DOCUMENT RESUME

ED 153 618	IR 005 885
AUTHOR	Chandler, John S.; DeLutis, Thomas G.
TITLE	A Methodology for Multi-Criteria Information System Design.
INSTITUTION	Ohio State Univ., Columbus. Computer and Information Science Research Center.
SPONS AGENCY	National Science Foundation, Washington, D.C.
REPORT NO	OSU-CISRC-TR-77-14
PUB DATE	Jun 77
GRANT	SIS75-21648
NOTE	36p.
EDRS PRICE	MF-$0.83 HC-$2.06 Plus Postage.
DESCRIPTORS	Designers; *Design Preferences; Evaluation Criteria; *Evaluation Methods; Goal Orientation; *Individual Needs; *Information Systems; Information Utilization; *Performance Criteria; Performance Factors; *Program Design; Use Studies

ABSTRACT

The complexity of the design problem for modern computer based information systems has increased significantly over its predecessors. The problem presented to the designer is to configure a system which satisfies the user criterion while achieving system resource related performance criteria. The purpose of this paper is to present an evaluation technique for addressing the problem of multiple criteria performance evaluation of information systems. The goal of this technique is to provide the designer with an evaluative procedure which: (1) seeks to establish a causal relationship between user goal attainments and system resource expenditures, while allowing for the collection of heretofore hard to obtain evaluation information; (2) measures the impact of individual user classes on internal system performance; and (3) identifies those information system activities which should be modified to improve system performance related to those goals. All of this is achieved by maintaining resource utilization statistics on a user class basis. It is thought that such an evaluation framework is capable of eliminating many of the numerous non-satisfactory designs by directing the designer to the most advantageous ones. (Author/DAG)

OSU-CISRC-TR-77-14

"A Methodology for Multi-Criteria Information System Design"

by

John S. Chandler and Thomas G. DeLutis

Work performed under
Grant Number SIS75-21648
The National Science Foundation

Computer and Information Science Research Center
The Ohio State University
Columbus, Ohio 43210

June, 1977

TABLE OF CONTENTS

PREFACE

This work was supported by Grant Number SIS75-21648 from The National Science Foundation to Dr. Thomas G. DeLutis, Associate Professor of Computer and Information Science, and conducted at the Computer and Information Science Research Center of The Ohio State University. The Computer and Information Science Research Center of The Ohio State University is an interdisciplinary research organization which consists of the staff, graduate students, and faculty of many University departments and laboratories. This report is based on research accomplished in cooperation with the Department of Computer and Information Science. The research contract was administered and monitored by The Ohio State University Research Foundation.

1. Introduction

In March of 1973, ACM and NBS sponsored a Workshop[(1)] on computer performance evaluation. One of the major results of that Workshop was a consensus that there have been two separate approaches to the evaluation of information systems performance -- one which focuses on the computer system domain and the other whose attention is directed at the application system (user) domain. Each has its own goals and measures: the computer system domain measures are based on resource queueing and utilization statistics and the user domain is evaluated through the performance of requested services. Measures such as throughput and response time are common for the latter. The Workshop also concluded that any performance analyses "should recognize both the costs of a computer installation and the needs of users for service".

The complexity of the design problem for modern computer based information systems has increased significantly over its predecessors due to:

a. the servicing of an expanding range of user or uses with corresponding diverse performance goals and resource requirements, and

b. the dynamic and unpredictable behavior of the system as a function of design decisions and load mix.

Thus, it is quite possible to improve the performance of the system with respect to one or more users at the expense of others. Likewise, because system resources are used by different users, improving the performance characteristics of one or more resources for the benefit of specific users may have an overall detrimental effect on performance. The problem presented to the designer is to configure a system which satisfies the user criterion while achieving system resource related performance criteria.

(1) This was one in a series of Workshops sponsored jointly by ACM and NBS to examine the major issues involving computers. Performance evaluation was chosen as the topic of this Workshop because of its significant impact on computer usage. A summary of the conclusions appears in BOE75.

A computer based information processing system can be viewed as a symbiotic relationship between the system's users and its hardware, software and data resources. Ideally, the system will perform "optimally" when it achieves its user oriented objectives within a minimum cost system. However, optimal solutions are seldom achieved when systems are complex, ill defined or constrained for reasons outside the control of the designer, and thus, the designer usually settles for a satisfactorially behaving system. Hopefully, systematic procedures are employed to achieve system configurations which concurrently meet the user objectives while obtaining efficient utilization of its resources. Current evaluative technologies focus on only one criteria in the system design equation, either the user's or the system's resource performance. The ability to simultaneously ascertain the impact of resource performance on user goals or vice versa is not readily achievable through these methodologies. The purpose of this paper is to describe a methodology which establishes a formal liaison between the evaluation of user goals as a function of system behavior and the analysis of resource performance as a function of user activity.

User oriented analyses with objective functions based on response time, throughput, and cost have been (and are continuing to be) reported in the literature. Most frequently, analytic approaches use queueing models as their basis (GAV76, BUZ73, NEI76 are representative of this type of analysis). Due to the necessity to maintain tractable models, many simplifications are required for a model's analytical solution. Simulation models have also been applied to user oriented analysis (CON65, ROE70). Unfortunately, these models yield only average and/or aggregate measures of system response. As a result of these simplifications, the analyses produced by both of the approaches fail in many cases to identify the relationship between users and resources. Therefore, they are suspect when used to predict the impact on system performances of modifying the current environment.

Alternatively, performance analyses can be made from the system's standpoint, treating the user and his goals in the aggregate. The most common approach is a subsystem study, where a particular part of the information system complex is isolated, with the subsystem user(s) represented by a stochastic generator, both analytic and simulative. The most emphasized areas of research has been the I/O subsystem (ABA68, NAH73, SHE76, HEL70) and CPU utilization (KLE72, AGR76, LEW71). The problem with this level of evaluation is that, although providing valuable local intuitive insight, these models rarely relate to the ultimate information system user, and, therefore, do not provide realistic insight into global performance.

Examinations of complete systems have also been made. Exhaustive hardware/software measurements have been analyzed by Gonzales and Cantrell (GON76, CAN68) while simulation models, including an aggregate user component, have been built by Reeves and Pooch, Norland, and Lum (REE75, NOR71, LUM70). Although results of the evaluations include resource utilization statistics and user oriented measures such as response time, there is little attempt in these models to relate particular resource usage to the effect on user goal attainment. (Two exceptions are Lindsay's study of the KRONOS system (LIN76) and Hall's data base investigations (HAL74).) But from practical experience it is evident that there is indeed a relationship between user goals and resource usage. In fact, Buzen (BUZ76) has recently proposed some fundamental laws for computer performance which relate resource activity to global system/user measures such as response time and throughput.

