1972

A Planning and Cost Allocation Procedure for Computer System Management

J. Nunamaker

A. Whinston

Report Number:
72-052

A PLANNING AND COST ALLOCATION PROCEDURE FOR

COMPUTER SYSTEM MANAGEMENT

J. F. NUNAMAKER* AND A. WHINSTON**

PURDUE UNIVERSITY

TR 52

JUNE 1972

*Department of Computer Sciences and School of Industrial Administration

**Economics Department and School of Industrial Administration

Key Words and Phrases

Cost Allocation

Pricing of Computer Services

Allocation of Computer Resources

CR Categories:

2.40,2.41,2.43,2.44,3.50,3.52,3.72,4.22

A PLANNING AND COST ALLOCATION PROCEDURE FOR
COMPUTER SYSTEM MANAGEMENT

BY

J. F. NUNAMAKER AND A. WHINSTON

ABSTRACT

The problems of allocating costs of a computer system are discussed, and a procedure is presented to solve these problems. The concept of a Responsibility Center is presented, and cost allocation rules for the operation of the Responsibility Center are developed. The cost allocation scheme influences users of a computer facility to adjust their demands for processing to that level most beneficial to the overall organization in question. Four conditions form the basis for the development of the cost allocation formula: (1) Charges for the use of a joint facility must cover costs; (2) a user's charges are based on the incremental cost caused by the user; (3) the charge is independent of the names assigned to users or ordering of users (if some users cause the same incremental costs, then the user's charges are the same); and (4) if the user changes his requirements and as a result his incremental costs are changed, then the cost allocation is changed appropriately. The costing procedure based on the above four conditions provides a rational way to distribute costs; it allocates greater costs to the user whose alternative costs are greater. A five step planning procedure implements this cost allocation procedure. (1) Statement of long-range and global requirements, (2) detailed statement of requirements, (3) translation of requirements into a design, (4) specification of cost allocation, and (5) determination of the best systems design.

A PLANNING AND COST ALLOCATION PROCEDURE FOR COMPUTER SYSTEM MANAGEMENT

BY

J. F. NUNAMAKER AND A. WHINSTON

Introduction

A recurrent theme in management thinking and writing on the use of computers shows costs of service soaring but the quality not moving commensurately. Often the head of a company's data center lobbies for new equipment to keep up with the technology rather than on the basis of prudent profit-cost calculation. While many company presidents would agree with this remark they in turn would raise the question: What is a basis for making prudent decisions on data processing? The costs of the information system are easily determined; e.g., salaries, leasing charges, imputed costs of equipment, etc., but the value of the information produced is elusive. In fact, rarely is any attempt made to place a value on the services of the information division. With costs the only data, management's concern is easily appreciated.

Beside top management's inability to evaluate the data center there is another related question. Since the costs of the information system are not allocated to each operating division, little incentive is provided for economy in the use of the facility. Except for informal sanctions by top management, the operating divisions use data processing services as if they were free and thus devote essentially no time considering less costly ways to obtain adequate information. In effect the very organizational structure that has been created causes an escalation in costs.

The role of planning and design of a corporate information system is an interactive affair between the Information Processing Department (IPD) and the various line organizations. Each group may have a preferred system that it hopes would be adopted by the entire organization. While the groups may differ about the configuration of equipment most desired, they probably would all agree that a larger system is to be preferred. Therefore, the groups must compete with each other for their share of the corporate treasury, and must cooperate when seeking an enhanced information system. The sequence planning can be laid out in exacting form in a PERT network, from the preparation by line groups of system development proposals through the actual development phase to the shakedown, but how do we know what system to select? How can the corporate group, admittedly ignorant on the merits of the requests for enhancements, assure itself that the final result will be best for the company as a whole? Put differently, how can the top corporate group determine which groups have chosen properly, and which have made unwise decisions? From the point of view of the company, the ability to evaluate will improve the selection processes.

Consider the recent phenomenon of the facilities management company. Typically the computer operation is managed and operated by a specialist company for a prearranged fee. Often the entire staff of the current computer operation is employed by the facilities management company. Since the fee must on the one hand generate a profit for the management company and on the other hand be attractive to the original company, some real improvements must be made. In effect, a gain appears to be made by an organizational change. In determining a fee the original company is forced to specify the kinds and level of service that each of its user groups should have and to compare this with the charge set by the facilities management company. If gains can be made why can't an internal mechanism be set up to achieve these gains? This paper argues that a rational approach to planning and cost allocation of computer facilities can be developed.

Responsibility Center Concept

In the introduction to the paper the problems of planning the design of a computing system were discussed. It was indicated that with no systematic method of evaluating alternate requests for service by the user groups there may easily be inefficiency. Furthermore, it should be pointed out that much research has been devoted to viewing the IPD as a Responsibility Center in terms of allocating computing facilities for a given design.[1,2,3] For example, a 1970 paper of Nielsen[4] has suggested flexible pricing for obtaining a better utilization of a computing center.

Most papers on pricing are concerned with best use of existing equipment. A set of charges is developed for different types of service and types of equipment such as CPU, storage, etc. Utilization improves when the pricing is used effectively. Flexible pricing is essentially a technique for smoothing the demand for computer service over a given time period. The pricing technique applies mainly to a computer utility characterized by a varying mix of many flexible users. In contrast we are interested in developing a planning method for use within a company with a relatively inflexible user group.

Note that pricing as an allocation scheme without the planning approach we are proposing can lead to poor results. Each division will be anxious to avoid high charges but will propose that large computer facilities be installed. As a result it is most important to the organization to state what is expected from the IPD. Ideally, all decisions in a corporation are made on the basis of the objective (or objectives) of the corporation as a whole. A corporation usually has several objectives not all of which can be readily quantified or even explicitly stated. However, let us assume that one objective of the firm is to maximize the present value of the stream of future earnings for n years. A typical mathematical statement[5] of this objective might be:

Let E_t = Earnings in period t

NR_t = Net Revenue from Operations in period t

IC_t = Information Processing Costs in period t

OC_t = Other Costs in period t

μ_t = Discount rate for period t

t = Time period, 0, 1, 2, 3..., n

then $E_t = NR_t - IC_t - OC_t$

and the objective is to maximize E, where $E = \sum_{t=0}^{n} \mu_t E_t$

The formulas are not intended to be comprehensive but rather to show the overall form of the objective function. In practice other factors would be included.

