy associated with the DSD environment be understood--the current body of knowledge concerning information theory is a good place to start.

- The interference patterns associated with centralized versus distributed data bases can lead to unmanageable levels of entrophy on the information network unless proper distribution of data bases (and their backups) are made among the various network models, and the proper data models are achieved. Tools and aids needed are as follows:

  - Tools to support data base prototyping, data flow, data documentation, and performance measurement and estimation have already been emphasized in this section, and they all apply to problems of entropy.

  - Specific tools to measure and predict entropy of both data bases and information flow are required in order to facilitate the development of the more general tools described earlier. It is beyond the scope of this study to define such measurement and prediction tools in detail--indeed, fairly sophisticated models of specific data bases and information requirements may be necessary before general-purpose measurement tools can be developed. However, it is anticipated that hardware tools in terms of data base processors will be required in order to contain the entropy problem until it is better understood.

5. MAINFRAME PROCESSING VERSUS DISTRIBUTED PROCESSING

- If entropy of data and information are understood, a significant portion of the distributed processing problem will be solved since it is anticipated that most processing power expended is merely a measurement of data entropy. However, it is also anticipated that analysis tools will become increasingly complex as decision support systems evolve with expert systems. Decision support models developed on a prototype basis (especially using tools of operations research and artificial intelligence) have a tendency to grow exponentially in their processing requirements with only linear increases in basic parameters (the traveling salesman problem from operations research is frequently cited). It becomes possible for relatively simple decision support models to exceed the processing requirements of even supercomputers.

- Knowledge of the limits of current tools and aids for modeling is obviously necessary. In addition:

  - Improved algorithms and approximation techniques are going to be required.

  - Specialized processors, unburdened by general-purpose operating systems overhead, will be required. Models (programs) will be directed toward appropriate processors on the network based on these processing requirements--as parameters of the model change.

- In order to make this reasonably automatic tools to analyze tools are necessary. (Performance measurement and prediction tools against programs as well as against data.)

6. CONFLICTING MANAGEMENT INFORMATION

- Improving productivity of both systems analysts and end users in generating information (reports) from both corporate and distributed data bases is going to contribute substantially to the emerging problem of "information overload." However, the most important impact on information quality is going to be the interference pattern created by conflicting or misunderstood management reports. The problems associated with understanding become increasingly difficult as data are distributed and transformed outward (and upward) from their sources. In other words, what is hopefully understood by the data base administrator becomes difficult for the systems analyst/programmer, extremely difficult for the end-user analyst/report generator, and virtually impossible for the corporate executive. (The point can also be made that the executive could have a clear understanding of his information requirements and these become distorted as they descend on the data base administrator.)

- Even with well-established internal audit procedures, corporate executives in today's environment must make decisions based on trust (and perhaps prayers) that their information sources are accurate and that their understanding of the information is correct. The DSD management information hologram created by multiple information sources (see Exhibit IV-2) has the potential for obfuscation at best and deliberate misrepresentation and fraud in its extreme forms.

- In responding to the perceived problem of conflicting management reports, both vendors and experts emphasize management understanding but their reactions varied considerably.

- The "big bang theory" of corporate data base development (and control) is still put forward by those who do not fully understand the problems associated with the DSD environment (or at least feel that strong central control can be maintained). Essentially, it offers the ultimate panacea of determining once and for all the corporate data requirements, establishing the corporate data base, and everyone living in a perfect state of harmony and understanding ever after. It is also recommended that the tremendous investment in hardware, software, and human effort to establish the ultimate data base be capitalized so that it can be spread equitably over time and across the broad user base it anticipates. INPUT does not believe such an approach is either appropriate or realistic for the DSD environment.

- A step below the "big bang theory" there are those who recognize the problems associated with information flow but feel that relational data bases and fourth-generation languages will solve these problems. INPUT's opinion, as represented by the emphasis of this report, is that the current tools of DSD have the potential for creating a new set of problems that will be far more complicated than in the past.

- Then there are those who recognize the problem, but insist that management is too smart to be confused by the new environment. Essentially, the theory is that management has traditionally ignored most of the more advanced decision support systems outputs anyhow, so a few more information sources will not make any difference. While INPUT agrees that most critical management decisions ultimately get down to judgment and/or intuition, it is felt that this laissez faire attitude has a serious deficiency and is extremely dangerous. The complex management information interference pattern in the holographic model of DSD implies that fundamental information sources (data bases) may be contaminated, and the growing concern about audit trails is probably justified.

- Finally, there are those who have a sufficient understanding of the management information problems to suggest the identification of "survival information" that is absolutely essential to run the business. Once identified, the flow of this essential information (and supporting data) through the organization would be controlled and its quality monitored. For the benefit of those who think this is a simple task, please realize that information in this sense includes much more than computer-stored-and-generated information. It includes all essential communications whether paper, electronic, voice, or face-to-face meetings. It is INPUT's position that such general information management has been virtually ignored in the rush to apply computer technology to portions of the management process. Once again, it becomes a question of information flow (or process) versus information processing (or a data processing application).

- Many of the tools and aids to facilitate control of the interference patterns associated with the holographic DSD information systems model are essential to establishing and controlling "survival information" flow. However, there is also a need for a document control system that goes beyond those currently employed. Among the features would be:

    - Treatment of all critical documents in a standard manner regardless of media, format, or storage facility. In other words, a critical document might be a single screen of information displayed on demand in the president's office, archival documents on microfiche, or paper documents stored in a public (secure) warehouse.

    - Automatic reconciliation facilities (or warnings) as required. For example, if changes in information derivation occurred, such changes could be automatically noted on the next document (or a warning would be displayed). Copies of archival paper documents provided through the system would be accompanied by necessary explanatory notes produced by the system.

- Central security, access control, and usage monitoring. Whether the document (or display) was created from a corporate computer data base or was removed from a file cabinet, it should be processed through the system for purposes of control. In other words, electronic and paper security systems should be integrated.

- Such an integrated document storage system is essential for implementing integrated office systems that will take advantage of new technology such as optical memories (which can contain images of paper documents, voice documents, and video information). The impetus for such a document system should come from the survival information required by the corporation. Survival information will become the coherent information source that can reconstruct information out of the interference patterns of conflicting management reports.

D. USER PERCEIVED NEEDS FOR FACILITATING AND CONTROLLING THE DSD ENVIRONMENT

- INPUT developed the holographic model of the DSD environment and established the approaches for controlling the interference patterns that were considered important, because research disclosed that:

  - Although users were sensitive to the potential problems associated with the DSD environment, and were proceeding through such an environment quite rapidly, there was little currently being done (or planned) to contain or solve problems. There was little awareness or thought being given to the need for tools to facilitate and control the development of the DSD environment.

- Exhibits IV-4 to IV-7 summarize the reactions of IS department when questioned about what they are currently doing about the problems associated with the DSD environment.

    - Twenty-three percent stated they were developing security procedures and/or systems. (However, most of these did not seem adequate to control today's central data base environment, much less the distributed data base environment that seems inevitable as DSD develops.)

    - Seventeen percent stated that micro-mainframe links were "no problem yet"; therefore, these respondents were doing nothing.

    - Ten percent stated they were doing "nothing."

    - The remaining 24% were taking specific but somewhat vague actions, such as: standardizing the environment and data base administration, tightening controls, controlling micro use, archiving IMS data, etc.

- When asked about the tools needed to facilitate DSD, respondents reacted in the manner shown in Exhibit IV-5, which suggests that the movement toward DSD was probably being led by the users (or by some separate organization). For example:

    - Twenty-three percent did not respond to the question at all.

    - Seventeen percent stated "education" was the primary tool required.

    - Seventeen percent placed emphasis upon some aspect of networking (Micro-mainframe links, micro-micro communications, protocol converters, communication expertise).

    - Thirteen percent frankly stated they did not know what was needed or that they had not given it any thought.

EXHIBIT IV-4

POTENTIAL MICRO-MAINFRAME PROBLEMS

What are you doing about potential micro-mainframe problems?

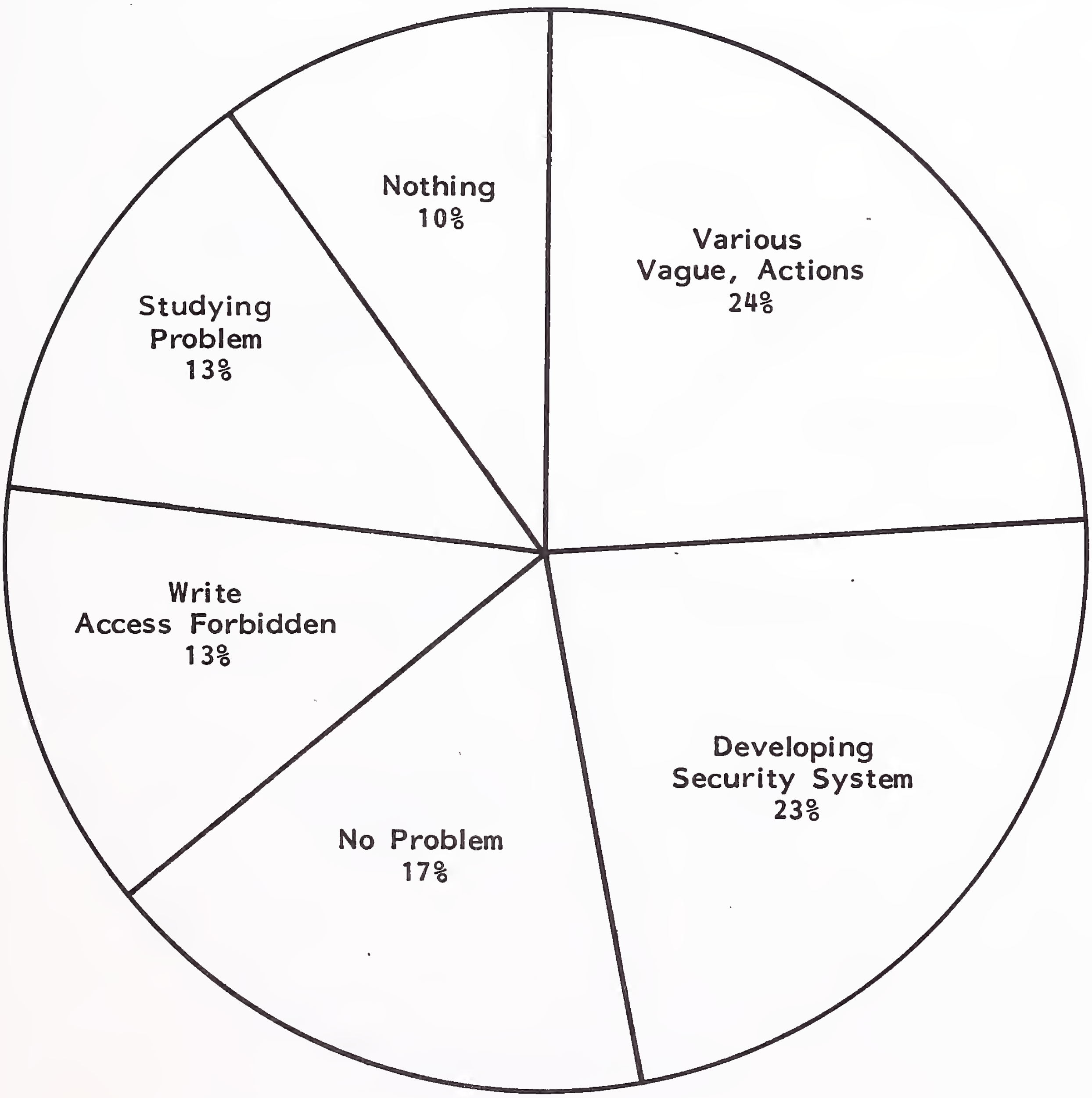


EXHIBIT IV-5

DSD TOOLS

What tools are needed to facilitate DSD?

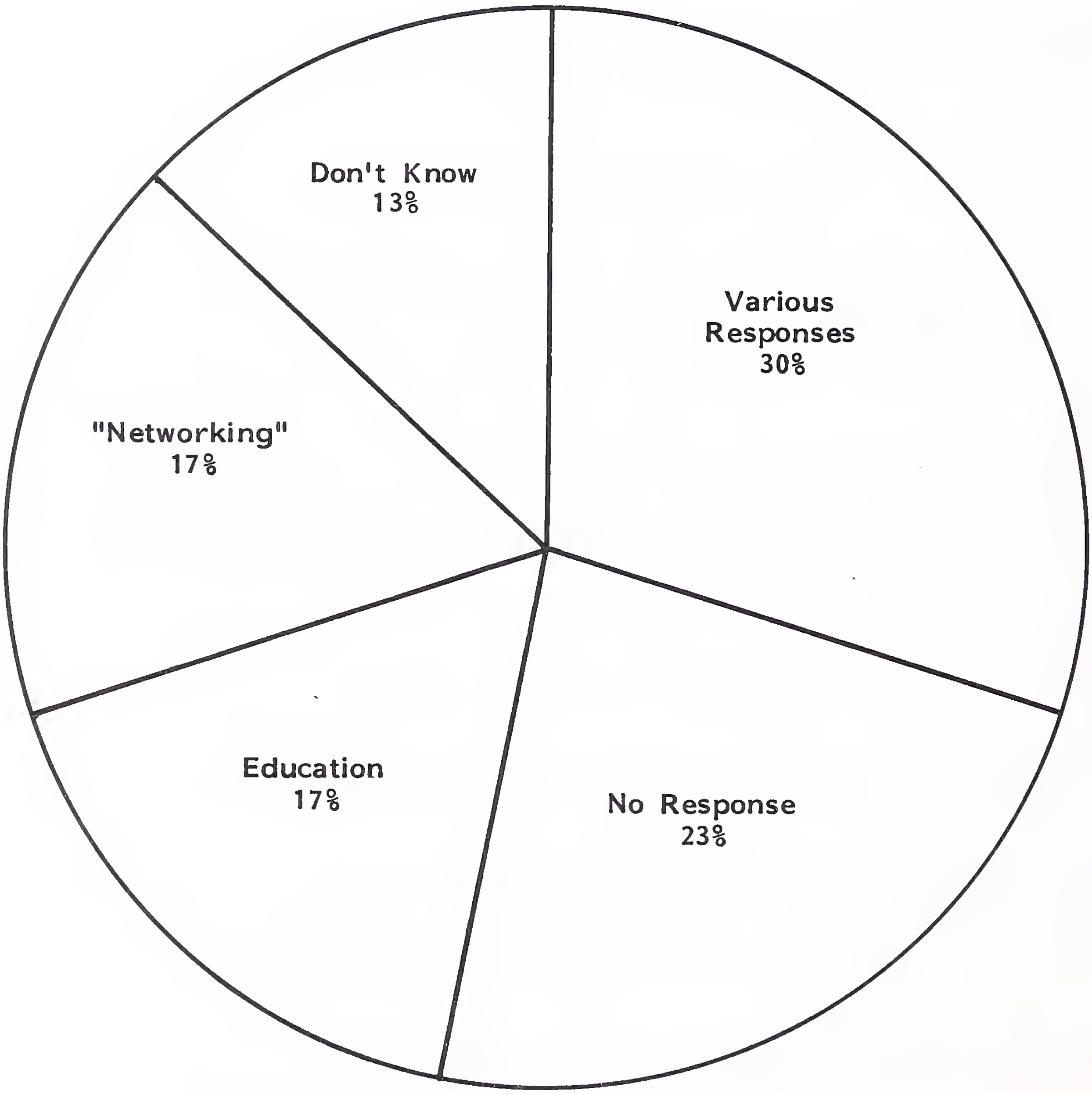


EXHIBIT IV-6

DSD CONTROLS

What tools are needed to control DSD?