It is assumed that the objective of good system design is to satisfice both performance related criteria. However, in light of the complexity of modern systems, many design decisions tend to be made without proper supportive evidence on performance. The crux of the problem is to establish a causal relationship between user goal attainment and system resource expenditures. The methodology to be discussed has been designed to establish such a liaison and will be shown to allow for the collection of heretofore hard to obtain evaluative information. The methodology measures the impactof individual user classes on internal system performance and identifies system bottlenecks which inhibit the attainment of user goals. This is achieved by maintaining resource utilization statistics on a user class basis. This methodology presents an evaluative framework which is capable of eliminating many of the numerous non-satisfactory designs by directing the designer to the most advantageous ones. This methodology is an iterative one with each iteration involving three separate but integrated stages. Figure 1 illustrates the activities for an iteration. Briefly the responsibilities for each stage shown in this figure are:

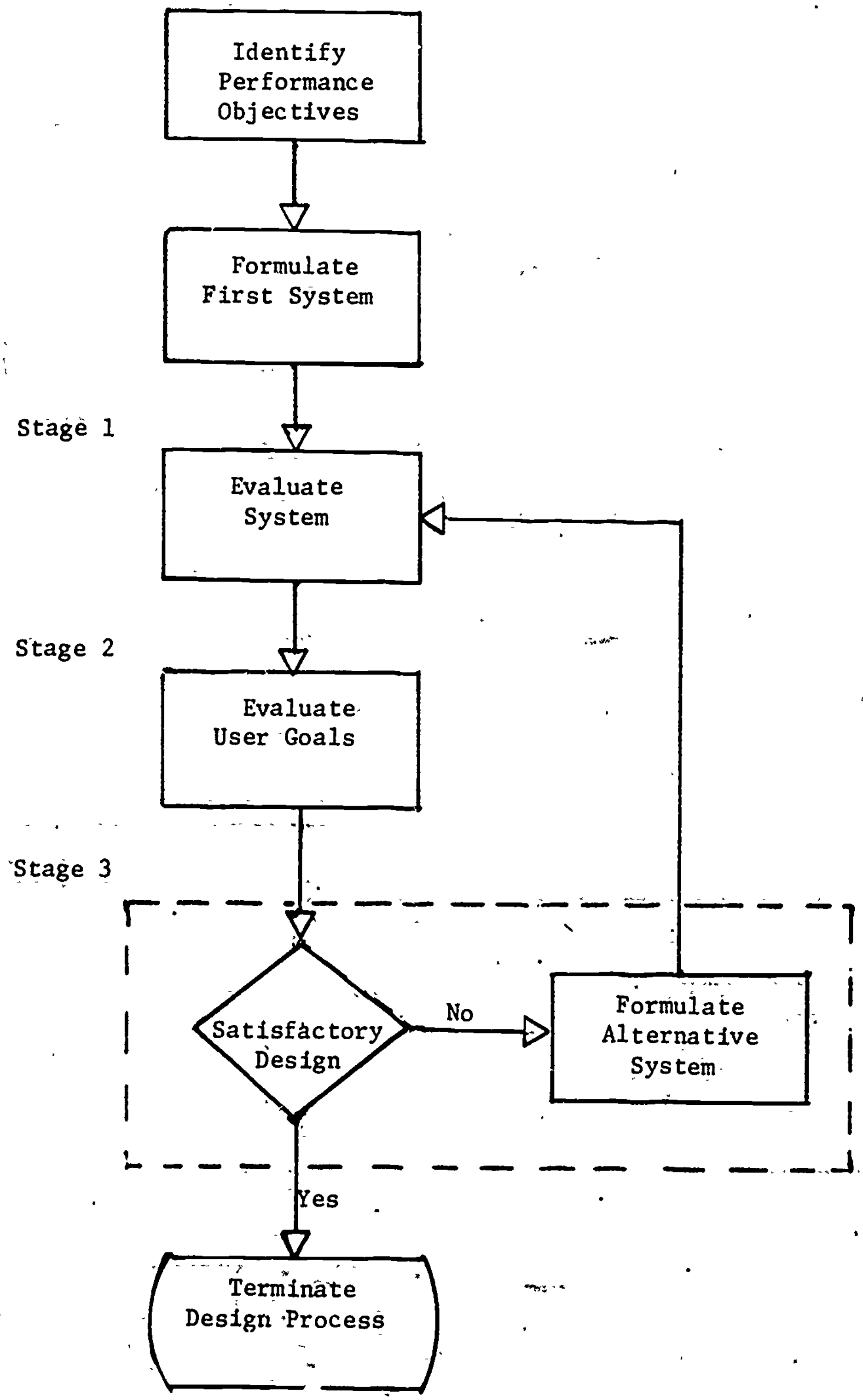


Figure 1. Stages in System Design Process

Stage 1: System Evaluation

This stage is responsible for evaluating the behavior of a specific information system model. It does this by associating the hardware, software and data activities belonging to a specific design with the system's user activites. The outputs of Stage 1 are performance statistics for the resources in the aggregate and for their behavior with respect to identified users (or uses). To perform this function, the IPSS Simulator is employed(1).

Stage 2: User Goal Evaluation

Stage 2 has two purposes, the first is to ascertain whether the user goals are being either over or under achieved. The second purpose is to determine the "best" set of guidelines for altering the current system configuration in order to obtain the user goals with minimum penalty for either under or over achievement. Multiple goal programming is used for this purpose. As will be seen, "best" is a function of the assigned penalty coefficients in the goal programming objective function.

Stage 3: Design Evaluation

Stage 3 has two functions. The first is to ascertain whether or not the current design's performance is satisfactory with respect to both the user criteria and the system criteria. If the design is not satisfactory, then this stage's second goal is to define a new system based upon the current design, prior alterations, and the results of the Stage 1 and Stage 2 analyses. Heuristic procedures are currently employed for Stage 3.

The focus of this paper is on the Stage 2 formulation and its formal liaison to Stage 1. The paper also identifies the unique features of IPSS which permit this multi-stage multi-criteria methodology to be achieved. The paper concludes with a discussion of the use of the Stage 1 and Stage 2 results in the Stage 3 heuristics.

(1) IPSS is a special purpose discrete event simulator whose development was conducted with the support of the National Science Foundation, Grant No. GN-36622.

2. Evaluative Requirements

For the purposes of this methodology, an information system is viewed as including its users and their goals, and the system's services and their subordinate activities. This is illustrated in Figure 2. It is assumed that the system's analyst can identify and classify the system's users according to their service request characteristics and according to the performance constraints imposed upon the system (i.e., goals) when honoring their requests. It is also assumed that the analyst can identify those information system activities which are critical to system performance. Obviously, the complexity of the problem is increased substantially when a system supports diverse users or provides a wide spectrum of services. Whether or not the system is complex or simple, the criteria for identifying system activities should be based upon the sensitivity of the system's performance with regard to changes in their behavior.

Information system services are viewed as being a series of distinct yet interconnected activities which are invoked during the processing of a stream of user requests for the service. Again, Figure 2 illustrates this view of a system. Most likely, system activities are aggregations of one or more traditional computer system functions that perform the following tasks:

1. request (job) scheduling,
2. task management,
3. resource allocation,
4. secondary storage I/O processing, and
5. application processing.

The choice of what constitutes an activity is part of the art of performance evaluation, however, a necessary condition for their selection is that they be measurable and that these measurements distinguish the service rates for separate classes of system services. It is also assumed that the role of performance measurement is to determine the current processing rate for the jth activity with respect to the ith service.