If NR and OC were independent of IC, maximizing E would be equivalent to minimizing IC. However, NR and OC are quite dependent. Increased IC should yield an increase in NR or a decrease in OC or both. We might expect non-linearity, diminishing returns and time delays. To determine optimum values of IC, one must discover the interrelationships among IC, NR, and OC. These relationships are not well understood and warrant their own research. We shall assume in this paper that a Responsibility Center uses estimates of the value of information provided by the user groups.

We now present the outlines of a Responsibility Center, and the next section will introduce specific cost allocation rules for the operation of the center. To create a Responsibility Center we must state the goals or criteria and the constraints which it would impose on the computer center. The Information Processing Department can be viewed as one which attempts to minimize costs for a given set of information requirements which it should supply to the operating divisions. For a given set of physical resources, i.e., programmers and equipment, the requirements may be impossible to satisfy. Some informal administrative adjustments would be needed to provide more resources for computing or reductions in the demands of some groups. Furthermore, in times of a bad economy, arbitrary reductions in the budget occur even where obvious gains can be achieved. The Responsibility Center concept allows the computer center to operate so as to allocate resources efficiently, including computing equipment, personnel and software. The computer center or IPD can grow only so long as the divisions grow.

Planning Process

The success or failure of a computer installation depends on planning, but this is perhaps the aspect most ignored. It is critical to the successful implementation of the cost allocation procedure proposed in this paper that a computer system planning process be used. Let us assume for the purpose of discussing the cost allocation procedure that a planning procedure for computer system management exists. The planning process is briefly discussed to set the framework in which the cost allocation procedure is most useful. The steps in the planning of a computer installation are given in Figure 1 and can be summarized as follows:

1. Long-Range and Global Requirements Phase

   Identification of long-range and global factors for the planning of a computer system. Included here are objectives of the planning effort and the allocation of resources and hence the profitability of the organization.

2. Detailed Statement of Requirements

   Various line user groups in consultation with the IPD specialists translate global requirements into detailed requirements and then state their detailed requirements in a problem statement language described below.

3. Translation of Requirements into a Systems Design

   The problem statements are translated to detailed design specifications and then the best design is selected from the set of feasible alternatives. If there are inconsistencies in the problem statement or if the problem statement is incomplete, we must return to step one or step two. Several approaches[6,7,8,9,10] are discussed that can be used to translate the statement of requirements into a system design with the aid of a systems analyst.

4. Specification of Cost Allocations

   The best design and its related costs along with the alternative costs of each user group are used to allocate costs to the user groups.

5. Determination of the Optimal Computer System

   If the user groups are not satisfied with their cost allocation or systems design, they revise their requests based on consultation with the IPD. The revised requirements are then returned to step one, step two or step three and the process repeats. When the user groups are satisfied with their cost allocation, then the process stops.

With respect to the considerations mentioned in step 5, the user must be able to evaluate the value of the information and services provided. However, the value of more accurate information or more detailed information is not easy to determine. Judgement of information's value must rest with the individual department managers in an organization. The value of information cannot be determined by the manager of the computing center. Clearly, it is impractical to measure this value precisely for to do so would necessitate comparing organization performance in actual business situations both with and without the information system.

One approach to estimating the value of such information would require each department to construct an information budget. This budget would consist of estimates of the value of information (e.g., how much would it be worth to the production department to know that the actual demand for product X is exceeding previous estimates by 50%). Even this "simple minded" approach gives some estimate of the value of information. The critical issue is whether or not management would accept its own estimates as justification for the design of an information system. Other departments (marketing, production and sales, for example) depend on forecasts, and the Information Processing Department should not be treated any differently.

The steps associated with the allocation of costs are discussed in detail in later sections, and an example problem is presented to illustrate the concepts involved. Steps 1, 2, and 3 in the design process are covered only briefly in this paper since details can be found in other papers.[6,11,12] However, to achieve a successful corporate system, the complete planning and systems design process should be understood, and planning is an important prelude to the costing procedure.

Long-Range and Global Requirements Phase

The planning process for a computing facility starts with the identification of long-range and global requirements. Long-range and global planning is defined as the process of setting formal guidelines and constraints for the level of computing services management desires for the organization. It also involves the