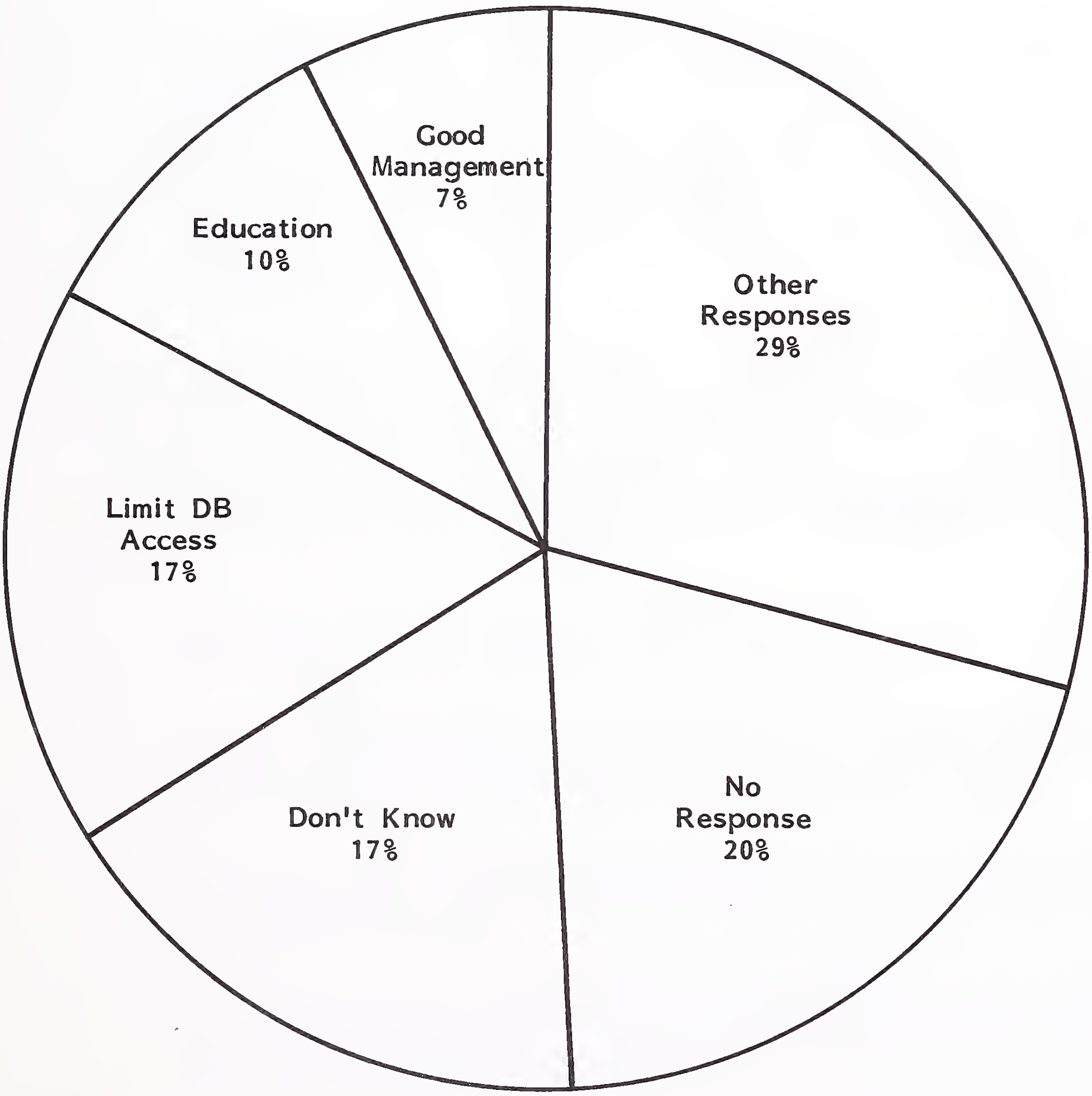
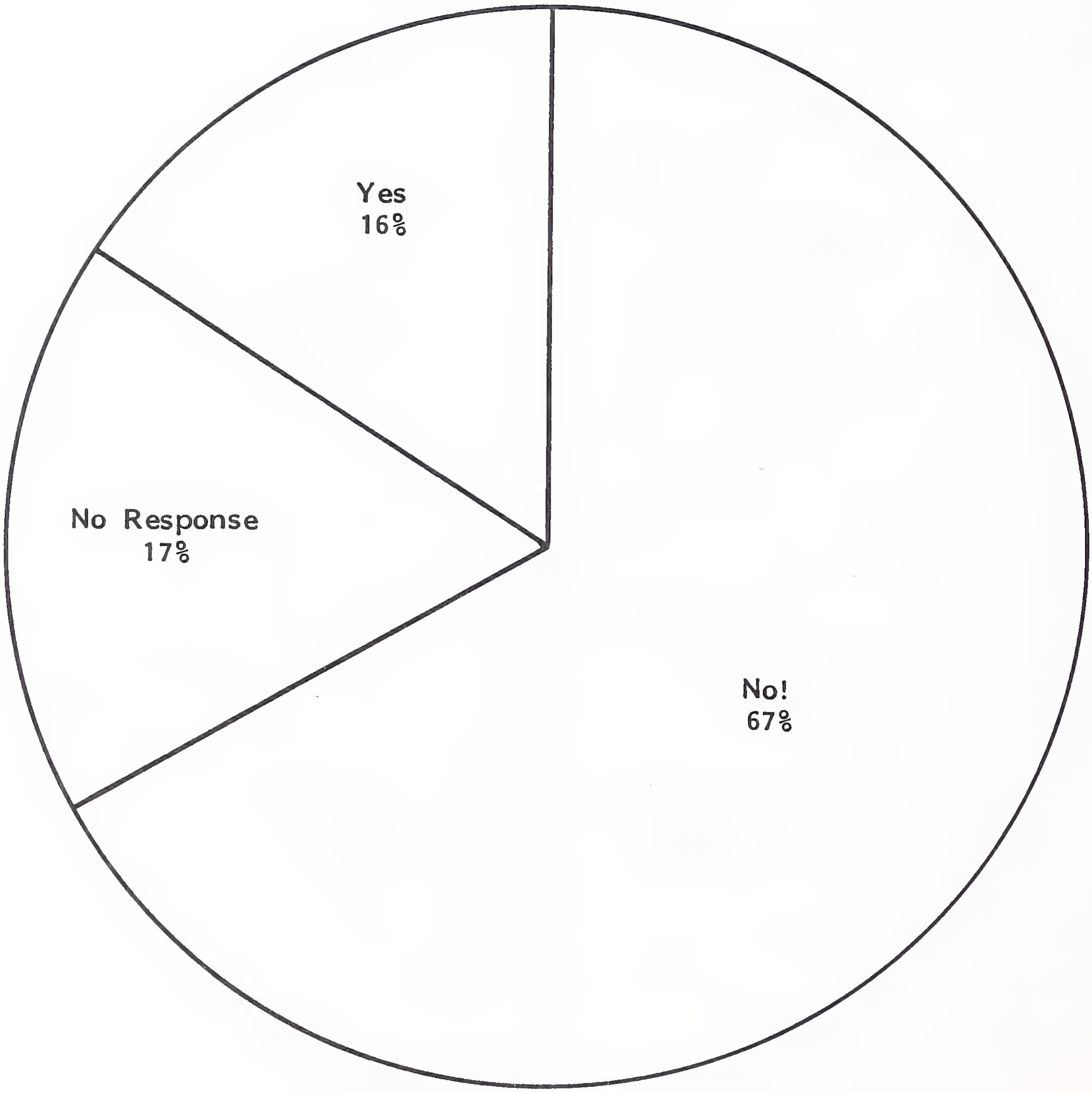


EXHIBIT IV-7

NEW DSD TOOLS

Have you heard of promising new tools to facilitate or control DSD?



- The remaining 30% mentioned a variety of tools or actions that they felt were "needed" to facilitate DSD. Among the items felt to be needed were the following: a program generator, better prototyping, an appropriate DBMS, a relational DBMS, decision support software, more people (evidently to user organizations), SAS and SAS graphics, RACF (IBM), and one respondent simply stated that it was necessary for IS to "maintain control."

- The responses are not surprising; the trend toward DSD is less than specific in its implementation. It is less a specific action than a process, and the process may be so gradual that it happens without any specific thought about what is needed to make it occur, or it may be so rapid that it doesn't give any time for thought about what is "needed" to facilitate it.

• The question concerning the tools needed to control DSD was intentionally asked after asking the questions concerning the potential problems that might be associated with DSD. Nevertheless, over one-third of the respondents shown in Exhibit IV-6 did not have any comments on the tools that might be needed.

- Twenty percent did not respond at all.

- Seventeen percent stated that they could not think of any "unknown" or said that no thought had been given to the need for such tools.

- Seventeen percent vaguely mentioned security and/or limiting access to corporate data bases.

- Ten percent felt that education of end users was the primary way to control the development of DSD.

- Seven percent stated that "good management" would solve any problems.

- The remaining 29% gave the following suggestions for controlling the DSD environment: standards, monitor data (transmitted) to PC, budgets, data dictionary, capacity planning, structured project management, accounting packages (JARS), a tool for data integrity (back-up), and one respondent who stated it was "a good question--no one seems worry about tools for control."

- Once again, the process of distributing systems development to end users appears to be proceeding along a path of its own with impetus from end users and vendors, and with little direction (control) from IS management.

- When asked whether they had heard of any new tools to facilitate or control DSD, there was a resounding negative response, as shown in Exhibit IV-7.

  - Sixty-seven percent said "no" or that they had not bothered to look for new tools.

  - Seventeen percent did not respond at all.

  - The remaining sixteen percent mentioned: RACF (IBM), DB1 (presumably IBM's large mainframe version), Golden Gate (Cullinet), Montis-ITS (Cincom Systems), and an unspecified education package for information centers.

- In the search for productivity improvement, certain tools and aids (hardware and software) have been applied, and the result has been the distribution of some systems development responsibilities to end users (the DSD environment). It is INPUT's opinion that the DSD environment, as described in the holographic model, poses serious threats to information systems quality that may more than offset any advantages gained through improved productivity in the development of specific applications. If quality is to be maintained in the DSD environment, tools and aids to establish, monitor, and control information flow are required. These requirements will now be examined against the tools and aids currently being employed in the DSD environment.

V ANALYSIS OF CURRENT TOOLS, AIDS, AND APPROACHES

V ANALYSIS OF CURRENT TOOLS, AIDS, AND APPROACHES

A. THE QUEST FOR THE MAGIC BULLET

- There is no shortage of tools, aids, and approaches to improving productivity. All you have to do is pick up any computer publication and solutions to productivity problems will practically overwhelm you. Some solutions even work pretty much as advertised--the difficulty is that they want you to change your problem to fit their solution. As one expert respondent stated: "There isn't any productivity problem. We just have too many people working on the wrong problems--for example, on the productivity problem."

- An amazing questionnaire recently appeared in a national publication. It purported to measure IS departments in terms of whether they are leading or lagging behind in the use of advanced tools, aids,and practices. It consisted of 20 sections and 261 categories of things that (theoretically) should be done to keep a company from falling behind, as shown in Exhibit V-1.

  - One rates each category on a scale of one to five, (in terms of your use), adds up the total, and determines whether one's company is on the leading edge or is lagging behind.

  - It is an exercise in how to feel inadequate, and it is probable that few IS managers would even pass a simple quiz on the meaning of the categories.

EXHIBIT V-1

"LEADING EDGE" QUESTIONNAIRE OUTLINE

SECTION	NUMBER OF CATEGORIES
1 - Planning/Estimating/Controls	18
2 - Requirements and Design	10
3 - Purchased Software Evaluation	10
4 - Code Development	14
5 - Data Base Design	15
6 - Data Center Management	10
7 - User Documentation	15
8 - User Education	8
9 - Technical Information	10
10 - Technology Exploration	10
11 - Education of Staff	10
12 - Project Libraries	10
13 - Pretest Defect Removal	10
14 - Testing Methodologies	10
15 - Maintenance and Enhancements	10
16 - Communications	15
17 - Security and Disaster Recovery	15
18 - Personnel Categories	23
19 - Personnel Policies	21
20 - Emerging Policies	17
	261

- However, expanding down to the category level does not provide a good framework for examining the complexity of the productivity improvement problem. The requirements and design methods, code development, and data base design and development sections are broken down into their categories in Exhibit V-2.