Figure 3 is a schematic of the functional composition of system activities. Each is viewed as an individual queueing system containing

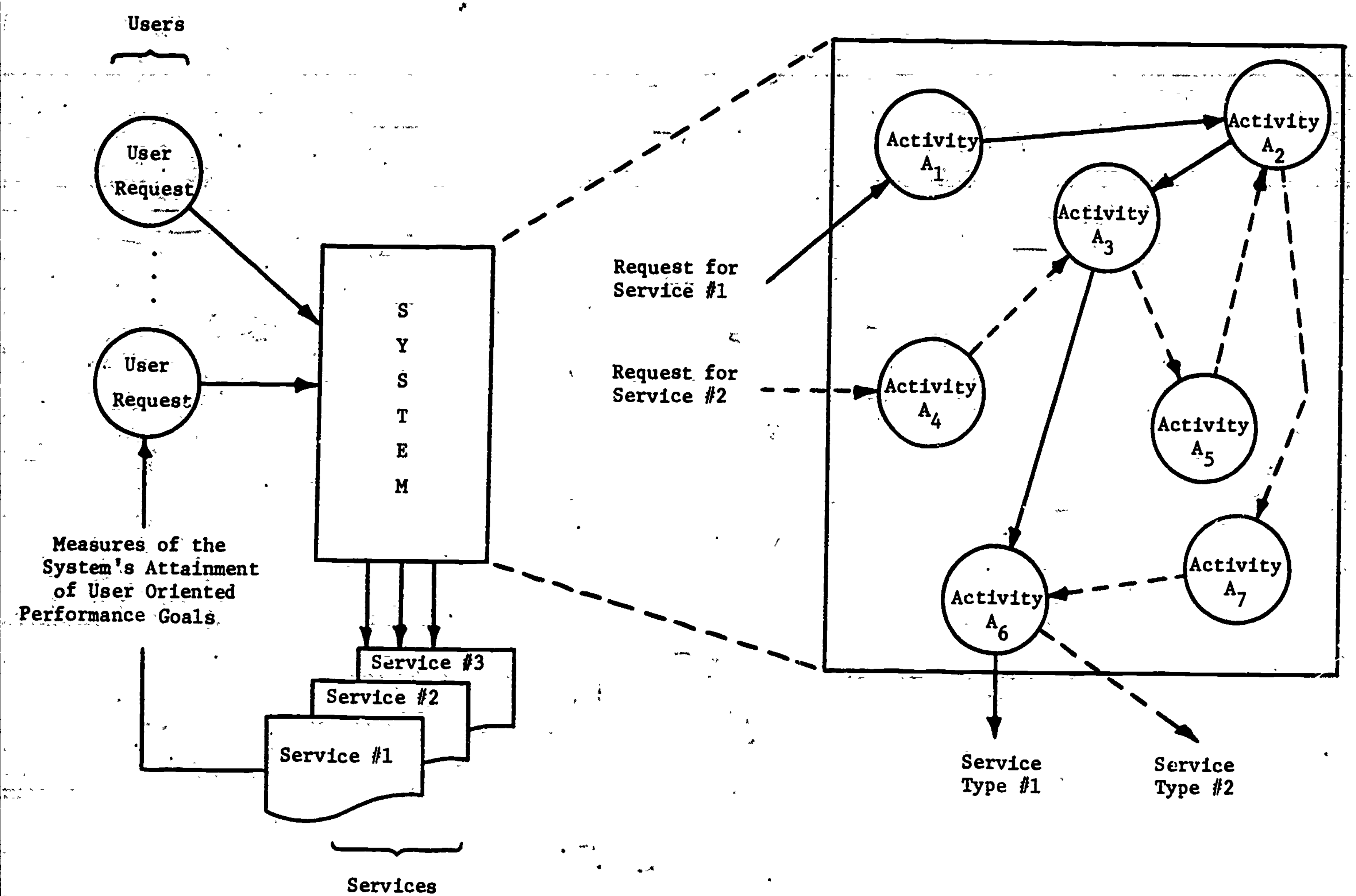


Figure 2. The Methodology's View of an Information System

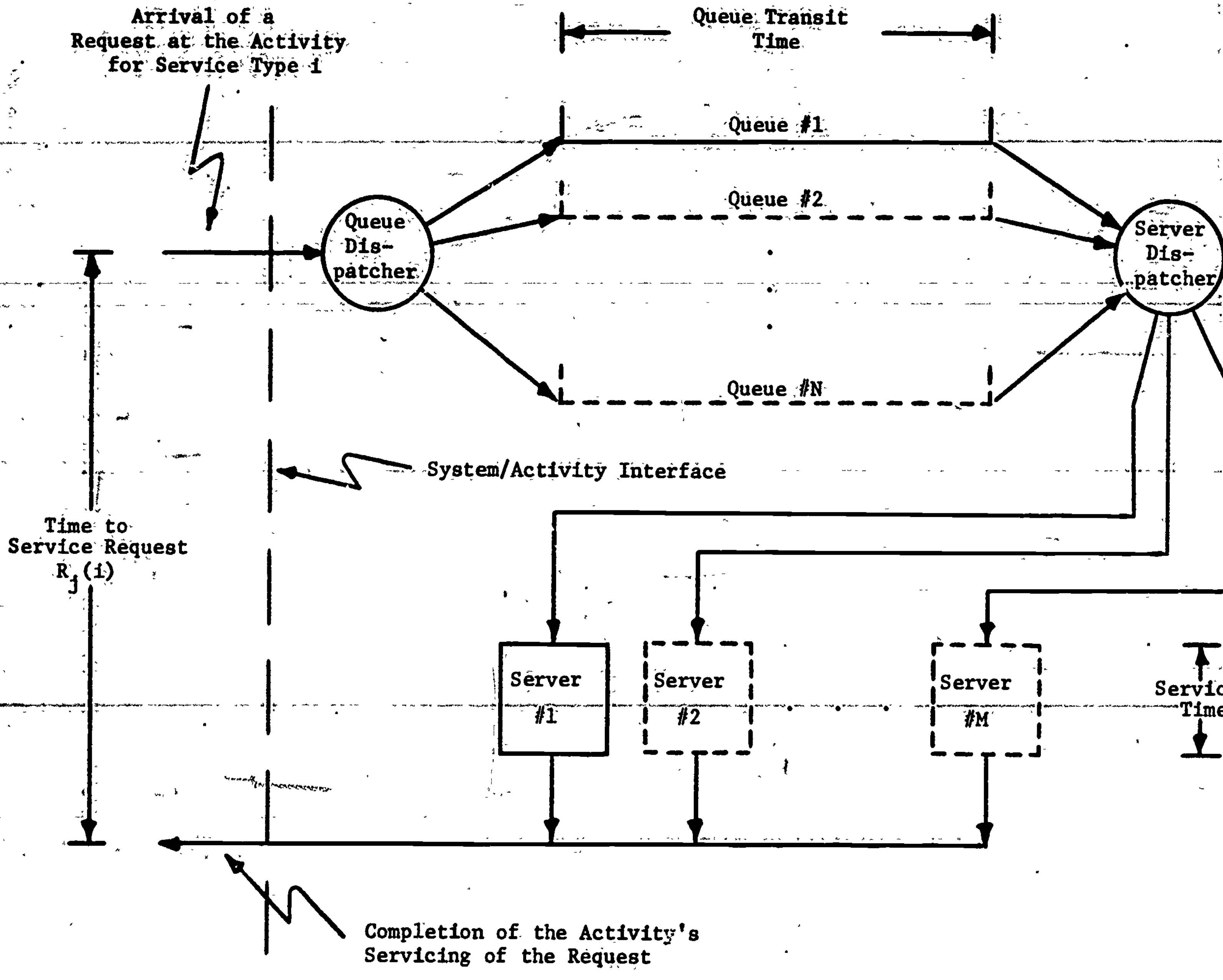


Figure 3. A Conceptual View of an Activity

one or more priority queues and one or more identical servers. Additionally, the performance measure for the activity in processing a request type is the sum of both the queue performance and service functions of the activity. Throughout this paper, the variable $R_j(i)$ is employed to identify this performance of activity A_j, with respect to Service S_i. It is assumed to be the average of performance for all the executions of A_j for S_i. Also associated with each activity A_j is a performance factor β_j which is interpreted as the scaling factor to be applied to the $R_j(i)$ to obtain the level of performance for the jth activity which minimizes the goal programming objective function. It should be noted that the problem of identifying a "good" level of performance for activities, i.e., determining the appropriate values of the $R_j(i)$'s is compounded by the multiple use of the activity by different and possibly conflicting services. Therefore, the modification of an activity's processing rate to achieve one goal may be counter productive to the attainment of another goal. It is to this possibility of multiple conflicting interactions and goals that this methodology is focused.