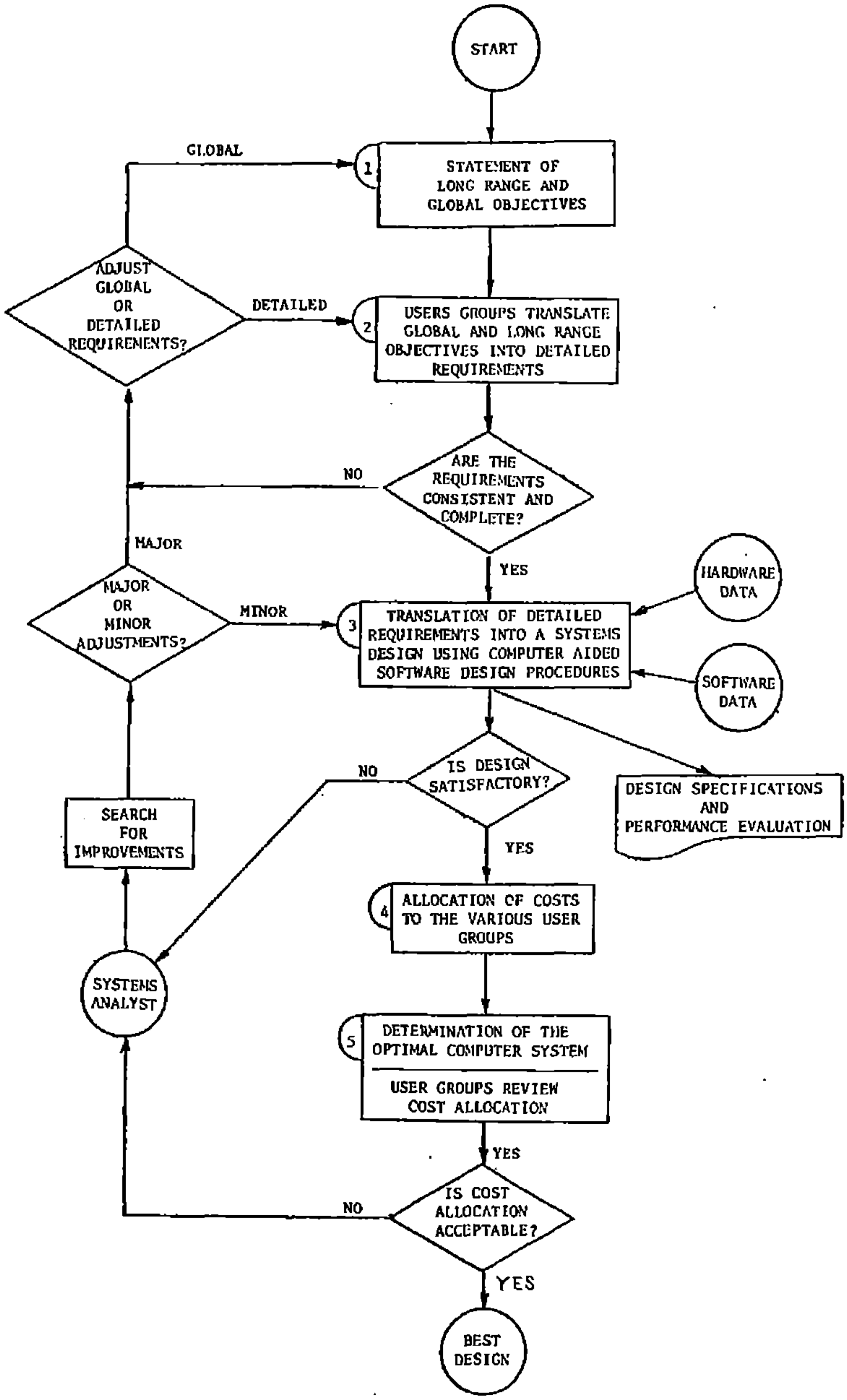

Figure 1 Planning and Cost Allocation Procedure

definition of the estimates of user group needs typically arrived at through feasibility studies.

Next, the goals and objectives of the organization are translated into an elemental set of system building blocks. This set of building blocks is quite different from the set of detailed requirements described in the next phase. The long-range planning requirements must include "early warning" sensitivity to business trends, identify problem areas and identify trends. A discussion of this phase can be found in Blumenthal[11] and Rothery.[12]

Detailed Statement of Requirements

The detailed data processing requirements of the organization must be stated so that a system analyst can translate them into a design. We suggest the use of a "problem statement language". The term "problem statement language" is used here to mean a language which expresses the requirements for an information processing task without specifying a procedure to accomplish the task. The language is used to capture the requirements of each department. The language, while precise, is easily understood by management personnel. Thus management can make key decisions during the design phase of the information system. Numerous papers proposing and discussing such languages have either been published or been distributed as working papers.[13,14,15,16,17,18,19,20] The specifications for a Problem Statement Language are currently being developed for the ISDOS project by Daniel Teichroew[21] at the University of Michigan. Under the sponsorship of the ISDOS project a skeletal problem statement language called SODA/PSL (Systems Optimization and Design Algorithm/Problem Statement Language) was implemented by Nunamaker.[6]

Stating a problem without stating a procedure to solve it differs from the present ad hoc techniques used to state information processing requirements of an organization.[22] One deficiency of present approaches to problem statement is that the problem statement is specified with an implied design procedure. A problem definer is of course not aware that he has biased the design, but in fact he has made major decisions that constrain the actual systems design activity. He influences file organization, program structure and program sequencing without realizing they are consequences of the manner in which he has stated the problem. The problem definer using any of the techniques or languages referenced in the previous paragraph avoids some of these biases because the processing requirements are stated nonprocedurally. The nonprocedural approach enables the problem definer for the user group to define completely the requirements of the information processing system without specifying the procedure required to complete the task. Clearly, the procedures for accomplishing a task are not required to be part of the problem statement. These procedures can be specified at a later stage in the design process when the total system requirements are available. The advantage is that the system design is not bound too early in the activity.

Translation of Requirements into a System Design

An Information Processing System is a set of hardware, software, personnel, and procedures assembled and structured to accomplish given data processing requirements in accordance with given performance criteria.

After the problem has been stated the requirements are translated into specifications for a systems design. The statement of requirements, equipment availability and constraints (such as the existing system) are considered in the design phase. The design phase produces specifications for the five major parts of the IPS:

- Hardware that will be used
- Software packages that will be used
- Programs to be written
- Operating Schedules to sequence the running of the programs
- Data Organization to specify the data base and the file structure

First it is necessary to generate alternative designs and then to evaluate them. There are two approaches:

1. Manually
2. Interactively (man-computer)

Manual methods for generating alternative systems design and evaluating performance are often made heuristically. The problem becomes more acute as the system becomes more complicated. It is becoming increasingly difficult to justify a manual approach to the problem. It used to be fairly easy to determine how a particular configuration would handle a work load. The systems analyst gathered data on processing flow, file sizes and computer characteristics and calculated the processing time by looking at device speeds and instruction times. As systems become increasingly complex, the analyst must use the computer more and more for assistance in the decision-making process.

This decision making problem can be handled interactively using software packages such as SCERT[7] (Systems and Computers Evaluation and Review Techniques), CASE[9] (Computer Aided System Evaluation) and SODA[6] (Systems Optimization and

Design Algorithm). SCERT, CASE and other design simulators,[8,10] represent a family of computer programs used to simulate the performance of users' processing requirements against cost/performance models of selected configurations. The man-machine procedures emphasize performance evaluation. However, systems analysts are still needed to generate the alternative designs to be evaluated and someone must compare the results. A systems designer is also needed to interpret the results and to spot areas where improvements can be made.

SODA differs from SCERT and CASE in that it does not involve simulation as a solution technique. SODA requires less manual interaction than the system simulators but has other restrictions.