  - For the categories in the requirements and design methods section, it is necessary to make further breakdowns into some of the particular approaches advocated by those we have come to identify with particular methodologies.

  - Under data-analytic design methods the names Michael Jackson, Kenneth Orr, Jean-Dominique Warnier, and others come to mind.

    - Structured methodologies prompt thoughts of Yourdon and Demarco.

    - Formal requirements methods remind us of IBM's Harlan Mills.

  - For code development categories, the alternatives are staggering.

    - It is estimated that 15 to 20 new products are being announced each year in the fourth-generation-language category.

    - More than 100 spreadsheet processors (in integrated systems) are available to end users.

    - Functional packages such as SAS are constantly being expanded to cover an increased range of information systems applications.

  - The categories under the data base design and development section are subject to the same proliferation of solutions.

EXHIBIT V-2

PRODUCTIVITY IMPROVEMENT CATEGORIES

SECTION	CATEGORIES
Requirements and Design Methods	1 - User Sign-Off on Requirements 2 - Data-Analytic Design Methods 3 - Structured Design 4 - Prototyping 5 - Interactive Text/Graphics Tools 6 - Interactive Syntax Checking 7 - Interactive Control Flow, Data Analysis, and Consistency Checking of Design 8 - Program "Blue Prints" 9 - Formal Requirements Methods 10 - Pseudo Code
Code Development	1 - High-Speed Prototyping 2 - Internal Reusable Code Library 3 - Commercial Reusable Code Tools 4 - Structured Coding Methods 5 - Applications or Program Generators 6 - Fourth-Generation Languages 7 - Data Base Query Languages 8 - Standard Functional Packages 9 - Spreadsheet Processor 10 - Information Centers 12 - Object-Oriented Languages 13 - End-User Programming Support Group 14 - Individual Terminals or Workstations for Technical Staff

EXHIBIT V-2 (Cont.)

PRODUCTIVITY IMPROVEMENT CATEGORIES

SECTION	CATEGORIES
Data Base Design, Development	1 - Active On-Line Dictionary 2 - Data Base Administrative Function 3 - Enterprise Business Analysis 4 - Enterprise Data Dictionary 5 - Data Base Development Tools 6 - Support for Microcomputer Data Interchange 7 - Support for Subsetting Mainframe Data Bases for Microcomputer Users 8 - Relational Data Bases 9 - Hierarchical Data Bases 10 - Network Data Bases 11 - Extended Data Bases 12 - Planning for Optical Disks 13 - Long-Term Data Retention Plan 14 - Use of Commercial Information Data Bases 15 - Document Digitizer and Associated Recall and Scanning Facilities

- Dictionaries cover a wide range of products (and INPUT has already suggested that none may be adequate for the DSD environment).

- It was recently reported that a software development team evaluated over 100 mainframe DBMSs in order to select one suitable for a securities trading and accounting program; after selecting the "best" one stated that although the DBMS was "riddled with termites" (from a security point of view) it at least provided a clear audit trail.

- Relational DBMSs are being announced at all levels in the processing hierarchy despite open questions concerning the exact definition of a relational system and performance problems inherent in the relational model. (See INPUT's Relational Data Base Developments, August 1983.)

- So it can be seen that the 20 sections and 261 categories (Exhibit V-1) really break out into practically an infinite variety of "solutions" to the productivity problem. There does not seem to be any shortage of tools, aids, and approaches to improving productivity. In fact, for those searching for the "magic bullet" of a simple solution, the availability of such a wealth of solutions presents incredible complications. Although there are no simple solutions, the mind-set of the questionnaire developer is really frightening.

  - Each category is given equal weight, and the more categories you emphasize (or use), the higher you score. The higher you score, the closer you approach the leading edge; that somehow is assumed to be "good."

  - It is possible that a company could spend itself into bankruptcy pursuing the leading edge of informations systems technology and never get a workable applications system developed.

- INPUT does not recommend the unending quest for the magic bullet or the extravagant quest to be on the leading edge. This position is supported by the companies interviewed during the research for this study.

B. RESPONDENT PRODUCTIVITY IMPROVEMENT PROGRAMS

- Respondents were asked how they thought productivity in systems and software development in their companies had changed since 1980 (when they had been interviewed as part of INPUT's comprehensive productivity study), as shown in Exhibit V-3.

- Respondents were also asked to name the primary change in their company that influenced productivity.

    - Thirty-nine percent of those indicating that productivity had improved stated the primary reason was because terminals for interactive development had been made available.

    - Thirteen percent of those reporting productivity improvement attributed it primarily to the installation of fourth-generation languages.

    - The remaining 48% of those reporting improvement attributed it primarily to a wide variety of things ranging from the very general to the very specific. For example:

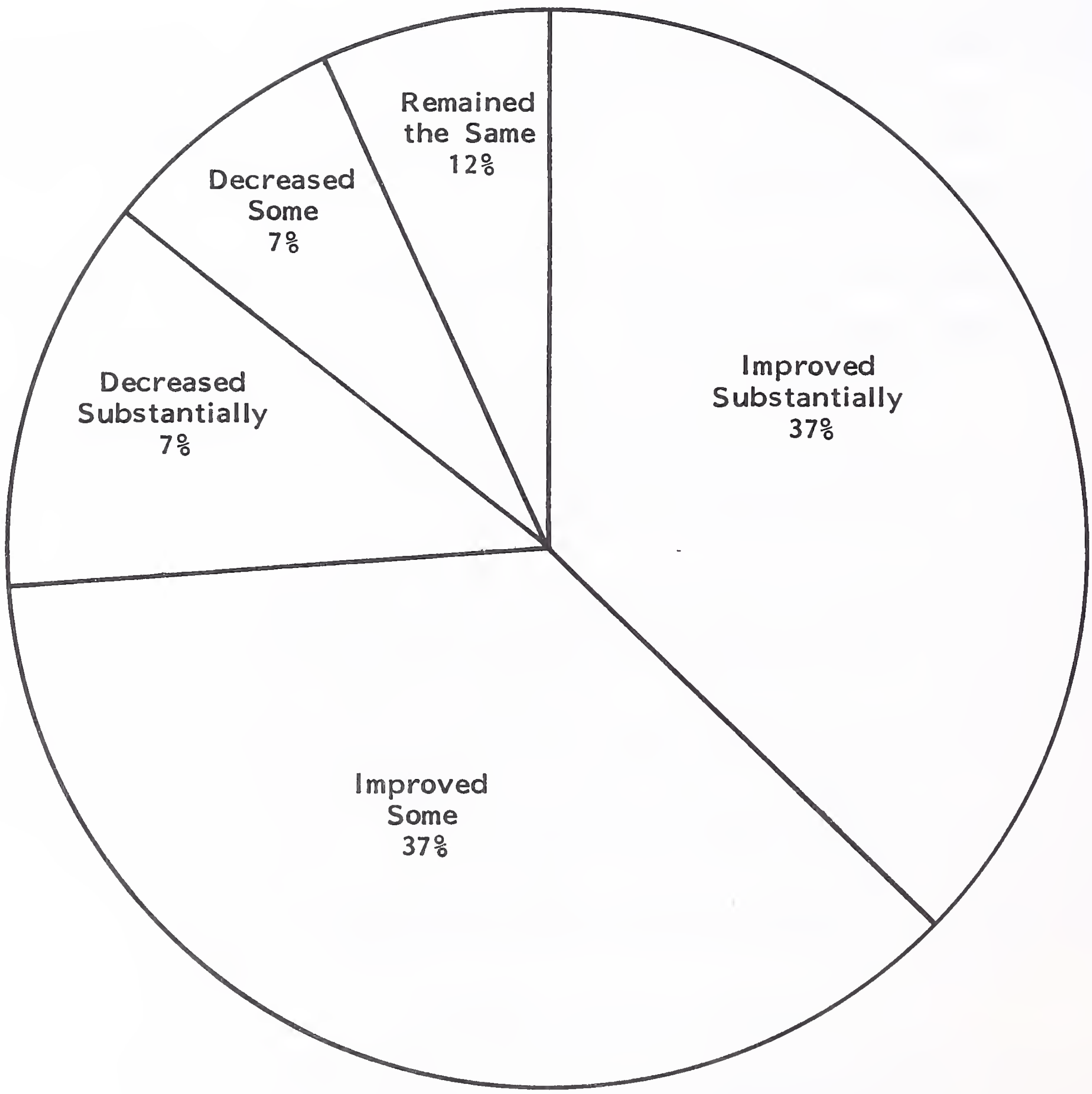
        - Structured analysis and prototyping.

        - Installing a program generator.

        - Structured COBOL.

EXHIBIT V-3

RESPONDENT ESTIMATES OF
PRODUCTIVITY CHANGES SINCE 1980



- In-house tools (developed).

- Purchase of batch software.

- Installed new DBMS.

- Use of ISPF (IBM).

- New hardware (two respondents, one installed HP 3000s, the other IBM System/38s).

- Better planning.

- When productivity decreased, the following reasons were given: "aging systems" (evidently the resources required for maintenance were the primary impact), "loaded systems" (evidently the inability to support the development of new systems), "reorganization" (implied interfacing and political problems), and "we were acquired" (probably the same impact as a major reorganization).

- It is important to recognize that turnaround problems for systems developers existed from the early days of computing, and the solution (interactive terminal support) has been around for twenty years. Elaborate studies of the impact of turnaround (and response time) on programmer/analyst productivity have been conducted, and yet during the last four years the "old solution" remains the most significant contribution to productivity improvement. (At least "individual work-stations for technical staff" was one of the 261 categories mentioned above.) It probably doesn't do to study problems to death, but it also means that companies have difficulty in cost justifying productivity improvements.

- Respondents were also asked whether they employed a systems design methodology and, if so, its specific type and impact on productivity.

  - Seventy percent stated they did employ a systems design methodology (SDM).

  - Sixty-eight percent of those employing an SDM had developed their own systems design methodology; others mentioned Yourdon, SDM-70, STRADIS, Spectrum, and "structured programming."

  - As far as the effectiveness of an SDM was concerned:

    - Forty-three percent felt it was a substantial aid to development.

    - Forty-three percent felt an SDM "helped some."

    - Ten percent felt it was "just common sense."

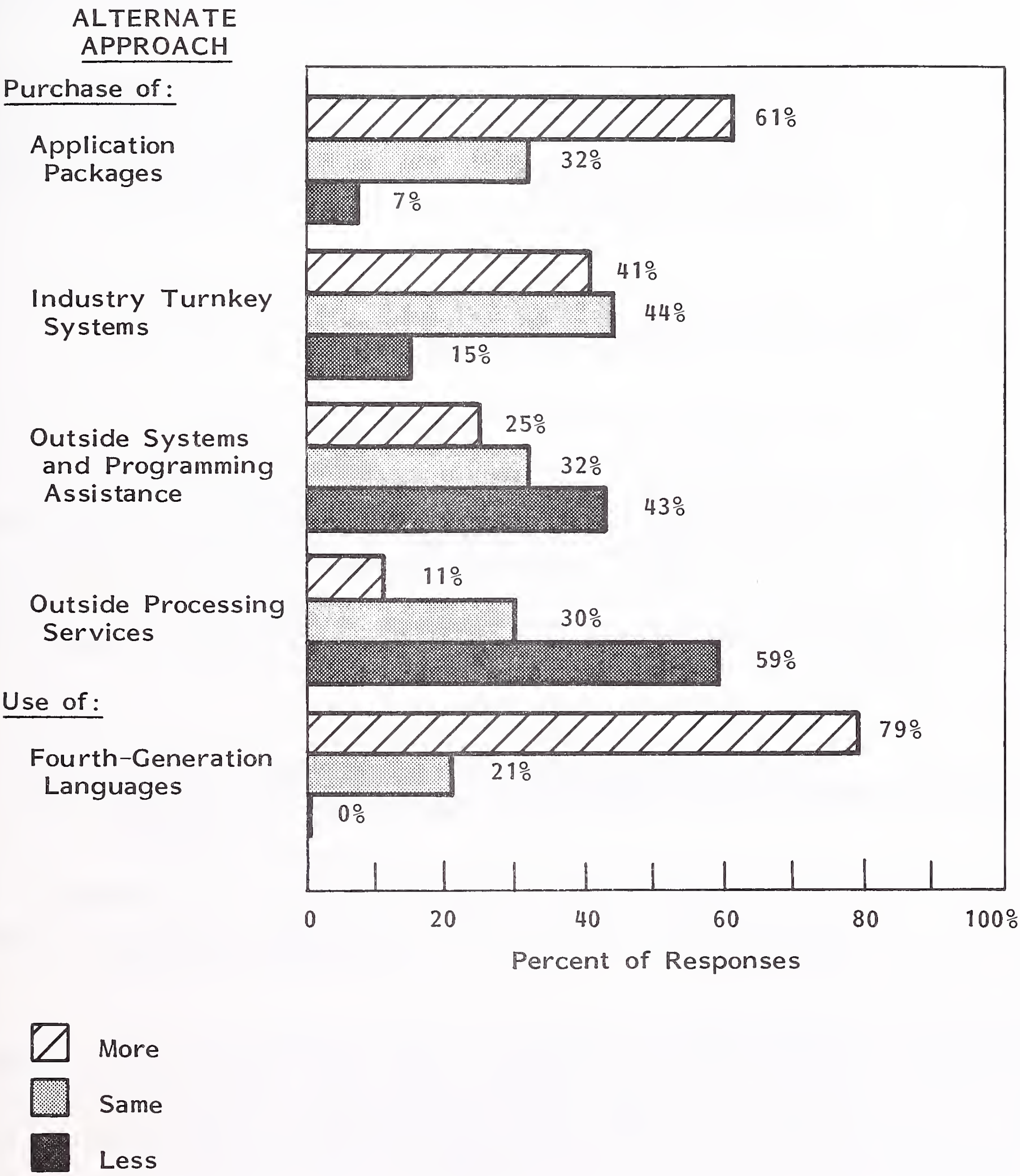
    - Four percent felt an SDM did not help very much.

- When asked about programming aids, 77% stated they used them; the ones used did not establish any significant pattern, since practically everything was mentioned. Respondents were unanimous in stating that tools were either "very effective" or "somewhat effective." This can be translated into the observation that a wide selection of programming tools are installed and that, unless they are effective, are probably dropped (or not used).