3. Formulation of the Stage 2 Evaluative Procedure

Stage 2 is based on an evaluative procedure commonly called multiple goal programming (MGP). The procedure was first formulated by Charnes and Cooper (CHA61) in 1961 to solve linear programming problems that had conflicting constraints. Ijiri (IJI65) developed the details of the procedure within the framework of mathematical programming. This technique has been used to solve problems in the areas of strategic management planning such as accounting control (Ijiri), advertising-media planning (Charnes and Cooper (CHA68)), and resource allocation (Lee, (LEE72)). The employment of goal programming in conjunction with information system performance evaluation is a new use of the procedure.

There are three reasons for choosing multiple goal programming for use in this stage of the methodology. First, this approach can evaluate linear and ordinal multiple goal situations, both of which are inherent to information systems evaluation. For example, one user class may pay twice as much for its service, and, therefore, satisfaction of its goals may be worth twice as much as others; a linear relation. On the other hand, certain users, such as a critical patient monitoring application, may have incomparable importance relative to other classes; an ordinal relation. Second, multiple goal programming produces a solution that not only evaluates the total goal situation, but also evaluates each goal, individually. One of the purposes of this methodology is to determine the critical user classes and associated activities. Third, MGP derives the "best" design under the given goal constraints. Other design approaches such as weighting, sequential elimination, and spatial proximity (MAC73), are based on selecting the "best" design from a finite set of alternatives. The purpose of the overall methodology, however, is to design an appropriate system to satisfy the user and resource constraints. The standard formulation of a multiple goal programming problem is:

[A] Minimize $P \cdot D$

Subject to $A \cdot X + D = G$

where

A = a matrix of technological coefficients which can be thought of as the rates at which the ith service uses the jth resource

X = the array of resulting system resource allocation levels

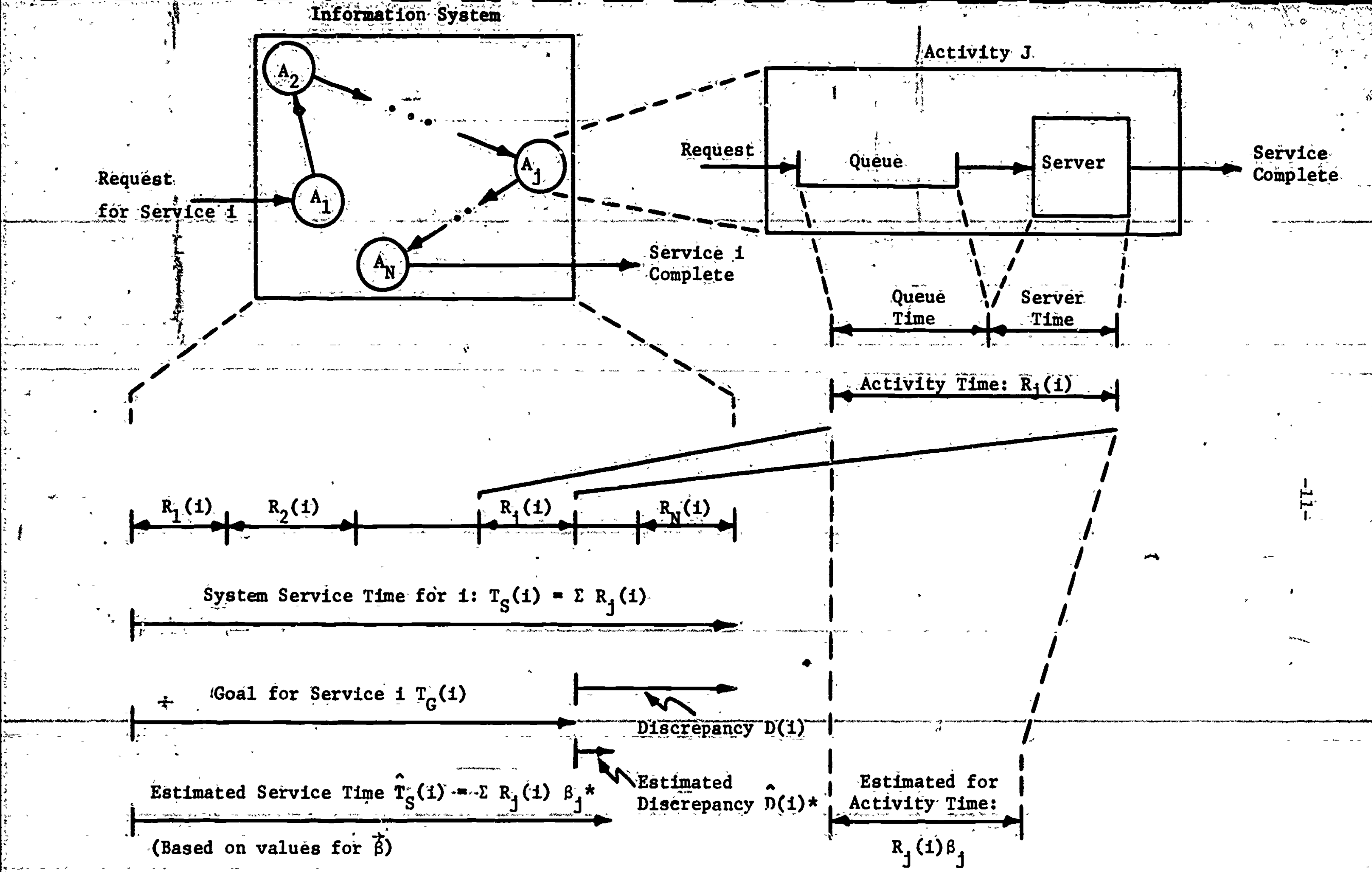
G = the array of service goals

D = the array of discrepancies from these goals

P = the array of penalties associated with the discrepancies in D

and where the objective function is to minimize the product of the discrepancies and their associated penalties. The solution to a multiple goal programming problem represents the best set of levels for the resource allocation vector X such that the objective function is minimized. The remainder of this section discusses the specific formulation for the Stage 2 component of the methodology.

Figure 4 illustrates the relationship between MGP, the information system activities, and its servicing of user requests. The servicing of a request type i is a sequence of activities, A_1, A_2, ..., A_n, each



*Values of β_j's and D(i)'s are the outputs of Goal Program Optimization Problem

Figure 4. Relationship Between Goal Programming and Information System Characterization

assumed to be measurable by $R_j(i)$. In general, the measure can be a function $M(R_j(i))$ of the service time, however, just $R_j(i)$ will be employed in the following discussion.