However, SODA limits the alternatives generated. SODA is presently restricted to the design of uniprogrammed batch systems, sequential auxiliary storage organization, the specification of linear data structures, and the selection of a single CPU. The model is deterministic and is just a start towards the complete automation of the systems design function.

These software packages (SCERT, CASE, SODA) provide a methodology for making a choice of options relating to alternative hardware configurations. Our treatment of the cost allocation scheme requires that we specify the costs for a large number of alternative designs. The Computer Aided System Design procedures are presented as one way of providing these alternative cost estimates of an Information Processing System. However, the ability of any of these procedures to make the necessary design decisions depends upon the initial statement of requirements.

The discussion of steps one, two and three is presented in order to emphasize the importance of the planning process in the cost allocation procedure.

Allocation of Costs

We first develop the ideas theoretically and later apply them to a concrete example. Let $N = \{1,2,\ldots n\}$ denote the set of user groups in the company. Each user group should be a well defined unit with budgetary authorization for computer services. Further let us denote by a multidimensional vector

$$K_i = \begin{pmatrix} k_{i1} \\ \cdot \\ \cdot \\ \cdot \\ k_{iS} \end{pmatrix}$$ the level of computer service requested by

the i^{th} user group where there are S classifications of computer services.

For example, we might let k_{i1} represent the number of CPU computing units required by the i^{th} user, we might let k_{i2} represent the number of input/output units required by the i^{th} user, and we might let k_{i3} represent the total number of computing units (CPU computing units plus input/output units) required by the i^{th} user, etc. Let $C_K(N)$ be the total cost of the system serving the N users and having overall requirements designated by K, where

$$K = \sum_{i=1}^{n} K_i$$

and the elements K and K_i are vectors of S components.

The cost $C_K(N)$ is our prime concern and must be allocated among the users of the system subject to their budgets and needs. Therefore, the following questions should be answered:

1. For a given value of K how should $C_K(N)$, the total cost, be allocated among the user groups?
2. How do we determine a desirable value of K in terms of the corporate needs and budgetary restrictions?

We consider question one first and then show how the solution to one gives the basis for resolving question two.

To motivate our approach to the problem let us assume that n = 2 and S = 1, i.e. there are two users who have decided on a facility with capacity measured as 1002 computing units for a given time period. The cost can be broken down as \$100 fixed cost (rental cost) and a cost of \$.50 per unit of usage. For this example, as shown in Figure 2, we could have the following diagram.

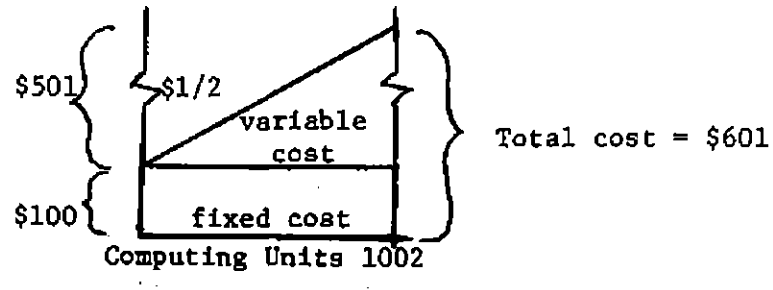

Figure 2. Computer Facility with Two Users

One way of dividing the costs of such a system would be to assign to each user the added or incremental cost which result from his introduction to the system. Thus suppose user one wants 2 units of computing and user two 1000 units. The incremental or added cost for user one is seen to be 101 if he were the "first" user. This results from the payment of $100 fixed cost and $1 variable cost ($.50 * 2 units) for the two computing units. In this situation, user two would have costs of $500 ($.50 * 1000 units). If user one were "second" then his incremental cost is $1, the added cost of using two units since the rental cost of $100 has been assigned to user two. User two then has a total cost of $600. Eitherway, the total cost of the system for these two users is $601 ($100 fixed + 1002 units * $.50). It is assumed that each user is equally likely to use the system first and thus have a probability of 1/2 of incurring the $100 fixed cost. The important point to note is that in attempting to define the concept of incremental cost for a user it depends on how the increment is computed. Where no natural order is assignable we can take the possible orderings as equally likely. Thus the average costs for user one may be calculated as follows:

where FC = Fixed Cost

VC_i = Variable Cost i^{th} user

TC = Total Cost

K_i = Demand of the i^{th} user for service

$C_{K_i}(i)$ = Cost for demand K_i to user i

The costs for user one and user two are computed from the following expressions:

$$C_2(1) = 1/2\ (FC + VC_1) + 1/2(TC - FC - VC_2) \quad (1)$$

$$C_{1000}(2) = 1/2(FC + VC_2) + 1/2(TC - FC - VC_1) \quad (2)$$

Note that $C_{K_1}(1) + C_{K_2}(2) = TC$ is an identity independent of the particular values of the fixed and variable costs. Therefore expressions (1) and (2) through substitution reduce to (3) and (4) respectively.

$$C_2(1) = 1/2\ (FC) + VC_1 \quad (3)$$

$$C_{1000}(2) = 1/2\ (FC) + VC_2 \quad (4)$$

The average cost for user one is:

$C_2(1) = 1/2(100) + 1 = \$51$

$C_{1000}(2) = 1/2(100) + 500 = \$550$

This formula, while developing an incremental cost approach to the problem, still has an important defect. The fact is that user one is a victim of economies of scale. User one may in fact be assessed a higher charge by this approach for his 2 computer units than if he were able to arrange this same service independently. User one would probably be able to go to an outside source and get his computing service more cheaply. Since the firm undoubtedly would like the in-house computer to be used for all computing, an undesirable situation arises. This factor must be considered in our approach. Suppose user one has access to an outside computing source, a system with a fixed cost of \$4 and a variable cost of \$1 per computing unit, while user two has access to a facility with a fixed cost of \$99 and a variable cost of \$.50 per computing unit. The alternative cost characteristics for user one and user two are summarized in Table 1:

Alternative Costs for:	Fixed Costs	Variable Cost	Computing Units	Alternative Cost
User 1	\$ 4	\$1.00	2	6
User 2	\$99	\$ .50	1000	599

Table 1. Alternative Cost Characteristics for User 1 and User 2

Again, the average costs must be found, however, in computing the cost allocation we will account for the fact that if user one is assigned the first increment then he would use the smaller computer facility with the above cost configuration.