- In order to establish respondents' receptivity for alternative solutions to in-house development (and to test a perceived trend toward fourth-generation languages), respondents were asked whether they were more favorably disposed (more than four years ago) toward the purchase of applications packages, industry turnkey systems, outside processing services, outside systems and programming assistance, and the use of fourth-generation languages. Details are in Exhibit V-4.

EXHIBIT V-4

ATTITUDES TOWARD ALTERNATE PRODUCTIVITY APPROACHES
(Compared to 1980)



- Generally speaking, respondents were more receptive to the purchase of applications packages (61%). About the same number were receptive to industry turnkey systems (44%); fewer were favorably inclined towards the purchase of outside systems and programming assistance (43%) and outside processing services (59%).

- The perceived trend toward the use of fourth-generation languages was confirmed, with 79% of the respondents stating that they were more receptive, 21% remaining the same as four years ago, and none stating they were less receptive.

C. CONSENSUS OF VENDORS AND EXPERTS

- It was the general concerns of both vendors and experts that the major IS tools being emphasized to facilitate distributed systems development were fourth-generation languages, applications generators, relational DBMSs, and micro-mainframe links. At the personal computer level the emphasis is on integration of existing tools for end users (spreadsheets, word processing, and DBMSs).

  - The vendor consensus was determined by the proliferation of products in the areas cited and by prior INPUT research.

  - The experts were in turn probably influenced by the proliferation of products and by their experience with clients and within their own organizations.

  - Whereas vendors obviously felt their products were the solution, the experts were inclined to take a less sanguine attitude toward the tools and aids. Essentially, the experts felt that the current emphasis was just another example of diverting attention from the underlying problems of systems development. As one expert put it: "Anyone who thinks fourth-generation languages and relational data bases are going to solve the problem doesn't understand the problem."

- INPUT's primary concern is that there is probably very little consensus (or understanding) of what constitutes a fourth-generation language, a relational data base system, or a micro-mainframe link among vendors, experts, or users. In addition, it is INPUT's opinion that integrated systems (with expanded functional capability) for personal computers may destroy the very ease-of-use and cost effectiveness that made PCs so valuable.

- In order to put current tools, aids, and approaches into perspective, it is necessary to relate them to these environments:

  - The IBM systems software environment (essentially SNA).

  - The DSD environment, which the tools themselves tend to create as well as support (essentially the holographic model).

  - The corporate (or organizational) information flow (which, like it or not, is still an essentially paper-based) environment.

D. PRODUCTIVITY TOOLS, AIDS, AND APPROACHES IN ENVIRONMENTAL CONTEXTS

I. THE IBM ENVIRONMENT

- The terms "centralization", "integration", "differentiation" and "mechanization" as used in describing the IBM environment are used according to the definitions of general systems theory (GST) which were presented in IBM Software Direction. See Appendix B for a definition of GST terminology.

- IBM's view of the DSD environment is still from the perspective of the large mainframe, and the primary emphasis remains upon SNA and the continued extension of mainstream systems software (MVS/XA), as shown in Exhibit V-5.

  - IBM's primary emphasis is on continued centralization, and the tightly controlled distribution of both processing power and data bases. As mentioned in previous reports (See "Large-Scale System Directions: Disk, Tape, and Printer Systems, March 1983), INPUT feels IBM's plan for distributed data base (and distributed systems development) will emphasize the use of DB2 on mainframes and intelligent workstations with SQL (Structured Query Language) and QBE (Query by Example) under QMF (Query Management Facility).

  - However, IBM is confronted with the requirement of integrating a variety of hardware and software under the greater SNA umbrella (including some of its own mavericks such as System/38).

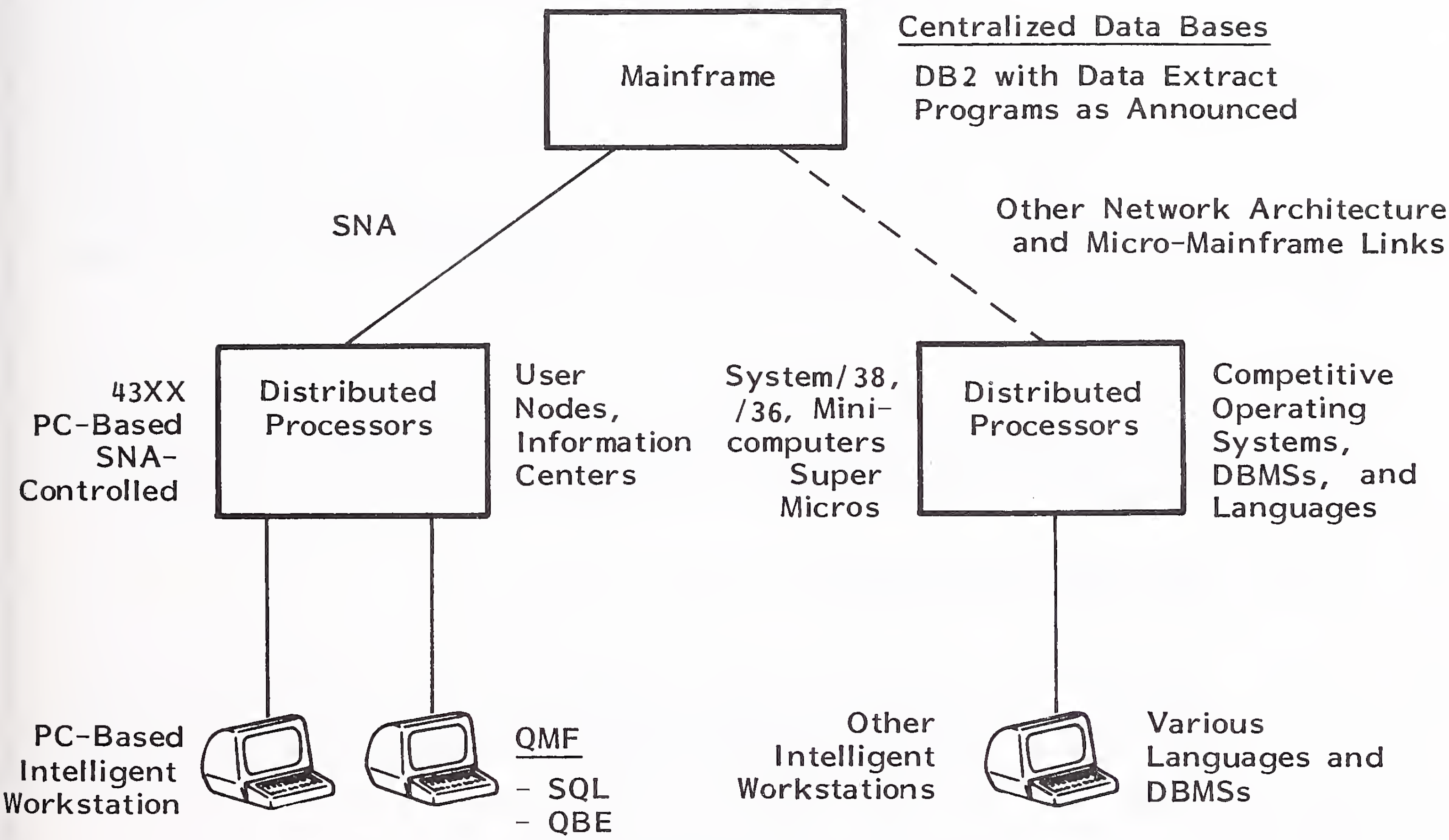
    . Minicomputers and microprocessors with UNIX-based (and other) operating systems will have to be accommodated within the DSD environment; IBM's emphasis on VM as a general operating systems umbrella is already apparent.

    . Micro-mainframe links of great variety must be integrated and attention given to developments in local area networks (LANs) as IBM keeps everyone guessing about exactly what its ultimate course will be.

  - IBM's emphasis upon centralization and its need to integrate competitive systems have led to a flexible, adaptive approach to the product differentiation that is rampant in the DSD environment.

EXHIBIT V-5

THE IBM SNA ENVIRONMENT



. IBM seems content to let languages and specialized applications proliferate for its PC-based systems, and seems confident that it can pick the winners and reap the benefits.

. This adaptive approach is evident even in IBM's marketing arrangement for Intellect (Artificial Intelligence, Inc.) and will unquestionably extend to expert systems. (The major trend is progressive differentiation in general system theory terms.)

- IBM has generally resisted mechanization of functions in hardware, preferring to maintain flexibility in the evolving software environment. Mechanization gives competitors a clearer target to shoot at than does the complex SNA/VM/MVS/IMS/DB2 type of systems development environment that IBM has created.

- Recognizing that various architectural trends progress in parallel, it is possible to state that IBM's priorities in approaching the DSD environment are: 1) centralization, 2) integration, 3) differentiation, and 4) mechanization.

2. THE DSD ENVIRONMENT

- Starting at the "other intelligent workstation" level in the IBM environment (see Exhibit V-5), it is possible to proceed from the bottom up on the same diagram and draw the following conclusions:

  - The primary systems emphasis in the DSD environment is on differentiation of product offerings:

    . Screens painted to please the individual.

    . Specialized languages and applications packages.

    . Personalized data bases (and appropriate facilities).

- Once differentiation occurs, there is a tendency to mechanize at the lowest level.

  . A workstation may be used only for word processing.

  . A language interpreter may be built into a ROM (or a spreadsheet package may be put on a chip).

  . The only data base management capability may be a relational capability.

  . A workstation may be authorized for only a single extract of data needed from the corporate (or departmental) data base. This extract could then be distributed automatically.

- However, once differentiated and/or mechanized, the emphasis must be upon integration with other distributed systems and/or central systems:

  . The exchange of information with other workstations on a LAN is obvious.

  . The micro-mainframe link is an exercise in integration.

  . The desire for integrated packages is a natural tendency to expand the functionality of the workstation.

- Since DSD is primarily a reaction to centralization (of both processing services and systems development), it has the lowest priority in the DSD environment. In fact, users do not want to become dependent on a "leading part" of the network for data or processing power. This will encourage the development of departmental data centers in competition with existing centralized facilities.

- Therefore, once again remembering that all architectural trends proceed in parallel, the DSD environment has the following priorities: 1) differentiation, 2) mechanization, 3) integration, and 4) centralization.

3. THE CORPORATE INFORMATION FLOW ENVIRONMENT

- Data is beginning to flow electronically through the corporation, but most information is still recorded, stored, and communicated on paper. In fact, despite all of the talk about electronic information systems and office automation, a good case can be made that both have contributed enormously to the paper-handling crisis and "information" explosion. As argued earlier in this report, the trend toward DSD threatens to increase the paper-handling problem while it threatens the quality of information produced.

    - The DSD environment and the tools which support it encourage the mechanization of information production--primarily in the form of paper.

        . Whether it is correspondence, management reports, or graphics, the primary means of information distribution remains paper, even though its handling has been automated through the use of electronic media and display.

        . Word processing systems, data base systems, statistical and graphics packages, spreadsheet processors, etc. all mechanize (and frequently stereotype) the production of information.

        . The ability to produce a variety of ad hoc reports, varying report formats, merging of personal data bases with official data bases, etc. will all contribute to the problems.

- The flexibility and ease-of-use associated with DSD tools facilitate differentiation in terms of format and meaning. Both types of differentiation can lead to confusion.

- The proliferation of reports in the information flow environment will lead to the need for more reports to reconcile conflicts. (Reports to reconcile other reports and to bring together audit trails are highly likely.)

- The centralization of a source of critical information is not a primary objective of the DSD environment.

- Therefore the tools to facilitate DSD can be viewed as emphasizing architectural concepts in the following order (as they apply to information flow): 1) mechanization of information production (essentially paper), 2) differentiation of format and meaning, 3) integration (forced by differentiation) of reports, and 4) centralization (of information sources) as lowest priority.

E. SUMMARY OF CURRENT TOOLS, AIDS, AND APPROACHES TO DSD

- Exhibit V-6 presents a summary of architectural trends precipitated by current tools, aids, and approaches in these systems environments.

  - At first glance, it would appear that IBM's emphasis on centralization and integration is precisely what is needed to address the problems associated with the holographic model of the DSD environment. However, the following are appropriate comments:

    - While IBM's large central mainframe-oriented strategy does provide needed central control, this is accomplished at great cost—in both hardware and software.

EXHIBIT V-6

SUMMARY OF ARCHITECTURAL TRENDS
(Associated With Current Tools, Aids, and Approaches)

GST* TREND	IBM ENVIRONMENT	DSD ENVIRONMENT	INFORMATION FLOW ENVIRONMENT
Centralization	1	4	4
Integration	2	3	3
Differentiation	3	1	2
Mechanization	4	2	1

* GST = General System Theory — for definitions see Appendix.

- There is a good possibility that heavily host-dependent systems may experience performance problems, which make them impractical.

- IBM is also working both sides of the DSD environment and is throwing pebbles into the DSD pond with abandon--selling differing solutions to the same problem.

- Increased emphasis upon mechanization in the form of data base machines, communication processors, and language translators may be forced on IBM in order to off-load the already overburdened mainframes.

- Performance is an area to which IBM has traditionally given little attention, and there are opportunities for competitive vendors to mechanize before IBM is forced to off-load highly centralized mainframe systems.

\- The DSD environment is feeding upon itself in the sense that the tools and aids gave impetus to DSD in the first place, and the general productivity problem in terms of prolonged systems development schedules (and poor IS responsiveness) demanded the tools:

- The potential impacts on quality have been emphasized in this report; tools and aids that do not address this problem will fall by the wayside.

- Because of the threat to quality, tools and aids are much more necessary for control purposes than they are to facilitate DSD. Most current products ignore this requirement.

- This information flow problem is especially troublesome because of the continued paper orientation of information systems. DSD will exacerbate this problem for the following reasons:

  . The heavy emphasis upon DSD will divert resources and attention from the integrated electronic office systems that could control the paper problem (to say nothing of the increase in paper output).

  . The continued lack of interest on the part of many systems/analysts and programmers in manual (paper) procedures.