The performance of the information system for service type i is given by the relation

$$(1) \qquad T_S(i) = \Sigma_{j=1}^{n} (R_j(i))$$

where $T_S(i)$ is the average system response time to service requests of type i. Assuming that the performance goal for the service is $T_G(i)$, then the discrepancy between performance and goal is given by

$$(2) \qquad D(i) = T_G(i) - T_S(i).$$

The objective of MGP is to determine new performance levels for each activity in such a manner that the weighted discrepancy, $P \cdot D$, is minimized (hopefully to zero). Letting β_j be a scaling factor to be applied to the jth activity, then the new performance level for the activity is $R_j(i)\ \beta_j$. Incorporating the β_j's into equation (1) results in the following expression for the discrepancies:

$$(3) \qquad D(i) = T_G(i) - \Sigma_{j=1}^{n} (R_j(i) \cdot \beta_j).$$

Observe that both positive and negative discrepancies are possible, and, therefore, the formulation of the user goal evaluation as a MGP problem becomes [(1)]:

[B] Minimize $P_D^+ \cdot D^+ + P_D^- \cdot D^-$

s.t. $R \cdot \beta + D^- - D^+ = G$

where

- R is a matrix of service rates
- β is the array of scaling factors
- G is the array of user goals
- D^+ and D^- are the arrays of, respectively, the positive and negative discrepancies from the user goals
- P^+ and P^- are the arrays of penalties associated with the corresponding positive and negative discrepancies.

(1) The complete derivation appears in a previous paper by the authors presented at the Annual Conference of the Computer Measurement Group, November, 1976 (CHN76).

The formulation serves two purposes: it evaluates goal achivement and produces the β_j's. By setting the values of the β's to reflect only the current configuration (i.e., $\beta_j = 1$), the evaluation of the system's attainment of the user goals is accomplished.

Experiments with formulation [B] produced valid, but impractical sets of β's. The MGP problem as stated allowed for the possibility of solutions where a β_j could equal 0, clearly an unacceptable situation. In order to inhibit this type of solution, limits were placed on the range of possible β_j values. This was accomplished with the following set of additional constraints:

(4) $$\beta_j + \mu_j^- - \mu_j^+ = L_j$$

(5) $$\beta_j + \nu_j^- - \nu_j^+ = H_j$$

where

$$0 < L_j \leq 1$$

$$1 \leq H_j$$

L_j is used to restrict the alternative possibilities for the case that $\beta_j < 1$ while H_j is used for those cases that $\beta_j > 1$. In general, the set of all positive discrepancies for L_j's, $(\mu_1^+, \mu_2^+, \ldots, \mu_j^+) = M^+$ and likewise for H_j $(\nu_1^+, \nu_2^+, \ldots, \nu_j^+) = N^+$. ($M^-$ and N^- have similar definitions).

These constraints are reflected in the objective function in a manner different than previous constraints. Instead of minimizing both discrepancies, only one is minimized. In the case of L_j only μ_j^- is included, since, if μ_j^- is driven to zero, then $\beta_j - \mu_j^+ = L_j$, implying that $\beta_j > L_j$, the desired condition. Similarily, for H_j only υ_j^+ is in the objective function because minimizing $_j^+$ results in $\beta_j < H_j$.

These added constraints also have a physical interpretation relative to the evaluation of the system. No activity can be eliminated from a system (i.e., $\beta_j = 0$) since L_j must be greater than 0. In general, however, L_j represents the lower bound on the degree of reduction feasible for the current level of usage for an activity. For example, $L_j = .25$ implies that the usage time for activity j can be made, at most, four times shorter, being reduced to 25% of its current level. Similarily, H_j represents the upper bound on the degree to which an activity's level can be increased (made longer). It must be emphasized that these limits are only rough estimates, not exact values.

In order to reduce the number of alternatives one should minimize the number of modifications indicated per evaluation iteration. Since modifications are characterized by the production of β_j's not equal to 1, a secondary objective of Stage 2 is to produce as few $\beta_j \neq 1$ solutions as possible. This is accomplished by including the constraint equation

(6) $$\beta_j + \varepsilon_j^- - \varepsilon_j^+ = 1$$

while minimizing both ε_j^+ and ε_j^-

Constraints of this type provide a default value of 1 for the multiple goal programming procedures in the case where an activity is neither critically inefficient or excessive. (Note: $(\varepsilon_1^+, \varepsilon_2^+, \ldots, \varepsilon_j^+) = E^+$.)

As a result of these added constraints, the actual formulation of the MGP problem used in Stage 2 is given in formulation [C] below:

[C] Minimize $$P_D^+ \cdot D^+ + P_D^- \cdot D^- + P_M^- \cdot M^- + P_N^+ \cdot N^+ + P_E^+ \cdot E^+ + P_E^- \cdot E^-$$

s.t. $$R \cdot \beta + D^- - D^+ = G$$
$$\beta + M^- - M^+ = L$$
$$\beta + N^- - N^+ = H$$
$$\beta + E^- - E^+ = \bar{1}$$

where

R is the matrix of service rates

β is the array of scaling factors

G, L, H, and $\bar{1}$ are the arrays of goals for the user criteria and the respective β constraints

$D^{\pm}$, $M^{\pm}$, $N^{\pm}$ and $E^{\pm}$ are the arrays of positive and negative discrepancies from the respective goals

$P_{\bar{D}}^+$, $P_{\bar{M}}^+$, $P_{\bar{N}}^+$, and $P_{\bar{E}}^+$ are the arrays of penalties for the associated discrepancies.

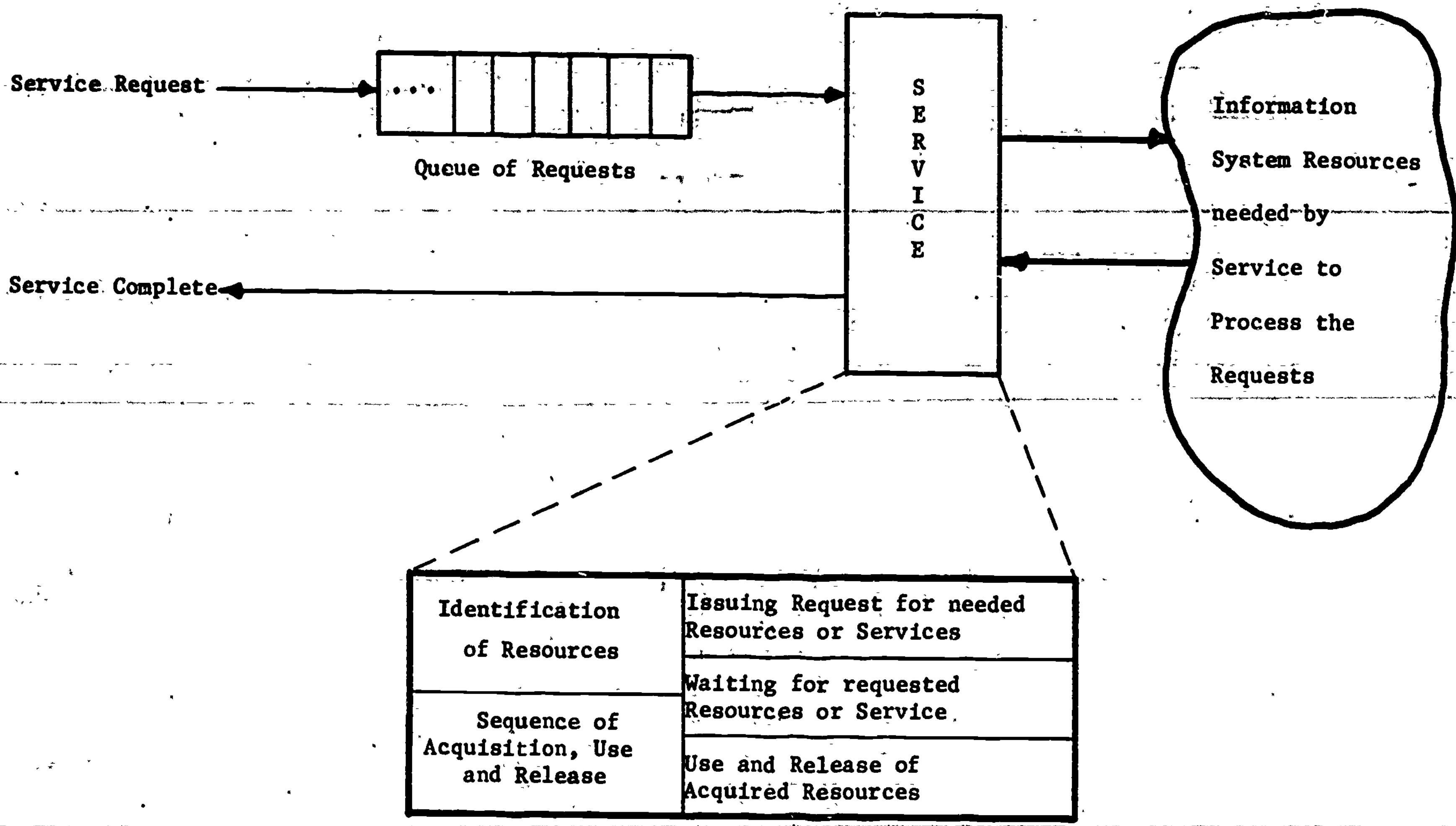
The solution variables for the MGP problem are the β's. They identify those activities that must be altered in order to improve user based or system based performance. If the value for a $\beta_j = 1$ then the service characteristics of activity j were adequate to satisfy all the user's criteria. If a $\beta_j < 1$, this implies that the current service rate of activity j is insufficient to meet the system's needs. The new service rate for the activity should be $R_j(\cdot) = (\beta_j) * (R_j(\cdot))$. If a $\beta_j > 1$ then the current service rate of activity j is faster than necessary and there exists the possibility of excess capacity. The new unit rate should be $R_j(\cdot) = (\beta_j) * (R_j(\cdot))$.