Thus each user can be assigned at most his alternative cost. Let $C_2(1)$ and $C_{1000}(2)$ again denote the cost allocations. We have:

AC_i = Alternative Cost to the i^{th} user for an alternative source

TC = Total Cost of the in-house computer system

The costs for user one and user two based on alternative costs are computed from expressions (5) and (6)

$$C_2(1) = 1/2\ (AC_1) + 1/2(TC - AC_2) \qquad (5)$$

$$C_{1000}(2) = 1/2(AC_2) + 1/2(TC - AC_1) \qquad (6)$$

Note that $(TC - AC_i)$ implicitly assumes that $TC \leq \sum_i AC_i$.

$$C_2(1) = 1/2(\$6) + 1/2(\$601-\$599) = \$4$$

$$C_{1000}(2) = 1/2(\$599) + 1/2(\$601-\$6) = \$597$$

Thus, in defining the cost added by a user, the proper concept is the incremental cost with respect to the best alternative available to the user. This can be clearly shown by comparing the costs illustrated in Table 2. Observe that if either user has a more attractive alternative cost then the allocation of costs cannot be made.

Costs for:	Method based on sharing fixed costs	Alternative Costs	Method based on incremental cost
user one	$ 51	$ 6	$ 4
user two	$550	$599	$597

Table 2: A comparison of 3 approaches to cost allocation for user one and user two.

We now generalize the cost allocation formula to the case of n users whose participation may be ordered in n! ways. We assume all users are participating and all orderings equiprobable. Again we let K be the total user demands of K_1, K_2, ...K_n. G is the subset of g members of N and C(G) is the cost of operating a computing facility for the members of G. Note that the cost C(G) is predicated on supplying the total service $\sum_{j\varepsilon G} K_j$ and the cost C(G-i) is based on supplying $\sum_{j\varepsilon G-i} K_j$.

The weighting factor F_g on the additional cost due to user i when his demand is ordered in the g^{th} position is given by

$$F_g = \frac{(n-g)!\ (g-1)!}{n!}$$

In computing F_g we assume that all n users are always participating. This coefficient F_g is derived from the fact that when the i^{th} user's demand is considered to occur in the g^{th} position there are (g-1)! arrangements of those whose demands occur before his and (n-g)! arrangements of those whose demands occur after his. These arrangements can be viewed as follows for n = 11 and g = 4, i.e., the i^{th} user is in position 4.

$$\underbrace{xxx}_{g-1}\ \ i\ \ \underbrace{xxxxxxx}_{n-g}$$

F_g takes into account all possible orderings of the users based on the assumption that any ordering of user arrivals is equally likely. We wish to determine the average incremental cost attributable to user i. For any ordering of the n users, let G be the set of all users up to and including user i. The incremental cost attributable to user i is C(G) - C(G-i). How many of the n! orderings produce exactly this incremental cost? There are (g-1)! ways to order the i^{th} user's predecessors and (n-g)! ways to order the i^{th} user's successors.

The general expression for computing the cost for the i^{th} user based on an incremental cost approach is given by

$$C_{K_i}(i) = \sum_{i \in G \subseteq N} \frac{(n-g)!\ (g-1)!}{n!}[C(G) - C(G-i)] \qquad (7)$$

To clarify the form of the general formula consider the case of N = {1,2,3}. The index set G ranges over all subsets of N which contain the i^{th} member of the set. For $C_{K_1}(1)$ the index set G takes on the values {1}, {1,2}, {1,3} and {1,2,3}.

Thus we have:

$$C_{K_1}(1) = \frac{(3-3)!\ (3-1)!}{3!}[C(1,2,3) - C(2,3)] + \frac{(3-2)!\ (2-1)!}{3!}[C(1,3) - C(3)]$$

$$+ \frac{(3-2)!\ (2-1)!}{3!}[C(1,2) - C(2)] + \frac{(3-1)!\ (1-1)!}{3!}[C(1)].$$

G identifies which users make up the groups that are participating in computer services and F_g takes into account all possible orderings among predecessors and successors.

Another way of viewing the determination of the cost allocation formula is to consider a set of conditions or postulates which imply this formula. These conditions can be thought of as giving a fair rule for the cost allocation. If the user groups accept the rules, they must then accept the cost allocations implied. We give below the following four necessary conditions which determine the cost formula we are using.

1. Charges for the use of the joint facility must cover costs.
2. A user's charges are to be based on the incremental cost caused by him.
3. The charge is independent of the names assigned to users. If some users cause the same incremental costs, then the user's charges are the same.
4. If the units of measurement of incremental costs are changed or if the user changes his requirements and his incremental costs are changed, then the cost allocation is changed appropriately.

Consider the following situation for condition number 2. The user that claims in his statement of requirements that he must have access to a "complete" Data Management System should pay the incremental costs of the main memory and other resources required for this service. Quite often, however, the user that runs batch COBOL jobs has to share the costs of the increased resources even when he doesn't need them.

Company Y Example

Let us consider as an example a company consisting of 4 user groups. Company Y is a wholesaler of parts and the activities of the user groups are as follows:

User Group A - Receiving, Warehouse, and Shipping Operations

User Group B - Purchasing and Accounts Payable

User Group C - Payroll, Management and Internal Audit

User Group D - Sales and Accounts Receivable

The incoming transactions and outgoing documents of the four user groups consist of customer orders, payments from customers, bills of lading, vendor invoices, purchase orders, payments to vendors, sales reports, inquiries, etc., all on different schedules.

The Company Y example was solved using SODA. The data processing requirements for all the departments in the Company Y example were stated in the SODA/Problem Statement Language. Then, the statement of requirements was translated into alternative system designs by SODA.

The various hardware and software alternatives are given below:

Annual Costs for Configurations Available: ($000)

CP - Central Processor Unit and CM - Core Memory

There are three central processors (CP_1, CP_2, CP_3) and a total of six main memory sizes available for selection. The main memory available and cost for each CP and CM combination is given below:

	CM_1	CM_2	CM_3	CM_4	CM_5	CM_6
CP_1	15	22	33	48		
CP_2		31	42	57	72	90
CP_3				68	83	101

P - Peripherals and Auxiliary Memory

There are seven peripheral and auxiliary memory configurations ($P_1 \ldots P_7$) available for selection. Peripherals include readers, printers and punches. Auxiliary memory includes some combination of tape units, disks and drums.