  . The distraction of user personnel from paper-oriented problems toward the more rewarding (in terms of experience and "fun") and glamorous computer applications.

- The primary tools and aids currently being employed in these three environments emphasizes ease of implementation, and they are effective in this regard.

I. FOURTH-GENERATION LANGUAGES/APPLICATIONS GENERATORS

- Although the definition of fourth-generation languages (FGL) remains somewhat loose and measurements of improved productivity improvement remain subject to criticism, there is little question that the combination of FGL with an application generator can expedite applications development substantially.

  - Actual case studies of ADF, ADS/O, DMS, FOCUS, MANTIS, UFO, and RAMIS reveal that, although actual results vary considerably, vendor claims that a "typical application" can be developed in one-fifth the time required in COBOL (or another procedural language) are probably reasonable for small applications.

- While vendors claim improvement in systems design, data analysis, testing and maintenance, the primary improvement for FGLs is in coding portions of systems development. Since programming represents approximately 20% of the system development cost (Exhibit III-2), savings of over 15% of total development cost seem to be readily available (the 20% being reduced to 4% or one-fifth the time).

- However, the productivity "problem" is associated with the system's life cycle, and coding (during development) represents only 7% of the life cycle costs (Exhibit III-3). This would mean that life cycle costs might be reduced by only 5%, but this is nonetheless significant. For those who state maintenance is improved, one expert interviewed stated: "You don't maintain systems written in FGLs, you rewrite them."

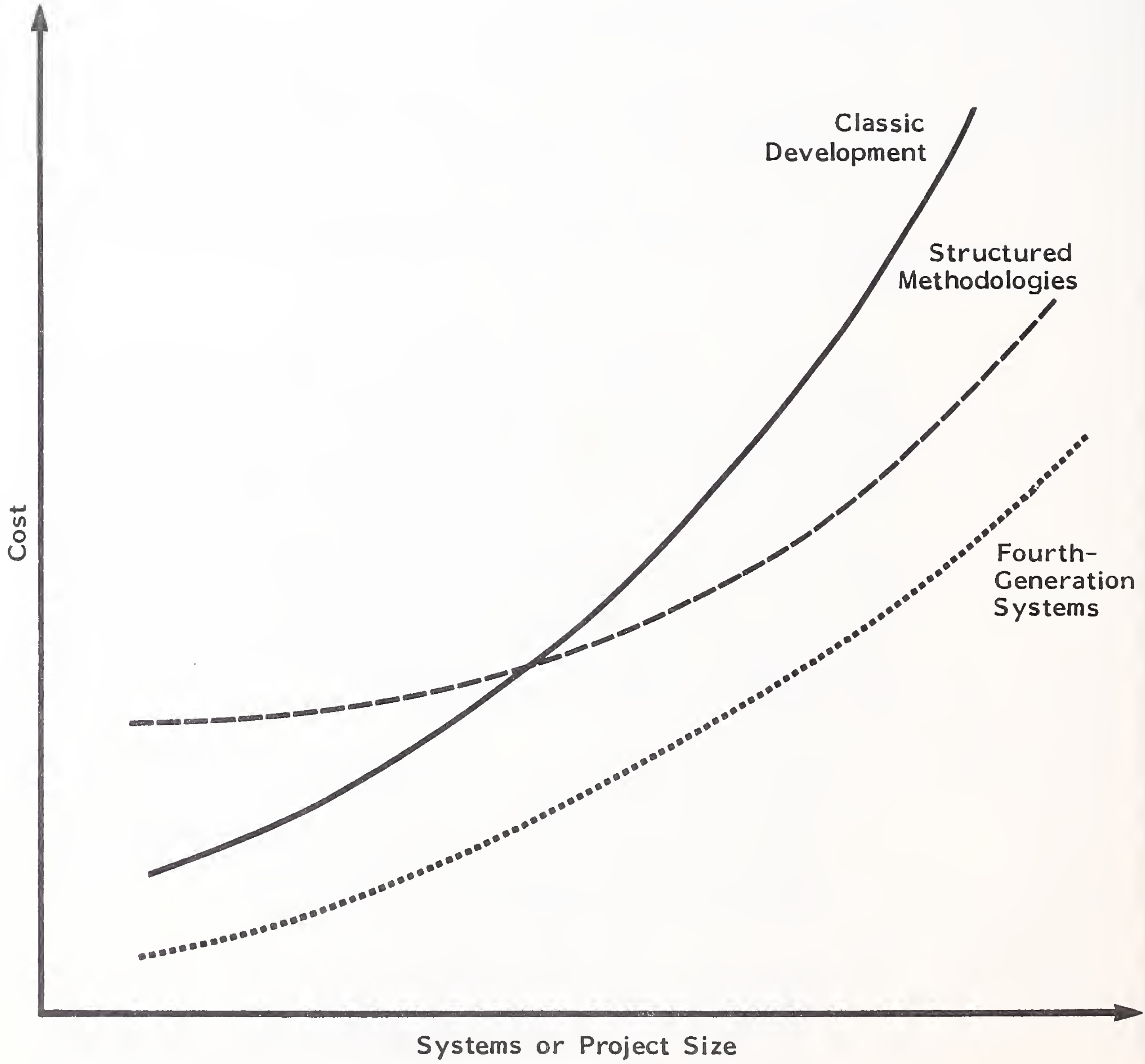
- The question of application size becomes critical in considering the impact of FGLs/applications generators. One respondent stated: "You can generate applications but you can't generate systems." Without getting involved in questions of when an application becomes a system, it is possible to state that even most vendors will concede that their products need improvement for "large application," if for no other reason than performance of generated applications. However, there is one observation about FGLs and structured methodologies that INPUT believes to be important, as shown in Exhibit V-7:

    - The use of structure methodologies has adverse effects compared to even "classic development" methodologies for small projects, but FGLs (as a tool embedded in a methodology) are effective on even small projects.

    - The crossover point for structured versus classic development is not known, but it is probably higher than the one-man-year size frequently used as a rule of thumb for mandating the use of structured methodologies.

EXHIBIT V-7

COST/SIZE IMPACTS OF
SYSTEMS DEVELOPMENT APPROACHES



Note: This chart is for purposes of illustration only, and is not scaled.

- While this still leaves open the question of applications size as an upper limit for various FGLs, it does make a valid point concerning the relative effectiveness of structured methodologies.

- One other point should be made concerning FGLs and applications generators. There is a tendency to include functional packages for statistical analysis and graphics. There is a natural evolution of SAS-type packages toward FGLs and FGLs toward incorporating more statistical facilities. This is an encouraging sign since statistical analysis and graphics can solve a high percentage of the applications requirements in a DSD environment.

2. RELATIONAL DATA BASE SYSTEMS

- When Dr. E. F. Codd delivered the 1981 ACM Turing Award Lecture, the title was Relational Data Base: A Practical Foundation for Productivity. It presented relational data base management as a foundation for attacking the productivity problem from two approaches:

  - Providing end users with direct access to information stored in computers.

  - Increasing the productivity of data processing professionals in the development of application programs.

- There are numerous relational and relational-like systems being developed for all levels of the processing hierarchy, and the degree to which they satisfy the productivity improvement objectives outlined by Codd varies considerably. However, there can be no question that the relational data model is especially appropriate for the DSD environment (distributed, end-user data bases) and for emerging expert systems as well. The primary problems with relational data base systems are:

- The persistent denial that there are any intrinsic performance penalties associated with the use of the relational model.

- The resulting, stubborn insistence of certain relational "purists" that the relational model is the solution to all data base problems.

- Implementations by purists that refuse to recognize the existence of sequential files or tables.

- INPUT believes the relational model is desirable, and perhaps even necessary, in the DSD environment. However, relational data base systems can impose severe performance penalties depending upon specific implementations, and applications systems built upon the relational model may prove impractical at certain levels of data base size and/or activity. (See INPUT's Relational Data Base Developments, August 1983, for more detailed analysis.)

3. MICRO-MAINFRAME LINKS

- Micro-mainframe links are of particular concern to users and vendors this year. INPUT has thus included numerous reports in its various programs:

  - Micro-Mainframe: Processing Services and Turnkey Systems Market Opportunities.

  - Micro-Mainframe: Personal Computer Market Opportunities.

  - Micro-Mainframe: Telecommunications.

  - End-User Micro-Mainframe Needs.

- The problems associated with distributed data bases, balance of mainframe-workstations processing, and security/protection have been previously discussed, and these problems do not seem to be understood by current PC users, vendors, and especially IS management. Current products do not adequately address INPUT's concerns.

- The problems are extremely complex, and the variety of "solutions" in terms of currently available micro-mainframe products only compounds existing communication problems among network architects, communications engineers, computer systems engineers, and those individuals developing applications in the DSD environment.

VI CONCLUSIONS AND RECOMMENDATIONS

VI CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

- The combination of perceived low productivity (or low responsiveness to end-user requests) of the IS function, combined with the rapid development of computer technology (specifically micro-processors) has resulted in end users becoming directly involved in the applications development process. INPUT refers to this rapidly accelerating trend as distributed systems development (DSD). DSD presents a number of potential problems, which seem to be generally recognized by users, vendors, and experts; however, these problems are neither fully understood nor slowing the rush toward DSD.

- The characteristics of DSD are the current emphasis on: information centers, prototyping, personal computers, and micro-mainframe links. While all of the above theoretically result in improved productivity in terms of getting applications developed more quickly, there is a substantial threat to systems quality.

  - One aspect of the threat is the impact on data/information quality.

  - The other is the impact on systems performance.

  - If either of these impacts is severe enough, the resulting systems may be impractical and never be installed, may require excess maintenance, or may have to be replaced. Any of these results has a negative impact on productivity.

- The specific threats to systems quality are composed of conflicts in the DSD environment. INPUT finds it convenient to compare these conflicts to the interference patterns of a hologram. The interference patterns that seem to be inevitable are:

  - Top-down design (as advocated by the central IS functions) being interfered with by bottom-up design (as practiced in the DSD environment). Or, depending on one's perspective, top-down could interfere with bottom-up design.

  - Security interfering with user access to corporate data bases and vice versa.

  - Added functional capability interfering with end-user ease of use.

  - Increased entropy of data/information in the DSD environment as the natural tendency to disorder is amplified by the complex interference patterns created between central and distributed data bases.

  - The interference pattern created by "unreasonable processing demands" on mainframes by intelligent workstations, and the "unreasonable expectations" of offloading mainframe processing to workstations as the DSD environment evolves.

  - Conflicting general systems trends of centralization, differentiation, integration, and mechanization, all engulfing IS management with an exceptionally complex interference pattern in hardware/software planning.

- The failure to recognize the threats to systems quality (or the disregard of these threats) has resulted in a restructuring of INPUT's productivity pyramid (Exhibit III-4) to give top priority to end-user involvement and little commitment to quality (Exhibit III-11). In addition, undue emphasis is currently being given to the quest for magical solutions to the productivity problem by finding the right tool, aid, or approach. This emphasis and the DSD environment itself has resulted in:

    - The proliferation of tools, aids, and approaches to the productivity problem that are too numerous to mention and virtually impossible to evaluate except on an individual basis.

    - The mere availability of so many "solutions" and the attitude, prevalent among some in the industry, that more is better actually exacerbates the productivity problem. More resources get expended on analyzing the software development problem and buying solutions to fix that problem and less resources (human and financial) are available to solve the parent companies' information systems problems.

- The tools, aids, and approaches to solving the software productivity problems are currently directed primarily toward facilitating the DSD environment (since the distribution of systems development responsibilities to end users is itself considered to be a solution). To the degree that the DSD environment adversely impacts systems quality, these tools, aids, and approaches will be counter-productive.

    - These tools, aids, and approaches are available in great quantity and are effective in expediting applications development.

    - While their effectiveness varies according to the particular requirements of the individual organization, properly applied they can improve productivity. There is nothing inherently wrong with productivity tools and they obviously should not be avoided because they have the potential to adversely impact quality (just as hammers should not be banned because they can be used as bludgeons).

- It remains INPUT's opinion that an effective productivity improvement strategy must address all of the factors listed in the productivity pyramid, and that an effective program must be built from the bottom up. Giving primary emphasis to a commitment to quality in a DSD environment is more difficult than it was when information system development was highly centralized, but it is essential. Paradoxically, the commitment to quality requires that tools and aids for quality measurement, monitoring, and improvement be built into the base of the pyramid (while conventional tools and aids remain the capstone to be put in place only after the rest of the productivity improvement program is firmly in mind). The reconstruction of the pyramid will be discussed under recommendations.

- Research for this study disclosed that most users are doing very little about controlling the development of DSD, have given little thought to solving the problems they anticipate, and have few ideas about tools and aids to assist in quality control. INPUT has concluded that the DSD environment must be viewed from the perspective of the data/information flow process (as opposed to the perspective of discrete systems and/or applications) for purposes of quality control. With that in mind, several categories of tools and aids to control the DSD environment were outlined in the body of this report.

  - Expanded data/information dictionary capability to permit data prototyping and necessary data/information flow monitoring.

  - The development of a meta language (with appropriate translators) to facilitate communications among mainframes, minicomputers, and intelligent workstations (and the people who use them).

  - Data flow performance monitors/estimators to assist in determining location and structure of distributed data bases.

- An integrated system for access control, information flow control, and data base certification (at various levels) is required for purposes of security and protection.

- A development of a communications command language that will facilitate a view of various data/information structures and their transformations as they flow through the network. For example, icons to depict the difference between paper files, stored images, and encoded data will become necessary (as will various data models for encoded data).

- Easy-to-use, and even automatic, tools for end-user security and protection of data/information are required.

- Tools and aids to promote an understanding of data/information entropy in both storage and communications are required. This implies monitoring and prediction of processing required to maintain order among data at various processing levels and redundancy in communication necessary to assure understanding of information transmitted.

- As the tools of operations research and artificial intelligence are applied to decision support systems, tools to analyze these tools (primarily from a performance point-of-view) are required.

- A document storage and control system to serve as a means of integrating existing paper systems with electronic systems is required.