Assuming that an activity follows the characterization in Figure 3, then the analyst has three avenues of action when a $\beta_j \neq 1$. First, he can analyze the queue dispatching discipline in order to increase queue throughput (or possibly replace it with a simpler one if $\beta_j > 1$). Second, he can alter the service rate characteristics of the servers, e.g., slower hardware devices. And third, he can increase (decrease) the degree of parallelism among servers, for example, by adding (removing) a second channel, controller, etc.

4. Liaison With Stage 1

The critical factors in the Stage 2 evaluation are the values for the $R_j(i)$'s needed by the MGP formulation. These values are calculated in Stage 1 and are the statistical measure produced vis-a-vis the simulation. The specific model to be evaluated is the result of the heuristic procedures constituting Stage 3. The liaison is based upon the assumption that an information system can be viewed as a collection of resource allocation and task management activities and user oriented services. This view is supported by the literature, e.g., Madnick (MAD74) and Zurcher and Randall (ZUR69). IPSS also views the modeling of an information system in a similar manner, and thus, facilitates the development of the formal liaison with the MGP user evaluation.

The view of the system taken in Stage 2 (as illustrated in Figure 2) has an analog in IPSS. Its basic modeling concept is that of a service as shown in Figure 5. The service is classified in IPSS as a procedural



-16-

Figure 5. Functions of IPSS Service Entities

facility and is capable of representing any information system activity including request (job) scheduling, task management, resource allocation, secondary storage I/O processing and application software. Since services are allocatable facilities in IPSS they have associated with them both queueing and utilization statistics. Furthermore, service behavior can be predicated on the quantity and characteristics of other IPSS hardware and software facilities. Therefore, the statistics associated with service facilities have the appropriate structure to service as the R_j's needed in Stage 2.

To complete the formal liaison between the two stages a second feature is employed. This is the Task facility. Through its use, the service facility statistics can be automatically segregated into service statistics by user. In this manner, the statistic $R_j(i)$, required by Stage 2, is collected. Thus, the Stage 2 users (indexed by i) and the activities (indexed by j) are, respectively, an IPSS model's Task and Service facilities. An $R_j(i)$ is the sum of queueing and utilization statistics gathered for Service i when executing Task j.

Stage 1 must also be adaptive to model changes dictated via Stage 3. Again, the IPSS model synthesis philosophy and language constructs permit the desired adaptiveness. This is possible, for reasons too detailed to discuss in this paper. A complete description of IPSS is available in the document titled "The Information Processing System Simulator (IPSS): Language Syntax and Semantics" (DEL76). Briefly, however, possible modifications to an existing and executing model without requiring complete reformulation include changes to:

1. timing and space characteristics associated with secondary storage hardware and storage media,
2. the secondary storage I/O configuration,
3. user usage patterns and service requirements,
4. file organization methods and space management policies,
5. the queueing disciplines associated with job scheduling, resource allocation and task management, and
6. memory management policies.

IPSS supplies the Stage 1 processing with a capability of being self-adaptive with respect to Stage 3 outputs. Currently, the methodology employs modeler assistance in Stage 3. Future research will be directed at providing more sophisticated heuristics for Stage 3 in order to provide a truly self-contained iterative methodology for the multiple criteria evaluation of information systems.

5. Stage 3 Analysis

The functions of Stage 3 of the methodology are to determine whether the current configuration is satisfactory and to formulate a new model in light of the data provided from Stage 1 and Stage 2. Figure 6 shows the information flow to Stage 3. New models reflect the performance goals of both the user and the system. Heuristics using Sutherland's definition of a heuristic[1] are employed in Stage 3.

The problems encountered are complex and unstructured. Determining if the current design is acceptable requires a mixture of objective and subjective reasoning. It would be a rare situation if all the user goals and system constraints were satisfied simultaneously. Generally, an extremely wide spectrum of acceptable performance levels and alternative designs are possible, at each iteration, to satisfy both user and system criteria. Trade-offs will dominate the decision processes. Many factors affecting suitable designs may not be included in the formulations and procedures of the first two stages. For example, there may be external political, organizational or economic considerations that are not directly related to the performance of the system, but may be a major factor in the final decision. The methodology does assume, however, that the heuristic procedures have access to this external criterion.

When it has been determined that another iteration is desirable, it is assumed that the heuristics will examine current and past designs. Whatever the heuristic employed, ideally its objective is to produce a sequence of models whose β_j characteristics (for all β's) behave as follows:

(1) A heuristic is "a disciplined trial-and-error process,..., an exercise in successive improvement, where we may learn from both success and failure and where the criteria for success and failure may vary with what we have previously learned" (p. 183, (SUT75)).