Nunamaker and Whinston, Cost Allocation

P_1	P_2	P_3	P_4	P_5	P_6	P_7
37	43	49	57	69	81	103

S - Software and Operations Support

Software and Operations Support includes programmers, system analysts, operations personnel, software systems support and facilities costs. There are eight levels of Software and Operations Support available for selection.

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
48	63	103	136	173	189	209	239

In this example, we assume no inexpensive external alternatives exist, so that each user group sees as its alternative cost the least cost solution made by SODA if the user group sought services alone. However, in actual practice one would have to consider alternative sources from outside the company as well as in-house alternatives.

The annual costs for the various alternatives for each user group combination were generated by SODA and are summarized in Table 3 as follows:

Group Number	User Group(s) in G	Configuration Components	C(G)	Group Number	User Group(s) in G	Configuration Components	C(G)
1	(A)	$CP_2, CM_4, P_4, S_5 =$	287	9	(B,D)	$CP_2, CM_4, P_4, S_4 =$	250
2	(B)	$CP_1, CM_2, P_2, S_2 =$	125	10	(C,D)	$CP_1, CM_4, P_3, S_3 =$	200
3	(C)	$CP_1, CM_1, P_1, S_1 =$	100	11	(A,B,C)	$CP_2, CM_5, P_5, S_8 =$	372
4	(D)	$CP_1, CM_1, P_2, S_2 =$	118	12	(A,B,D)	$CP_3, CM_4, P_6, S_8 =$	388
5	(A,B)	$CP_2, CM_5, P_5, S_7 =$	350	13	(B,C,D)	$CP_2, CM_4, P_5, S_5 =$	299
6	(A,C)	$CP_2, CM_4, P_5, S_6 =$	315	14	(A,C,D)	$CP_2, CM_5, P_6, S_8 =$	392
7	(A,D)	$CP_2, CM_5, P_5, S_7 =$	350	15	(A,B,C,D)	$CP_3, CM_5, P_6, S_8 =$	$403
8	(B,C)	$CP_1, CM_4, P_3, S_3 =$	200				

Table 3. Annual Costs for all User Group Combinations

The number of different sets of groups for the Company Y example is $2^4 - 1 = 15$. The number of different orderings is $4! = 24$.

Table 3 provides all the data required to compute $C_{K_i}(i)$ from (7).

The weighting factors are:

g	F_g-Weighting Factor
1	1/4
2	1/12
3	1/12
4	1/4

Consider all of the possible orderings of the user groups in G given in Table 4 and all the possible orderings of user group A in Table 4.

G	g	incremental cost for A [C(G)-C(G-A)]	Possible ordering of 4 users	
A	1	287	ABCD ABDC ACBD ACDB ADBC ADCB	6/24 = 1/4
AB	2	350 - 125 = 225	BACD BADC	2/24 = 1/12
AC	2	315 - 100 = 215	CABD CADB	2/24 = 1/12
AD	2	350 - 118 = 232	DABC DACB	2/24 = 1/12
ABC	3	372 - 200 = 172	BCAD CBAD	2/24 = 1/12
ABD	3	388 - 250 = 138	BDAC DBAC	2/24 = 1/12
ACD	3	392 - 200 = 192	CDAB DCAB	2/24 = 1/12
ABCD	4	403 - 299 = 104	BCDA BDCA CBDA CDBA DBCA DCBA	6/24 = 1/4

Table 4. Possible Orderings of User Group A

Nunamaker and Whinston, Cost Allocation

The joint cost for $C_{K_A}(A)$ is computed as follows:

$$C_{K_A}(A) = 1/4(287) + 1/12(350 - 125) + 1/12(315 - 100) + 1/12(350 - 118) + 1/12(372 - 200)$$
$$+ 1/12(388 - 250) + 1/12(392 - 200) + 1/4(403 - 299) = \$196.$$

The weighting factors for the individual cost components in $C_{K_A}(A)$ are derived from enumeration of all the possible orderings of the user groups in G.

Consider the cost component $1/12(C_{AD} - C_D)$. There exists only two orderings in which user group A is second and user group D is first out of 24 possible orderings of two groups as shown in Table 4. Thus, the weighting factor of 1/12.

Also, consider the cost component $1/12(C_{ABC} - C_{BC})$. There exists only two orderings in which user group A is third and user groups B and C are either first or second in position out of 24 possible orderings of three groups as shown in Table 4. Thus, the weighting factor of 2/24 or 1/12.

The joint cost allocation computed from (7) is as follows:

	Joint Cost	Independent Alternative Cost
User group A	196	$287
User group B	76	125
User group C	51	100
User group D	80	118
	403	$630

The joint costs based on the incremental approach are then compared to the independent alternative costs.

The total annual cost for user groups A, B, C, and D going in together to obtain data processing services is $403,000. The cost of user groups A and B going together is $350,000 and the cost of user groups B and C going together is $200,000. This solution (A,D) and (B,C) with two Information Processing Departments would cost Company Y $550,000.

Determination of the Optimal Computer System

Step 4 in our outline of the planning system called for the revision and re-evaluation of requirements by the user groups. We assume that each group, faced with the costs of its particular set of requirements, will, acting in its own interest, determine whether the returns it perceives are adequate to justify the costs. In the previous section we assumed a given set of requirements for a computing facility and allocated costs. In this section we allow for revision of the requirements. For illustration purposes we again consider two users of a computer center. Each user i produces an information output y_i yielding a return $\pi_i(y_i)$. (Note that the user group must be able to evaluate the worth of the information and services supplied.) We want to show that the cost allocated to one user depends on the amounts of service to other users. We write $C_{K_1 + \hat{K}_2}$ (1)

to refer to the cost allocation of user one where his requirements are represented by the vector K_1 and user two is represented $\hat{K}_2$. For a given K_2, user one will decide on how much computer service is best for him. Thus, given user two's demand $\hat{K}_2$, user one determines his optimum demand by solving the problem

$$\max_{K_1, y_1} \pi_1(y_1) - C_{K_1 + \hat{K}_2} (1)$$

Subject to: $Z_1(y_1) \leq K_1$

where $C_{K_1 + \hat{K}_2} = 1/2(C(1)) + 1/2 \left[C(1,2) - C(2)\right]$ is user one's incremental

cost as defined above. The constraints $Z_i(y_i)$ relate the requirements of computer service to the information output y_i. This formulation makes clear the nature of the interconnections. Note that we have altered slightly the notation for the allocations to emphasize the dependency on the requirements vectors of the different user groups.