- The tools and aids outlined in this report will be described in more detail in the companion report, Impact of New Software Productivity Techniques. This report is recommended for those who may consider in-house development of tools and aids. Appendix C contains a more detailed summary of features and functions required.

- Many of the tools and aids defined for developers require additional research and inventions before they can be fully developed (this is especially true in the security/protection area). However, the lack of tools and aids and the complexity of the problems involved should not deter IS management from recognizing its true role in the DSD environment and from taking necessary action to effect the changes in attitude, and perhaps organization, that are required.

B. RECOMMENDATIONS

- The first thing to do is to explain the problem to management. Be blunt. State it simply: there is a substantial risk that increased investment in computer hardware and software may lead to deterioration of essential corporate data and information unless central control is exercised over the DSD environment.

- Present a plan of action that places emphasis on maintaining and improving the quality of information systems as the foundation of a productivity improvement program. Essentially, this plan represents the reconstruction of the productivity pyramid from its current state of disarray (see Exhibit III-11).

- Commitment to quality should include the following actions (unless they have already been taken):

  - In addition to development efforts under direct control of IS, walk-throughs should be conducted at appropriate levels for the following:

    - Requests for data from central data banks—know what the planned use is.

- Prototyped systems, regardless of the composition of the development team.

- Any system producing reports on a continuing basis, regardless of whether the reports originally started ad hoc from information centers, departmental data bases, or personal data bases. (This implies the establishment of a document control system.)

- Quality objectives should be established for all systems that are expected to go into production (or produce continuing reports). This should be done early in the systems development cycle, and the objective should be used to structure subsequent walkthroughs and testing. The objectives should cover:

    - Functional capability.

    - Performance characteristics (response time, processing power required, data base size, etc.).

    - Data requirements.

    - Supporting systems and procedures (including documentation).

    - Adherence to standards.

    - This requirement should apply to prototypes and specify iterations and/or checkpoints.

- A test plan should be required during the analysis phase of the development cycle. The plan should be geared to the quality objectives.

- The vital information (and underlying data) necessary to run the enterprise should be identified, isolated for special treatment (protected, secured), certified, and audited. The document control system should assure that documents are clearly labeled as to the data and systems that produced them. For example:

- A "certified" document could require that it be prepared directly from a secured data base by fully tested systems.

- Documents produced by prototype systems from a secured data base might be qualified with a statement that the producing programs had not yet been fully tested and might contain errors. (Similarly, fully tested programs using other than certified data should stipulate the data quality.)

- Documents produced from departmental (or personal) data bases containing data not registered with the central data facility should also be clearly labelled.

- At the lower limits, it might be necessary to generate a document label stating: "This document was produced from John Smith's personal data base, on his home computer, with the assistance of his 10-year-old as outside consultant."

- All quality assurance actions should be rigorously applied to purchased software as well as to that developed internally.

- End-user involvement is the name of the game in the DSD environment, but that involvement does not necessarily imply harmony with the IS function. In some companies, the effort at this level may lead to the IS department (or some central organizational entity) becoming directly involved in what the users are doing. The following actions could be taken to assure appropriate end-user involvement:

- DSD should be approached in a positive manner by the IS function. It is an opportunity to improve user relationships. The problems identified in this report should be presented as problems that must be jointly solved in order to maintain and improve quality of information and to make effective use of computer/communications technology. Do not adopt the attitude that users should learn the hard way—quality is a shared responsibility.

- Make maximum use of the DSD environment to improve user relations. Information centers, prototyping, and micro-mainframe links do provide appropriate vehicles for better communications. Respondents to this study mentioned education (both user and IS) as being of importance, and it is. Be sure that a good education program is established (both formal and informal).

- Build competence in areas that will cause users to seek assistance-- central data bases afford the opportunities for mutual involvement, consultants in advanced analysis techniques will be needed, traditional systems programmers will remain necessary, and one must become competent in the user's technology (hardware and software).

- A document control system implies involvement in paper systems and procedures that many IS departments have tended to ignore over the years. Many users will welcome the interest, and most IS departments will have a lot to learn. This involvement is going to be essential as office automation progresses and the paper information flow is brought under control. Start now.

- Security and protection is a nasty problem in the DSD environment and mutual understanding of responsibilities is going to be essential. Classify data and establish procedures for end-user protection of those data. Help users to protect themselves, but it now also becomes essential for the central facility to get its own house in order.

- Broadbased management is an extension of end-user involvement in the DSD environment. Terminals are beginning to appear not only at the clerical workstation but on the manager's desk. The systems being developed are not only important to operating and general management, but also are essential to the performance of management's specific functions.

  - All of the recommendations made concerning user involvement apply to management, but the integrated nature of the systems being developed must be emphasized. The interdependence of data and information across organizational lines must be understood, and the importance of quality should become apparent. The need for broadbased management is primarily to assure that the quality of information flow is maintained and improved.

  - The information systems plan should be tightly coupled with the business plan (not merely prepared in parallel). In fact, it should be made clear that the business plan cannot be any better than the information systems plan; on the other hand, the reverse is not true.

  - Support of top management should be obtained in stressing that subordinate management will be evaluated based on the quality of data and information interchange required to support the business plan. Attempts to subvert information quality for purposes of personal or organizational advantage (corporate politics) should be recognized as counter-productive at the very least. Active participation of management at all levels in information systems planning and development, from corporate IS budgets down to evaluation of individual management documents, should not be difficult to obtain once the importance and potential problems are understood.

- Effective personnel in the IS organization are essential to a productive environment, but the role of the IS organization must be clearly understood before even staffing requirements can be determined. It should be obvious by now that INPUT is convinced that the primary role of IS in the DSD environment is quality assurance of data and information. Once the base of commitment to quality, user involvement, and broadbased management has been established, organizational and staffing requirements become apparent.

- A separate quality assurance function should be established with personnel capable in the following areas:

    - Software systems design and implementation sufficient to develop quality objectives and lead walk-through panels or teams.

    - Systems testing, including both hardware and software performance. This implies a rather high degree of proficiency with statistical techniques and performance metrics.

- A consulting organization or function should be established (or enhanced) to include not only conventional applications and systems programming capability but the following as well:

    - An operations research and artificial intelligence unit should be formed (even if it consists of only one or two people initially).

    - Intelligent workstation (PC) specialists will be required.

    - Expertise in "paperwork management" should be included--everything from forms design, through office systems and procedures (including filing), to alternate storage media such as micrographics and optical disks.

- The data base administration function should be strengthened--really strengthened into an information management role. (Note the difference in level and action orientation represented by the words "administration" and "management"; also the difference between "data" and "information" that includes voice and image-based representations.) There must be a focal point for data and information flow control and quality assurance, and the existing data base administration should be absorbed into a new organization that includes the following capabilities:

- Expertise in data models and structures (this is too important to be left to applications programmers and/or designers) should be centralized because qualified personnel in this area are in shorter supply than are systems programmers.

- Knowledge of information theory, which is in even shorter supply, becomes critical in the DSD environment where high data entropy is inevitable and must be understood.

- Security and protection specialists will be required, and some of these problems are on the leading edge of computer and mathematical science.

- Information specialists, including librarians, capable of locating data and information sources from both internal and external sources are required (separation of electronic and paper data bases is a mistake).

- Recognition that effective personnel may have to be developed in these areas because they are not currently available is essential--if you view data base administration as a clerical function you are in real trouble.

- Insist that the corporate personnel department concentrate on the specific problems of establishing a stable, effective information systems staff.

- Arriving at the top of the productivity pyramid (tools and aids), it becomes apparent that the magic bullet for productivity improvement does not exist. However, once the lower levels of the pyramid are in place, there are specific things that can be done to pursue appropriate tools and aids in a meaningful fashion.

    - Standards can be established for use. FGLs will be used only for development projects of a certain size, the relational model will be used only for data bases of a certain size, all archival files will be sequential, structured methodologies will be employed on all systems of a certain size, etc. Even if such guidelines are at first relatively crude, they should nevertheless be established and refined.

    - Requests for tools and aids are directed to quality assurance just as any proposed system would be. Requirements flow to the base of the pyramid.

    - Development and purchase of tools and aids should be subjected to quality assurance. They are fed in through the base of the pyramid.

- INPUT remains convinced that the only solution to the "productivity problem" is the ordered approach represented by the productivity pyramid. The DSD environment holds the potential for chaos, but provides the impetus to do something about the problem. Productivity improvement is primarily a management problem. Effective management consists of more than "viewing with alarm" and protecting the status quo. Information systems management that does not now exercise leadership runs a high risk of being relegated to a clerical and custodial function, if it survives at all.

APPENDIX A: QUESTIONNAIRE

CATALOG NO. USSP□□□

USER QUESTIONNAIRE

1. In 1980 your company was interviewed as part of INPUT's comprehensive study on improving productivity in systems and software development. In the last four years, do you feel that productivity in your company has:

☐ Improved Substantially

☐ Improved Some

☐ Remained the Same

☐ Decreased

☐ Decreased Substantially

2. What do you feel is the most important change that has occurred (in your company) and has influenced productivity?

____

____

____

3. Do you currently employ a systems design methodology in your systems development cycle?

a. ☐ Yes ☐ No

b. If yes, which one(s)?

____

____

c. How do you personally feel about it?

☐ It is a substantial aid to development.

☐ It helps some.

☐ It is just common sense.

☐ It doesn't help very much.

☐ It creates a lot of work.

4. Do you currently employ programming aids during the normal systems development cycle?

a. ☐ Yes ☐ No

b. If yes which one(s)?

____

____

____

c. How effective do you personally feel these tools and aids are? (Identify which.)

1	2	3	4	
___	___	___	___	Very Effective
___	___	___	___	Somewhat Effective
___	___	___	___	Not Very
___	___	___	___	A Waste

5. It is INPUT's conclusion that the most noticeable change in the systems development process in the last four years has been to get end users directly involved. INPUT refers to that change as Distributed Systems Development - DSD. Do you currently:'

	Yes	No	Planned
Have an Information Center ?	☐	☐	☐
Use prototyping? (with end-user involvement)	☐	☐	☐
Make significant use of PCs? (Significant use means that some of the end users are doing work that would have formerly been done by IS.)	☐	☐	☐
Are PCs linked to mainframes?	☐	☐	☐

6. We would like to get your opinions about DSD in general. Do you agree or disagree with the following statements?

	Agree	Disagree
It has relieved user pressure on IS.	☐	☐
End users are happier.	☐	☐
Systems get done faster.	☐	☐
Programs get done faster.	☐	☐
Backlog has been (or will be) reduced.	☐	☐
DSD helps IS do its job.	☐	☐
DSD is going to cause problems.	☐	☐
End users don't know what they are doing.	☐	☐
IS productivity is improved.	☐	☐
End-user productivity is improved.	☐	☐
Corporate productivity is improved.	☐	☐
The systems will have to be rewritten by IS.	☐	☐

7. What are the main advantages and disadvantages of:

a. Information Centers

Advantages: ____________________

____

Disadvantages: ____________________

____

b. Prototyping

Advantages: ____________________

____

Disadvantages: ____________________

____

c. Standalone PCs

Advantages: ____________________

____

Disadvantages: ____________________

____

d. Micro-Mainframe Links

Advantages: ____________________

____

Disadvantages: ____________________

____

8. Past research has indicated that systems maintenance is the most costly part of the system's life cycle. What impact is DSD going to have on maintenance?

____

____

9. What tools and aids are currently being used to facilitate end user systems development? (Ask for both name and vendor).

Name	Vendor

10. How are these tools and aids selected?

IS User Combination

11. A great deal of apprehension has been expressed about micro-mainframe links. How serious do you consider the following potential problems?

	Very Serious	Somewhat Serious	Not a Problem
Data Base Synchronization	☐	☐	☐
Data Base Integrity	☐	☐	☐
User Understanding of Data	☐	☐	☐
Conflicting Reports to Management	☐	☐	☐
Mainframe Capacity Planning	☐	☐	☐
Mainframe Performance Impact	☐	☐	☐
IS Visability	☐	☐	☐
Cost to Company	☐	☐	☐
Impact on IS Budget	☐	☐	☐
Systems Quality	☐	☐	☐
IS Loss of Control	☐	☐	☐
Data Security/Protection	☐	☐	☐

12. a. What are you currently doing about these potential micro-mainframe problems?

____

____

b. What do you plan to do? _______________

____

13. What tools do you need to facilitate DSD? _______________

____

____

14. What tools do you need to control DSD? _______________

____

15. Have you either considered or heard of any tools for facilitating or controlling DSD that you consider promising? Specify name and vendor.

____

____

16. Compared to four years ago, how receptive are you to:

	More	Same	Less
Applications Packages	☐	☐	☐
Industry Turnkey Systems	☐	☐	☐
Outside Processing Services	☐	☐	☐
Outside Systems & Programming Assistance	☐	☐	☐
Fourth-Generation Languages	☐	☐	☐

17. How do you measure productivity improvement?

____

____

____

18. What is your understanding of knowledge-based or expert systems?

____

____

____

19. Do you anticipate that expert systems will be developed internally or purchased?

a. ☐ Purchased ☐ Developed

b. If developed internally how would they impact productivity of IS?

____

____

c. How do you feel expert systems will effect productivity of those using them?

____

____

20. How do you cost justify the purchase of productivity improvement tools and aids?

____

____

____

21. Any additional comments concerning productivity improvement?

____

____

____

APPENDIX B: DEFINITIONS OF GST TERMINOLOGY

APPENDIX B: DEFINITIONS OF GST TERMINOLOGY

- While it is beyond the scope of this study to pursue GST in any detail, it is important to understand four relatively simple concepts. GST states that all systems are subject to:

  - Progressive Centralizations which means that leading parts emerge to dominate the behavior of the system.

  - Progressive Integration which means that the parts of the system become more dependent upon the whole.

  - Progressive Differentiation in which the parts are becoming more specialized.

  - Progressive Mechanization which states that, while the overall system exhibits a wider repertoire of behavior, this is accomplished by progressive mechanization which limits parts of the system to a single function.