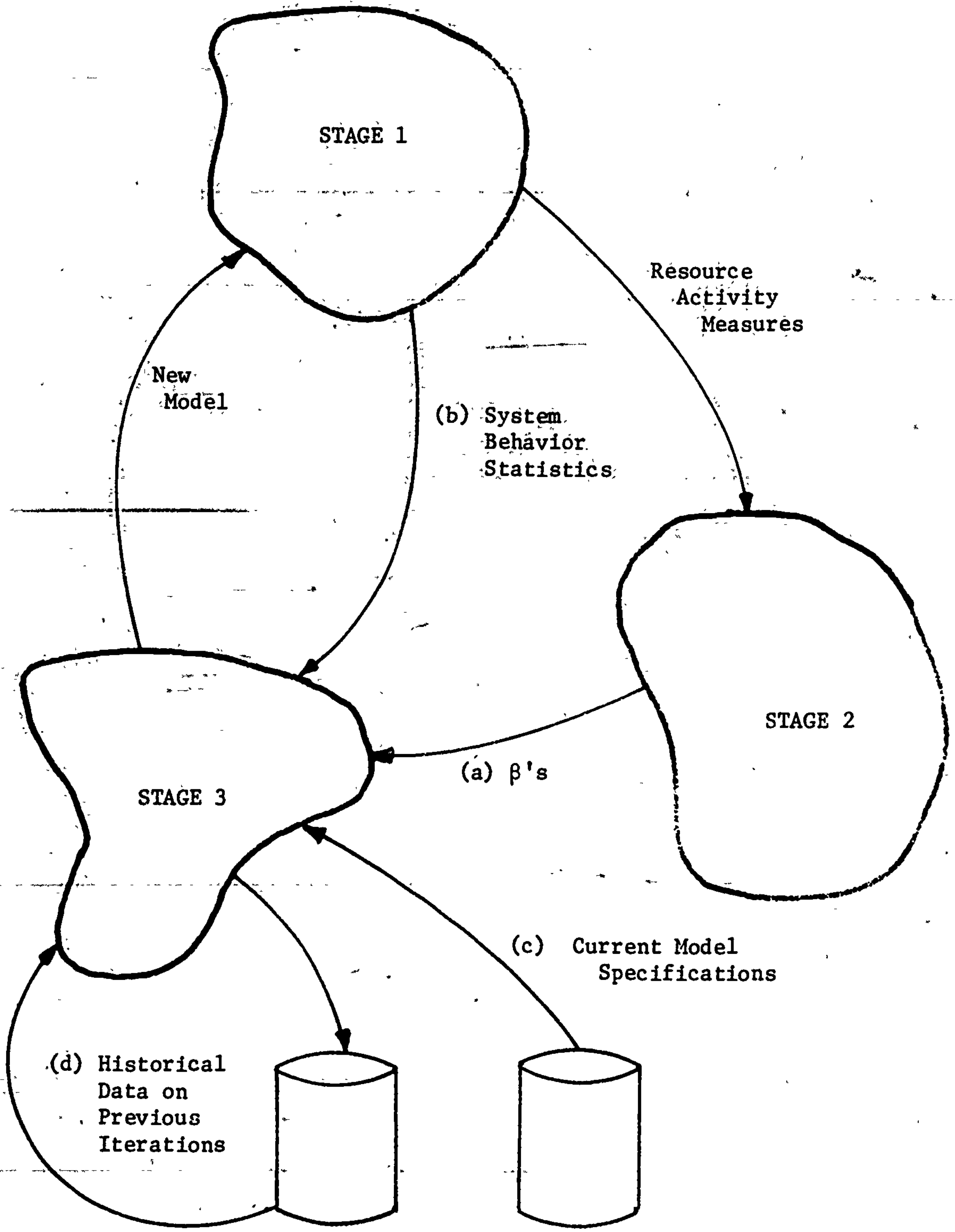
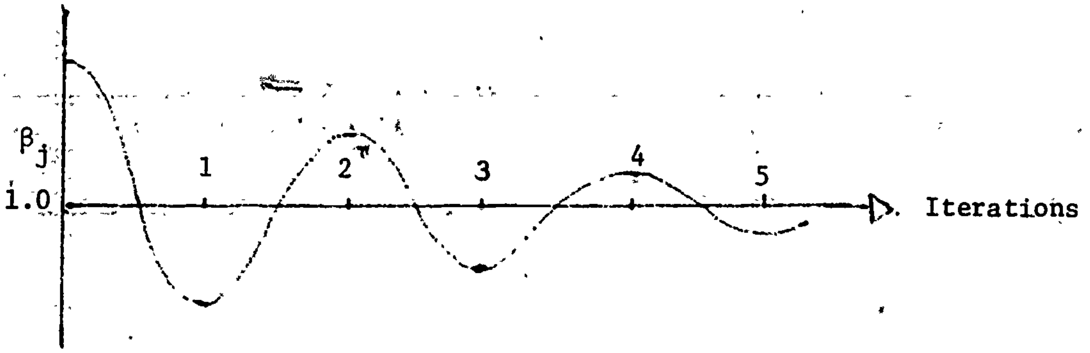


Figure 6. Information Flow to Stage 3



The emphasis of the current research is to provide insight into the decision process for improving the performance of information systems. Stage 3 is this decision process. It is aided by input from four sources within the methodology. These sources are: (a) the Stage 2 outputs, specifically the β_j's identified to improve user and system goal performance, (b) system behavior statistics from Stage 1, (c) the current model, and (d) historical data from prior iterations. It should be emphasized that at this juncture in the development of the methodology no formal heuristic procedures have been implemented. It is one of the purposes of this research, however, to investigate the appropriateness and success of various heuristic decision rules. Rules of thumb such as those proposed by Buzen (BUZ76) are possible avenues to be investigated.

6. An Example

In order to validate the procedures developed in this methodology and the liaison between Stage 1 and Stage 2, a test case was developed. This example was modeled and executed in IPSS to satisfy the Stage 1 requirements. Several iterations were applied, demonstrating the evaluative capabilities of the methodology. The following is a description of the problem, the corresponding model and the results of the first two iterations.

The example is a model of an on-line document retrieval system. There are three files associated with the system; an author/title index (A/T), a system document identification file (ID), and the document file itself (DOC). They are structurally related such that an entry in the A/T file points to one or more entries in the ID file and each ID file entry in turn is associated with only one DOC entry.

The model is designed so that a unique activity is associated with the accessing of each file; Activity 1 with the A/T file, Activity 2 with the ID file and Activity 3 with the DOC file. Each activity performs similar functions: obtaining and releasing devices, reading records, performing I/O techniques, but to different files.

It is assumed that the system supports three user classes, each with a different demand on the retrieval system and each characterized by a different combination of activities. The purpose of the first user class, $USER_1$, is to retrieve a document for a particular author, thus utilizing all three activities. Those in the second user class, $USER_2$, want to determine the existence/non-existence of an entry in the system for a given author, and therefore, need to use only Activity 1. The final user class, $USER_3$, already has the address of the ID entry and wants to retrieve the associated DOC entry requiring only Activity 2 and Activity 3.

In order to complete the formulation of the performance evaluation problem for this methodology, assumptions concerning the performance of the system were made. A summary of these assumptions for the user goals and β constraints and the penalties associated with the corresponding discrepancies in accordance to the requirements of formulation [C] is shown below.

$$G = (150.0,\ 100.0,\ 400.0)$$
$$L = (.2,\ .2,\ .2)$$
$$H = (5.0,\ 5.0,\ 5.0)$$
$$\overline{I} = (1.0,\ 1.0,\ 1.0)$$
$$P_D^+ = (1000.0,\ 10.0,\ 1000.0)$$
$$P_D^- = (1.0,\ 0.0,\ 0.0)$$
$$P_M^+ = P_N^- = P_E^+ = P_E^- = (1.0,\ 1.0,\ 1.0)$$

The model constructed in IPSS assumed a simple configuration of one processor and one bank of IBM 2314 type direct access devices. Under a given loading (which is not a controllable variable in this methodology) the resulting performance statistics, the values of matrix R, are shown below.

$$\begin{array}{lll} Q_1(1) = 0.0 & Q_2(1) = 0.0 & Q_3(1) = 11.4 \\ S_1(1) = 37.2 & S_2(1) = 37.1 & S_3(1) = 90.9 \\ R_1(1) = \overline{37.2} & R_2(1) = \overline{37.1} & R_3(1) = \overline{102.3} \end{array}$$

$$\begin{array}{lll} Q_1(2) = 12.2 & Q_2(3) = 0.0 & Q_3(3) = 8.9 \\ S_1(2) = 36.3 & S_2(3) = 39.1 & S_3(3) = 93.5 \\ R_1(2) = \overline{48.5} & R_2(3) = \overline{39.1} & R_3(3) = \overline{102.4} \end{array}$$

This performance information, coupled with the goal assumptions, was input to Stage 2. The evaluation of the current configuration's performance with respect to the set of user's goals is shown in Table 1. It indicates that the goals of user classes 2 and 3 were satisfied with a good margin of slack; (which is not penalized in this example) while the goal of the first user class was not satisfied (under-achievement implying non-satisfaction).