Similarly, given user one's demand $\hat{K}_1$, user 2 solves

$$\max_{K_2, y_2} \pi_2(y_2) - C_{\hat{K}_1 + K_2}$$

Subject to: $Z_2(y_2) \leq K_2^T$

Economies of scale may be expressed by $C_{K_1 + K_2}(1,2) \leq C_{K_2}(2) + C_{K_1}(1)$ so that the more one user demands, the less will be the costs to the other user.

Notice that the output of each is restricted by the amount of capacity that is demanded and paid for. In effect, by agreeing to pay the full costs due to this capacity, user i receives rights to K_i, i.e., he is guaranteed that he can demand at least K_i units of computer service at the agreed on cost.

By solving the above problem, for each value of $\hat{K}_2$, user one has an optimal value of K_1. A locus of these is obtained assuming that K_1 and K_2 are scalar and is shown in Figure 3.

Figure 3. Optimal Value for User One

K_1^o is the optimum capacity that user one would use alone if $\hat{K}_2 = 0$. K_1 increases with $\hat{K}_2$ since computing per unit becomes cheaper to user one the more user two demands. (For the same reason, net returns for user one are also increasing with increasing K_2.) Assuming $\pi_1(y)$ increases at a decreasing rate, the curve has a concave shape.

The same sort of curve is obtained for user two. We may put both of these on the same set of axes. The curve labelled I in Figure 4 corresponds to those points preferred by user one for values of K_2 while the curve II corresponds to those points preferred by user two for values of K_1.

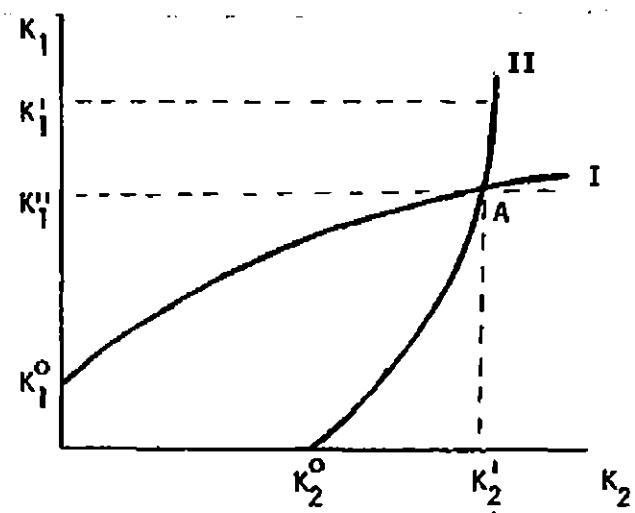

Figure 4. Optimal Values for User 1 and User 2

At the point A, user one is maximizing with respect to user two's demands while simultaneously user two is maximizing with respect to user one's demands. Since there is agreement on the values of K_1 and K_2 at this point, $C_{K_1}(1) + C_{K_2}(2)$ covers total costs by construction. A is preferred by both to points on curves I and II to the left of A. Points to the right of A on curve I are preferable to A for user one; likewise, points to the right of A on curve II are preferred by user two. Thus, points to the right of A on curves I and II would have one or the other user better off. However, in this region if each user paid only his incremental costs, costs could not be covered by payments; for example, at (K_1', K_2') costs would be covered if user one paid for K_1' units but his demand at K_2' would only be K_1''. It would then not be possible to cover total costs of a computing facility of size $K_1' + K_1''$ using incremental costs. The point A therefore gives the greatest capacity and the most net benefits to each user consistent with covering costs if each user pays only his incremental costs. Thus the point A, if it could be found, would be agreed on by the two users if no further cooperation is allowed between them. For this reason we call A the agreement point.

We have presented a cost allocation scheme for a given computer system configuration and extended it to a case where each user group could alter the system requirements. A point where, given the current configuration and the valuations of the information services produced by the configuration, each user feels that he has the best configuration is an agreement point. Recalling the dynamic adjustment process as summarized in Figure 5, we may inquire as to whether the agreement point can be reached.