APPENDIX C: TOOLS AND AIDS FOR CONTROLLING THE DSD ENVIRONMENT

APPENDIX C: SUMMARY OF FEATURES AND FUNCTIONS

A. BACKGROUND

- Most information systems (IS) managers responsible for supporting the DSD environment do not have sufficient knowledge or resources to evaluate the tools, aids, and approaches to productivity improvement which are currently associated with that environment. This summary is intended to enrich IS managers' knowledge and provide a framework of understanding.

- It is rather important to understand exactly what an information system is, and the current status the hardware/software technology to support such systems, before determining how such systems can be most effectively implemented. The first thing to recognize is that information systems existed before computers, and consist of only five primary processes: input, communications, calculations/manipulation (processing), storage, and output. The most important thing is that is happening in information systems is the change from paper to electronic media.

  - The historical information is presented only for perspective. The mechanical (and electro-mechanical) devices developed in the 1800s (cash registers, adding machines, punch card equipment, and typewriters) have been severely impacted by electronic counterparts. However, the telephone and telegraph remain virtually unchanged in terms of functions (despite electronic implementations).

- However, at this point, only the calculation/manipulation process is currently predominantly electronic rather than paper oriented. Very few paper and pencil calculations are performed, and mathematical tables have effectively been obsoleted.

- The other processes remian predominantly paper oriented.

  . Paper reports and records of transactions remain the primary source of entry into both information flow and computer data bases despite the development of some major operational systems which capture data at its source.

  . Paper remains the primary communications medium between individuals and systems.

  . Despite rapidly increasing use of magnetic storage and micrographic storage most information resides in paper libraries and file cabinets, and the volume of paper documents requiring storage (or disposal) continues to grow at an alarming rate.

  . As far as output is concerned, it is not unfair to state that the application of computer technology and office automation products has increased the amount of paper output astronomically, and then, in turn, has created the current productivity problems among white collar workers in general.

- It is, therefore, of extreme significance that technology to control this paper glut is becoming available. Specifically, it is INPUT's opinion, that the availability of cheap, optical storage is key to less paper in information systems, as opposed to paperless offices which will require reorientation of the entire work force.

- The important conclusion is that the substitution of electronic for paper media (in the fundamental information system processes) represents the primary design point for the systems which will be developed during the late 1980s and 1990s. This has many ramifications for IS management.

  - Understanding and becoming involved in current paper-based information systems and procedures becomes imperative for IS management.

  - The quality impacts of the DSD environment on information flow become of increasing importance as the replacement of paper-based systems and procedures becomes imminent, and IS management faces its responsibility for facilitating this replacement.

  - The tools, aids, and approaches IS management is going to need are going to become apparent only after current information flow is clearly understood and the impact of new hardware/software technology is fully appreciated.

- The DSD environment is designed to improve productivity in the sense of being able to provide quick answers to specific requests for information, typically ad hoc reporting, special analyses, and "what if" queries. To the degree that the quality of information systems is impacted by this environment, the most likely questions from management will probably change to "why?"

  - Why don't these reports agree?

  - Why does this information cost so much?

  - Why is this information wrong?

  - Why isn't the data base any good?

- Why do we need a bigger computer?

- Why do we get different answers to the same question?

- These questions will be substantially more difficult to answer than were the original requests for information, and they will have severe impact on both the productivity and the credibility of the IS function.

- Therefore, while IS management may not be able to specify the tools, aids and approaches they require in order to improve productivity, it is possible to formulate requirements by anticipating the hardware/software technological environment, the types of information systems which will be possible, and the problems which will be inherent in the development of these systems.

B. PRODUCTIVITY TOOLS AND AIDS IN THE DSD ENVIRONMENT

- It is obviously beyond the scope of this study to specify new tools and aids in detail. Some may appear to be currently available, and others may be merely a research direction to determine the best solution to a problem. However, they are all directed towards restoring commitment to quality to its proper place in the productivity pyramid. In addition, these tools and aids have two additional attractions:

  - They are especially well suited for providing not only control in the DSD environment, but also the necessary data and information to develop systems requirements and specifications for the electronic office (office automation system).

  - They will establish the foundation for extending decision support systems to expert systems (where quality must be fundamental) by providing at least a preliminary connection between data/information bases and the decision making process. By tracking data/information flow and associating it with use in decision making potential for expert systems may be identified, and at least some of the inputs to future knowledge bases will be qualified. The developers of expert systems have already determined that they must be prepared to answer "why?" questions. Understanding what any system is doing is fundamental, the quality control and improvement, and expert systems will require rigid and continuing quality contol.

I. AN INFORMATION BASE MANAGEMENT SYSTEM (IBMS)

- Earlier in this report the need for "expanded data/information dictionary capability" was identified as an aid to productivity improvement. Actually, as the requirements became more clearly understood, it was apparent that, even in it's simplest form, the system required was much more comprehensive than any extension of a current data dictionary. For lack of a better term, INPUT has called the system an Information Base Management System (IBMS). Essentially, it is a central locater of information sources, and can most easily be described as a supervisory system for other data/information dictionaries and directories.

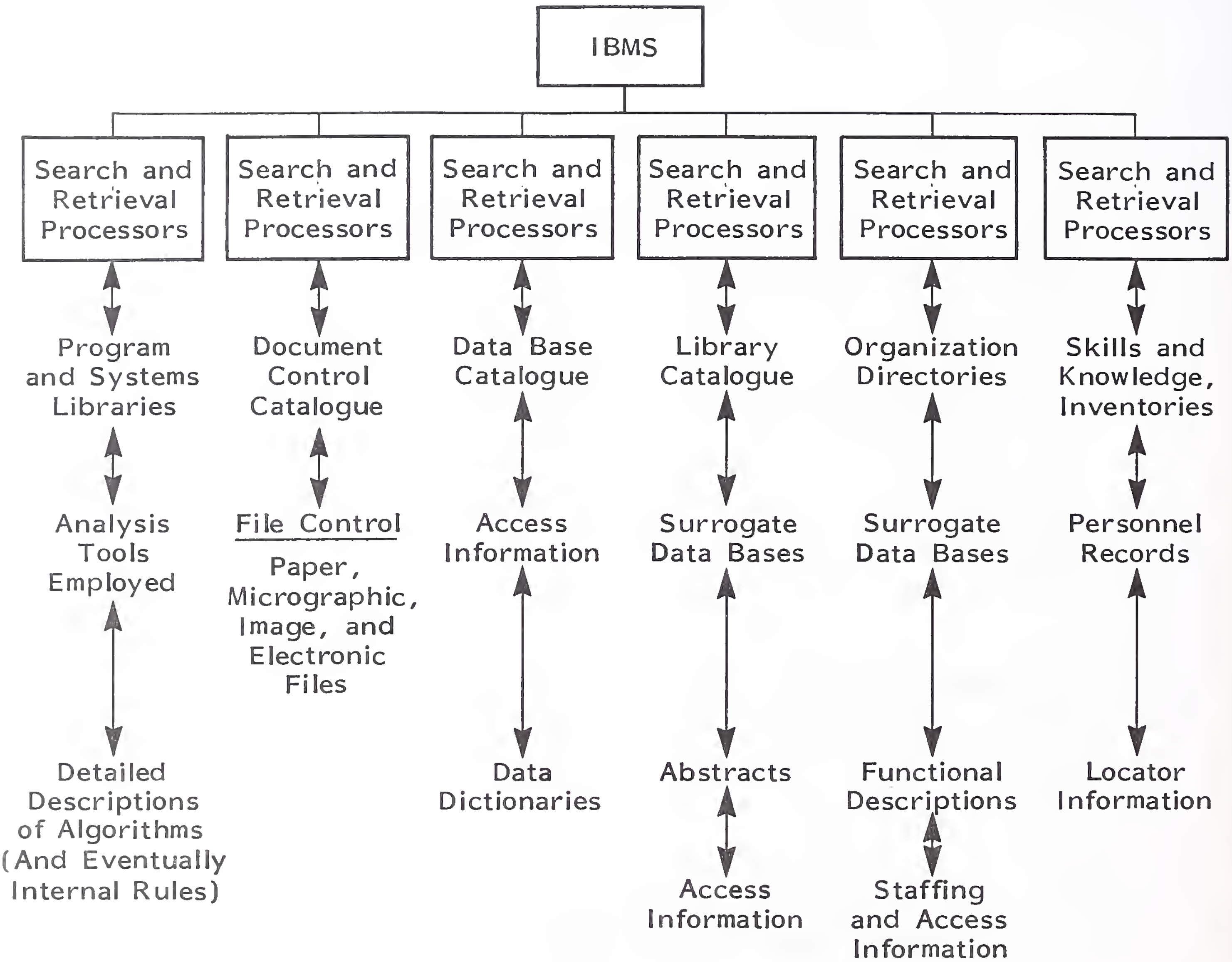
- The complexity of the system arises as soon as it is recognized that human beings and organizations are information sources as well as data bases, libraries and file cabinets. A rough diagram reveals the comprehensive nature of such a system, as shown in Exhibit C-1.

    - In its simplest implementation the IBMS could merely provide central access to various catalogues, directories, and dictionaries.

    - Search and retrieval capability against these catalogues, directories, and dictionaries could be enhanced by prompting to refine inquiries (and reduce information references).

EXHIBIT C-1

AN INFORMATION BASE MANAGEMENT SYSTEM



- More detailed vertical penetration would permit rapid browsing. For example, abstracts and descriptions could be prescanned and surrogate data bases developed (essentially key words are extracted) for fast searching. The bottom of the vertical chains would always provide specific access information and detailed descriptions and definitions of what is being accessed.

- In addition, the software programs used and/or available to develop information from data would be available. In other words the tools and aids would be classified and retrieved based on the information desired.

• Ultimately, the ability to associate and qualify the various information sources with each other in a meaningful domain for research and analysis on a specific project (subject) would be the goal of IBMS. This would have the following ramifications:

  - It would be an "expert system" in the sense that it would not search based on specified algorithms and would present a preliminary knowledge base rather than a list of information sources.

  - The ability to locate, qualify, and associate various information sources during the requirements phase of the systems life cycle would be an extremely effective productivity tool. Specifically:

    . Redundant information systems would not be developed where adequate information already existed.

    . Software systems unsupported by necessary data/information could be avoided.

  - Indeed, in certain decision support situation, the query might provide all necessary information. (The IBMS should be viewed as a shared resource among the IS department, information centers, and end users.)

- Since the quality of data and information are fundamental to systems quality, the IBMS is a necessary quality control mechanism.

- The development of an efficient IBMS obviously requires a great deal of effort (and perhaps invention) on the part of systems software developers and information specialists (librarians, data base administrators, etc.) However, it has the advantage of being modular and lends itself to phased implementation. Essentially, it is a shell to facilitate integration of existing systems.

2. A DOCUMENT CONTROL SYSTEM (DOCS)

- Perhaps the most important missing subsystem under the IBMS is a comprehensive document control system (DOCS). Most organizations have automated portions of the process (mailing lists, classified documents, engineering drawings, etc.), but IS departments have normally ignored paper-based systems and procedures until there are demands for computer-based systems. In addition, the paper mill mentality has contributed to the problem by facilitating the production of paper reports.

- The DOCS system should provide for the following:

  - Distribution control, perhaps enforced by requiring all forwarding of documents to be addressed through a central directory.

  - Classification, not only for security purposes, but for information quality. For example:

    . Produced from certified central data base by production programs.

    . Produced from certified central data base by prototype system.

- Produced from control data base extract by a specific personal computer software package.

- Produced from personal or organizational data base by registered program (tested and installed centrally).

- Produced from personal data base by special program.

- The variety of categories is enormous and obviously must be tailored to the organizations requirements, but the restriction of classification provides a means of lowering information entropy.

- Footnoting, in order to associate the specific document with other information under the IBMS. These could be selectively printed on the document or available upon request.

- Retention information, pertaining to the storage retrieval and disposal of the document (either paper or electronic).

- The DOCS system is obviously essential so more documents are stored on magnetic and/or optical media, but it also should provide the means for data and information flow analysis which is so essential in the DSD environment. The work required to implement such a system is considerable, and the need for imaginative tools and aids is limited only by the creativity of those addressing the problem. Once the DOCS structure has been established, the need for more refined analysis tools and control mechanisms will become apparent.

3. DATA FLOW MONITOR (DFM)

- At the present time, there is a tendency to down load data to microprocessors in report format, and it is possible that with a fully developed DOCS, data flow could be monitored by merely associating the document with the data base of IBMS, as shown in Exhibit C-2. However, since data will be distributed both through reports and through direct requests for data transfer, and it is anticipated that some of these requests will be "unreasonable".

- Data flow among systems and intelligent workstations must be monitored to determine performance (and cost) impact on the communication network, host systems, processing nodes, and intelligent workstations. A host controlled data flow monitor (DFM) will be essential if a proper distribution of both data and processing are to be maintained.

- DFM becomes activated at the point where data/information is to be actually transferred from or among systems (host or development processors) and/or intelligent workstations (in other words after the data/information has been located using IBMS). However, since the request for location information must be monitored, DFM treats IBMS as simply another query system to be monitored, as shown in Exhibit C-2.