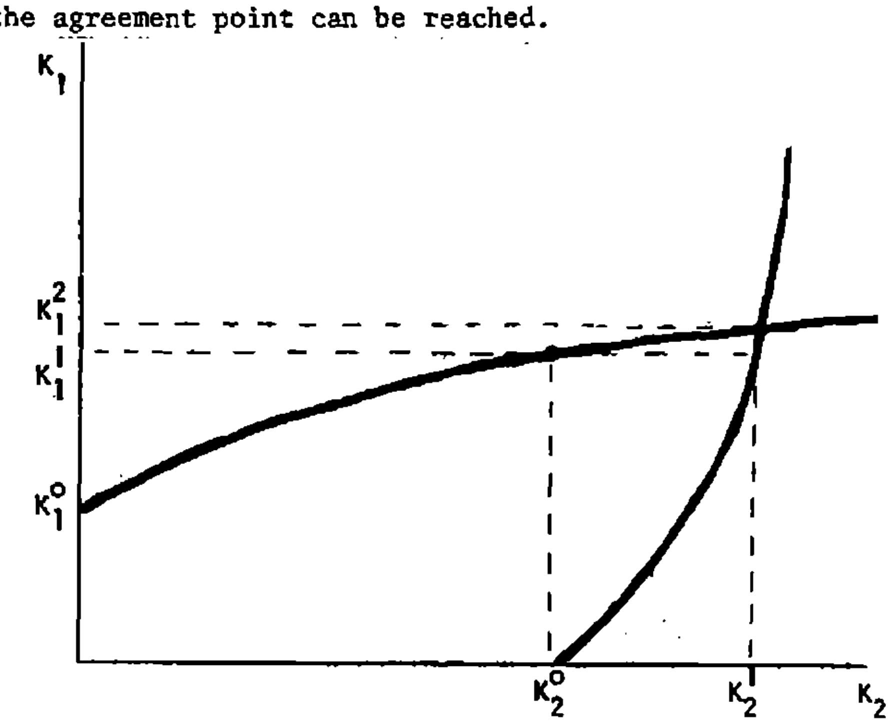

Figure 5 Agreement Point for User One and User Two

To study this question we abstract the planning process and assume that each user is informed of the total configuration presently requested and his own cost allocation. Thus initially, user one is asked his optimum K_1 given $\hat{K}_2 = K_2^{\circ}$. Denote this by K_1^1. Then user two is told K_1^1 and asked for the corresponding value of K_2^1; user one is told K_2^1 and then gives K_1^2, and so on. The concavity of the curves causes the process to converge on the agreement point as illustrated above in Figure 5. In the case of 3 users, for each value of $\hat{K}_2$, $\hat{K}_3$, user one will have an optimum K_1. Thus a surface of points $(K_1, \hat{K}_2, \hat{K}_3)$ is obtained for user one, and similarly 3-dimensional surfaces for users two and three. These surfaces should be concave due to diminishing returns. The surfaces will intersect in a point where costs are covered and each user's demands are maximized given the demands of the others. From the agreement point it is not possible to increase anyone's net returns so that the incremental cost scheme still covers costs. The agreement point is then a vector maximum of net returns under this system of charges given that the costs must be covered.

Conclusion

Allocations of cost to users is a complex problem that has long been overlooked. Many companies install computers on the basis of cost savings but give very little consideration to the allocation of cost. From personal experience it has been observed that most companies claim to have no cost allocation system and therefore no resource allocation systems. But, as was pointed out by Nielsen,[3] "If resource allocation is not done explicitly it is done implicitly; there is no such thing as 'no allocation'. Ignoring the problem will not make it go away; one will only choose an allocation mechanism by default." In most cases the default mechanism is a first come, first served procedure. This is particularly true in situations where there is "no rationing of computing, where everyone can submit as many jobs as frequently as he wishes."

The cost allocation scheme presented in this paper attempts to influence users of a computer facility to adjust their demands (problem statement) for processing to that level most beneficial to the overall organization.

Economies of scale dictate the establishment of large central computer facilities shared by many users. The fact that the activity of one user in such an environment can affect the quantity and quality of service obtained by the others implies that some sort of global control and cooperation is needed. This cost allocation procedure is one form of such control and is directed toward fair allocation to all user groups.

Our procedure provides a rational way to distribute costs; the procedure allocates a larger portion of the costs to the user that would have to pay a proportionately higher amount from an alternate source. This pricing procedure will not only help allocate the resources of a new system or an existing system, but will also form a guideline for any additional purchases for the information system.

References

1. N. Singer, H. Kanter and Arnold Moore, Prices and the Allocation of Computer Time, AFIPS Proceedings, FJCC 1968.

2. S. Smidt, The Use of Hard and Soft Money Budgets, and Prices to Limit Demand for Centralized Computer Facility, AFIPS Proceedings, FJCC 1968.

3. N. R. Nielsen, Flexible Pricing: An Approach to the Allocation of Computer Resources, AFIPS Proceedings, FJCC 1968.

4. N. R. Nielsen, The Allocation of Computer Resources - Is Pricing the Answer? Communications of the ACM, August 1970.

5. D. Teichroew and J. F. Nunamaker, Jr., Design of Information Systems, 10th Annual Conference on Recent Developments in Operations Research, Case Institute of Technology, June 1968.

6. J. F. Nunamaker, Jr., A Methodology for the Design and Optimization of Information Processing Systems, AFIPS Proceedings SJCC, Volume 38, May 1971.

7. D. J. Herman and F. C. Ihrer, The Use of a Computer to Evaluate Computers, AFIPS Proceedings, SJCC 1964, Volume 25, pp. 383-395.

8. L. R. Huesman and R. P. Goldberg, Evaluating Computer Systems Through Simulation, Computer Journal, August 1967, pp. 150-156.

9. The Case for CASE: Computer-Aided System Evaluation, Computer Learning and Systems Corporation, 1971.

10. Digest of the Second Conference on Applications of Simulation, sponsored by SHARE/ACM/IEEE/SCI, New York, New York, December 1968.

11. Sherman Blumenthal, Management Information Systems: A Framework for Planning and Development, Prentice Hall, 1969.

12. Brian Rothery, Installing and Managing a Computer, Brandon Systems Press, 1968.

13. J. W. Young and H. Kent, Abstract Formulation of Data Processing Problems, Journal of Industrial Engineering, November-December 1958.

14. CODASYL Development Committee, An Information Algebra Phase 1 Report, Communications of the ACM, April 1962.

15. Lionello A. Lombardi, Theory of Files, Proceedings of the Eastern Joint Computer Conference, 1960.

16. Lionello A. Lombardi, A General Business-Oriented Language Based on Decision Expressions, Communications of the ACM, February 1964.

17. B. Langefors, Some Approaches to the Theory of Information Systems, BIT, 1963, pp. 229-254.

18. B. Langefors, Information Systems Design Computations Using Generalized Matrix Algebra, BIT, 1965, pp. 96-121.

19. Hugh Lynch, Accurately Defined Systems: A Technique in Systems Documentation, SIG BDP Newsletter.

20. Joseph Kelley, "Time Automated Grid," in Computerized Management Information Systems, MacMillan, pp. 367-403.

21. Daniel Teichroew, E. Sibley, L. Metric and H. Sayani, PSL: A Problem Statement Language for Information Processing System Design, ISDOS Research Project, University of Michigan, February 1969.

22. Daniel Teichroew, William A. McCuskey, J. F. Nunamaker and E. Sibley, Automation of Software Construction, COINS Conference, Miami Beach, December 1969.

23. E. Loehman and A. Whinston, A New Theory of Pricing and Decision Making in Public Investments, Working Paper, Purdue University, August 1970.