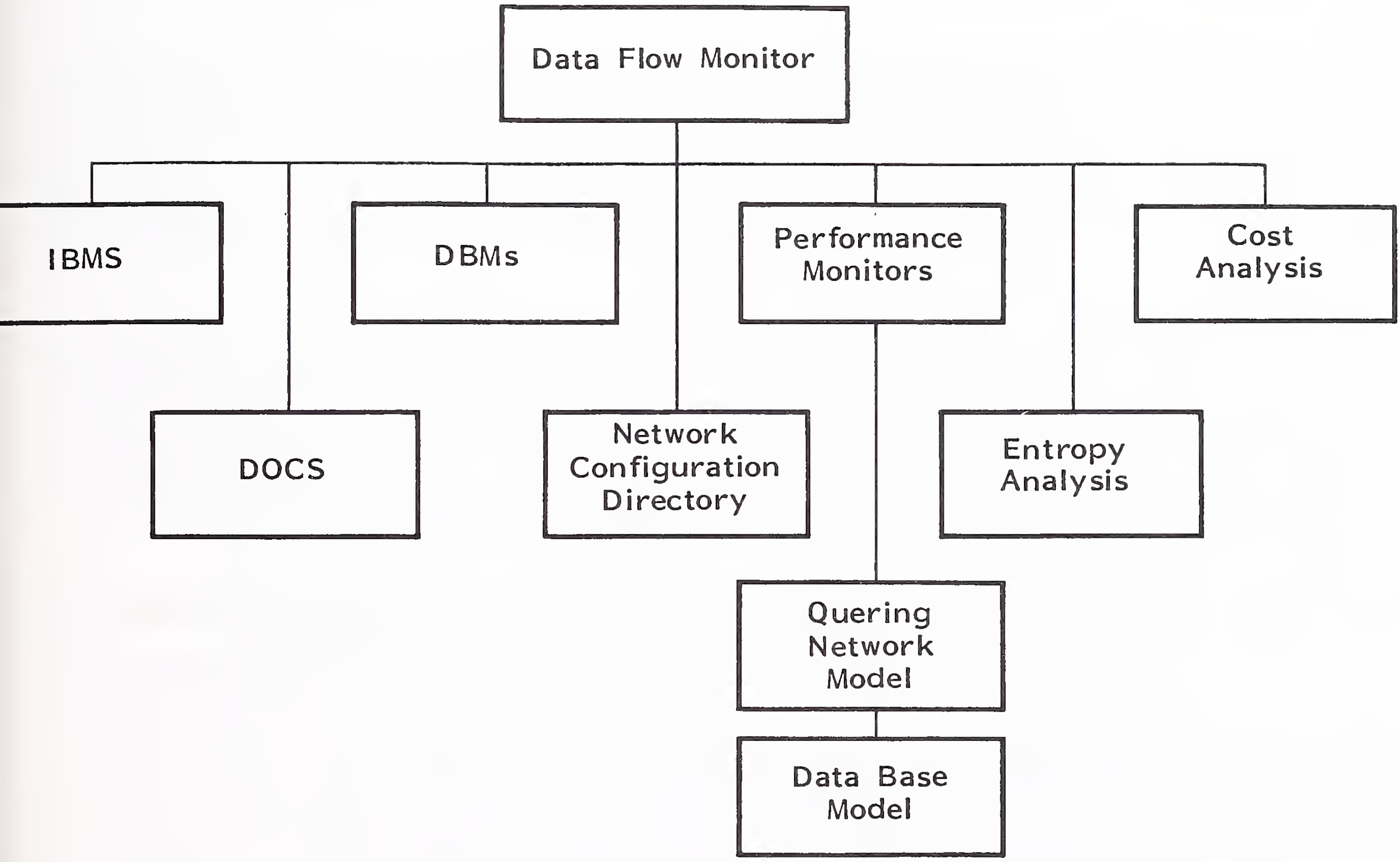
- The purpose of the DFM would be to analyze authorized requests for data/information (protection and security will be isolated in a separate system) either upon request or in anticipation of network performance problems, and to accumulate data/information flow statistics for analysis. Implementation of a DMF could vary from the very simple to the extremely complex.

  - Simple decision rules could screen out impossible requests, for example, you don't send a gigabyte data base to an intelligent workstation even if the requester is authorized to access the entire data base because it might be physically impossible.

  - Either the central processing required or the communications capacity required might be considered cause to reject a request based on anticipated impact. For example, performing a JOIN on relational tables beyond a certain size might be prohibited (in fact, building relational tables beyond a certain size might be prohibited), or single requests for

EXHIBIT C-2

A DATA FLOW MONITOR



INPUT
MSSP

data/information might be screened based on the capacity of the communications link.

- A further level of refinement might anticipate an error (or naive request) on the part of the requestor based on the volume or cost of the data/information requested.

- The statistics gathered by DFM are for use in both refining the protection-security system and in permitting management analysis for information flow and organizational studies. The level of statistics gathered could be geared to various levels from micro-organizational levels down to the individual.

- In addition, the statistics would be essential for network planning and control. In other words, the network configuration directory is really a model of the total hardware/software network which is subject to operational analysis for purposes of reconfiguration.

• The purpose of the DFM are to assure that unworkable (or unnecessary) systems do not evolve in the DSD environment. Conventional hardware and software performance monitors and accounting systems will be essential in implementing DFM, but new tools are also necessary. Some tools are beginning to emerge in work which is being done in artificial intelligence, and some are already available in the more pragmatic work which has been done in operations research. However, there is no question that creative adaptation and even invention are required for quality control in the DSD environment, which truly represents both challenge and opportunity.

4. OPERATING RESEARCH AND ARTIFICIAL INTELLIGENCE

• There is a connection between operations research, artificial intelligence, and security/protection which threads it way back to Great Britian during World War II and specifically to Alan Turning.

- The famous Turning test is still said as a measure of machine intelligence, and becomes especially appropriate in complex computer/communications environments.

- The term operations research was developed in combatting German U-boats in the North Atlantic.

- And, Turning was a key figure in developing the hardware necessary to break the German "Enigma" codes and consequently "read the mail" of the German communications network. Nowhere was this more effective than in combatting German naval operations.

● Since World War II operations research has taken a rather pragmatic approach to many problems associated with industrial engineering, artificial intelligence (until recently) became an academic research area, and security/protection has become a matter of substantial concern associated with both private and public data bases. It is only in the current environment of emerging expert systems that the connection between operations research and artificial intelligence is being established (or at least considered). Specifically:

  - The break between algorithmic and inference-based solutions to complex problems has resulted in an either/or mentality which appears to be reaching a point where it could be detrimental to both disciplines. The better informed practioners (on both sides) are beginning to understand the need for communication between OR and AI, but it is probable that the rift will result in serious problems with some early expert systems.

  - The important point is that elaborate expert systems of questionable value may be developed where the proper application of existing tools of operations research would provide better solutions; and/or, the tools of operations research will freeze "solutions" to problems which could benefit from the analysis and flexibility inherent in the development of knowledge-based expert systems.

- It appears that even DFM will require the proper application of tools from both OR and AI; and in fact, application of OR and AI tools may in themselves require a DFM.

- There has been a gradual recognition that the work done by OR researchers on queueing networks has application in resource allocation (performance monitoring) of both operating systems and computer/communications networks. One of the problems of acceptance was that the example OR researches used for queueing networks was a highway with multiple on and off ramps (how pragmatic can you get) and the problem of a CPU with multiple I/O devices was not readily apparent to computer scientists. Currently, application of queueing network theory to local area networks (LANs) is becoming apparent.

- A general analysis of queueing networks was contained in What Can Be Automated? (The Computer Science and Engineering Research Study) MIT Press, 1980), and a portion is quoted here because it has significance for the OR/AI interfacing problems which INPUT anticipates:

  - "There has been remarkable agreement whenever networks are used to predict device utilizations and throughputs; errors seldom exceed 5%. Network models are less reliable as predictors of queue length and waiting time; but even then, an error rate of less than 25% is common. This agreement is even more remarkable when we realize that the queueing theory seems to rely on the assumption of exponential service-time distributions in each system device, something that rarely happens in practice. The success of research in these models leads us to the conclusion that it is better to use a model whose structure is accurate and whose service-time distributions are approximate than to use a model whose structure is approximate and whose service distributions are accurate. In other words, more errors are introduced by approximations in the structure of the models than are introduced in the distributions.

- Under any circumstances, it is INPUT's opinion that the application of queueing network theory can make a substantial contribution to many of the functions associated with the DFM.

- On the other hand, problems of entropy in both data and information are not clearly understood.

  - Entropy is higher on large, flexible data bases, and it is assumed that more processing power (energy) is required to maintain quality. For example, it is apparent that a large data base employing the relational model has high entropy.

  - Rearrangement of the same data in many different formats (distribution of data bases) increases entropy.

  - Distribution of the same information in many forms increases entropy.

  - The more modes (whether hardware, software, or human) data/information flow through in a communications network the higher the entropy of the network.

  - To the best of our knowledge, effective models to measure data/information entropy do not currently exist. There are needs for analysis and control mechanisms at all levels in data/information networks. Research is required in many areas before practical tools can be developed to address all of the problems of entropy, but some progress can be made with better and expanded knowledge of information theory. Giving focus to the problem by establishing even the most rudimentary analysis tools to measure entropy is essential and this would be an objective of the DFM.

- It also occurs to INPUT, that workable information systems networks are being designed by "experts" who intuitively know that you don't dump massive reports on top management, and you don't filter essential operational information through excessive levels of management and expect to have effective decision-making at the top. There is a need to extend decision support systems to knowledge-based systems, and if this is to be done it must be done with an understanding of data/information entropy. The productivity tools and aids mentioned thus far (IBM, DOCS, and DMF) are all designed to contribute to the general knowledge base from which specific expert systems can be developed.

- There is an important paradox in all that has been described above, the tools of OR and AI appear to be essential in developing tools and aids to control quality in the DSD environment. However, the very OR and AI tools required may result in quality control problems of their own; especially in the area of performance. Hans J. Bremermann in his paper on "Complexity and Transcomputability" (The Encyclopedia of Ignorance, Pergamon Press, Ltd. 1977) points out that both operations research and artificial intelligence frequently require computational algorithms (OR) and searches through exponentially increasing alternatives (AI) which exceed the capacity of any computing resource on earth. In fact some OR and AI "solutions" can easily exceed the capacity of any computer which can even be built, and this is referred to as being transcomputable. Bremermann describes the problem as follows:

    - "We call an algorithm transcomputable if its computational cost exceeds all bounds that govern physical implementation of algorithms.

    - "It can be shown that the exhaustive search algorithms for chess are transcomputable. The same is true of many algorithms of artificial intelligence and operations research. In fact, any algorithm whose computational cost grows exponentially with a size parameter n is transcomputable for all but the first few integers of n.

- "This is a rather disturbing thought and many people have chosen to ignore it."

- Therefore, the only advice which seems appropriate when developing necessary quality control tools and aids which employ OR and AI is to proceed with caution, but by all means proceed.

5. SECURITY, PROTECTION, AND PRIVACY (SPP)

- Everyone knows that security and protection of both public and private data bases present problems which will only be compounded in the DSD environment. Any system which is developed without proper attention to these problems runs high risk of either being inoperable or subject to replacement. It is INPUT's opinion that even isolated cases of harassment of private citizens will soon lead to increased attention to the question of privacy, and this has additional ramifications:

  - There is the obvious potential for law suits which will lead to the requirement for some type of guarantee that data bases are secure.

  - Privacy legislation requiring that information access be made available upon request will become more common, and requests for such information by individuals will increase. This will have substantial impact in several areas:

    - It will require a computer based access and control system for paper-based files (similar to DOCS), and give impetus to the conversion to the electronic office.

- Most current data base systems will not be adequate in supplying required access information, and will have to either be replaced or enhanced.

- Many current public data base services may be severely impacted.

- There will be substantial opportunities for expanded security, protection, and privacy hardware/software systems.

- The SPP problems associated with distributed data bases and information flow have been anticipated and substantial research has been done. However, even rudimentary SPP facilities are not currently being provided in most commercially available systems, and are certainly not being incorporated in most systems being developed in-house. The SPP problem is increasing in complexity exponentially and even the linear advances in solutions are not being applied.

- Security hardware/software is going to be a big business for those who understand the problem and can provide even partial solutions which will extend the life of current systems; and at least contain the problems anticipated in the DSD environment. More important, DBMSs and micro-mainframe links which do not provide at least state-of-the-art SPP facilities are not going to find a ready market.

- While it is beyond the scope of this report to even address the current state-of-the-art in SPP (much less present a solution), there are several important structural considerations which become apparent in the DSD environment:

  - SSP in the DSD environment should preferably be separated (isolated) from the various subsystems. For example, a central SSP module should serve IBMS, DOCS, and various DBMSs which operate in a distributed data base environment. This means a centralized system for paper documents, encoded data bases, and even voice messages.

- While specific data base (or operating) systems might continue to incorporate their own SSP systems and procedures (for example, on a local area network), the quality of these specific systems would be a controlling factor in the distribution of data/information. In other words, the SSP facilities incorporated in a DBMS (perhaps DB2) or an operating system (perhaps Unix) could be a limiting factor in the distribution of data from a host system.

- SSP facilities should be as automatic as possible in the DSD environment. This emphasizes centrally controlled encryption and management of access codes and keys. For example, key and codes could be dynamic based on the preference of the local organization or individual. This would permit multilevel and random security interrogation from the central source if that was specified by the user. In other words, the user would be left with responsibility for establishing the level of security he deemed necessary, but implementation would be relatively automatic.

- The complex security problems of information flow in the DSD environment, while not readily solvable by known techniques, are best addressed for purposes of study and research by a central SSP in conjunction with the facilities of the DFM.

6. LANGUAGES

- It should by now be apparent that languages whether they are classifed as first, second, third, or fourth generations are going to proliferate. However, these designations are currently fuzzy at best and INPUT, rather than adopt the standard nomenclature of 4GL (fourth-generation languages), uses FGL (for fourth, fifth or future generation languages). In other words, languages are evolving and whether natural language or icons prevail is not the question. There are going to be a whole range of languages at the user interface, and this will become apparent in the electronic office.

- A real aid to both productivity, and the implementation of the quality control tools and aids which have been described above would be a meta-language which would incorporate the following:

    - A standard representation for various FGLs (in INPUT's sense) which would facilitate:

        - Communications between and among various systems and intelligent workstations, as well as personal computers.

        - The development of new languages at the user interface.

    - The meta-language would also describe communications and operating systems command languages in a standard fashion to assist in tracking data/information flow, and would facilitate the implementation of the quality control tools especially in the performance area.

    - The distribution of development activities to information centers and intelligent workstations could be enhanced to include all language interpretation (into the meta-language), and provide a single language for the receiving system (whether host or distributed system.)

- INPUT believes host systems are becoming either large data base machines or heavy number crunchers. If we assume that the relational model of data will become prominent in the DSD environment, this means that large systems will be dealing with arrays (for heavy computation) and tables (relational data bases). This leads INPUT to project that future large-scale system architectures, after Sierra, will reflect this requirement. For that reason, it would appear this should be considered in the selection of a meta-language. Without a great deal of analysis, the time may be right to consider APL (A Programming Language).