Table 1. User Goal Evaluation

Goal	Under-Achievement	Over-Achievement
$USER_1$	26.6	0.0
$USER_2$	0.0	51.5
$USER_3$	0.0	258.5

In the second phase of Stage 2, the β's are allowed to be manipulated until they satisfy the user goal constraints and best satisfice the system guideline constraints. The result is the identification of those activities whose performance can be, and needs to be, improved with respect to one or both of the criteria. The values for the β's as calculated were

$$\beta_1 = 1.0 \qquad \beta_2 = 1.0 \qquad \beta_3 = .74$$

These are interpreted as indicating that both Activity 1 and Activity 2 were adequate to meet the user demands put to them. Activity 3, however, was found to be insufficient to satisfy the requirements of user classes 1 and 3. The modification indicated is to reduce the present level of usage for Activity 3 by at least 1/4 in order to satisfy the user goals, in particular, the first user class goal.

The determination of whether to cease the design loop by accepting this performance or to continue by modifying the existing model is made in Stage 3. Given the stated goal/penalty structure, it was assumed that the under-achievement of $USER_1$ goal was at an unacceptable level and the design process must continue if possible. By examining the queueing and service time statistics for the first iteration, one can eliminate some of the modification possibilities. The result of Stage 3 analysis was a decision to replace the IBM 2314 type device with a faster one, i.e., an IBM 3330 type device.

The original model of this example system was dynamically altered to reflect this modification. Under the same loading as before, the following performance statistics were accumulated.

$Q_1(1) = 0.0$	$Q_2(1) = 0.4$	$Q_3(1) = 3.8$
$S_1(1) = 11.9$	$S_2(1) = 11.3$	$S_3(1) = 27.6$
$R_1(1) = \overline{11.9}$	$R_2(1) = \overline{11.7}$	$R_3(1) = \overline{31.4}$

$Q_1(2) = 3.8$	$Q_2(3) = 0.0$	$Q_3(3) = 0.0$
$S_1(2) = 11.1$	$S_2(3) = 12.0$	$S_3(3) = 29.1$
$R_1(2) \quad \overline{14.9}$	$R_2(3) = \overline{12.0}$	$R_3(3) = \overline{29.1}$

Stage 2 analysis showed that now all three user class goals were satisfied (i.e., over-achieved). The calculation of the β's, however, indicated that while Activities 1 and 2 were still adequate ($\beta_1 = \beta_2 = 1.0$), Activity 3 now had the possibility of excess capacity ($\beta_3 = 4.0$). Although a slower and probably less expensive device for

Activity 3 would be more appropriate, we had found in the first iteration that such a device was not able to satisfy all the user goals. Therefore, in future iterations, Stage 3 procedures had to examine more subtle methods of improving performance.

7. Conclusion

Modern information systems do not exist as entities unto themselves, but must interact with their environment, i.e., their users. The loading and mix of the users affect the performance of the system resources and likewise, the service characteristics of the system resources affect the satisfaction of user goals. In order to design such systems, one must satisfy a large set of users demanding a conflicting set of performance goals while operating within efficiency and minimum cost constraints. Thus, performance evaluation of information systems is a multiple criteria problem. Concurrently satisfying both of these sets of criteria is the goal of this methodology. Current available techniques, however, only address one side of the problem, either the user or the system. The methodology described in this paper establishes a formal liaison between the evaluation of user goals as a function of system behavior and the analysis of system resource performance as a function, of user demand, thereby, facilitating multi-criteria evaluation.

The methodology is iterative, comprising three separate but integrated stages. The first stage models and evaluates system behavior. The particular technique employed in the first stage is IPSS and it is able to collect the necessary statistic, $R_j(i)$. The second stage evaluates the user based criteria and provides evaluative insight into performance improvement. Solution of the MGP formulation in [C] produces a set of β's, the variables of Stage 2 which indicate inefficiencies and/or excesses in the current model. And finally, the third stage heuristically determines the current model's acceptability and need for modifications.

The evaluative procedures developed for this methodology have been shown to be valid in practice. Furthermore, this methodology provides an excellent basis for continued research into areas such as:

a. investigation into the causal relationships between user demand and system activity,
b. sensitivity analysis of these relationships,
c. investigation into suitable heuristics for Stage 3, either testing existing heuristics or development of new ones, and
d. development of heuristic/modification rules to close the design loop into an automatic self-modifying process.

REFERENCES

ABA68 Abate, J., Dubner, H., and Weinberg, S. B., "Queueing Analysis of the IBM 2314 Disk Storage Facility", JACM, vol 15, no. 4, (1968), pp. 577-589.

AGR76 Agrawala, A. K., and Larsen, R. L., "Experience with the Central Server Model on a Lightly Loaded System", Symposium on the Simulation of Computer Systems IV, (1976), pp. 102-109.

BOE75 Boehm, B., and Bell, T. E., "Issues in Computer Performance Evaluation: Some Consensus, Some Divergence", PER, vol. 4, no. 3, (1975), pp. 4-39.

BUZ73 Buzen, J. P., "Computer Algorithms for Closed Queueing Networks with Exponential Servers", CACM, vol. 16, no. 9, (1973), pp. 527-531.

BUZ76 Buzen, J. P., "Fundamental Laws of Computer Performance", in Proc. of Int'l. Symp. on Computer Performance Modeling, Measurement and Evaluation (CPMME), (1976), pp. 200-210.

CAN68 Cantrell, H. N., and Ellison, A. L., "Multiprogramming System Performance Measurement and Analysis", AFIPS Conf. Proc., vol. 22, (1968), pp. 213-221.

CHA61 Charnes, A., and Cooper, W. W., Management Models and Industrial Applications of Linear Programming, New York: John Wiley & Sons, Inc., 1961.

CHA68 Charnes, A., et al., "A Goal Programming Model for Media Planning", Management Science, vol. 14, no. 8, (April 1968), pp. 423-430.

CHN76 Chandler, J. S., and DeLutis, T. G., "A Methodology for the Performance Evaluation of Information Systems Under Multiple Criteria", Proc. of Computer Measurement Group, (1976), pp. 221-230.

CON65 Conger, C. R., "The Simulation and Evaluation of Information Retrieval Systems", Report 352-R-17, (April 1965).

DEL76 DeLutis, T. G., "The Information Processing System Simulator (IPSS): Language Syntax and Semantics, unpublished research report (Grant No. GN-36622).

GAV76 Gaver, D. P., and Hunfeld, G., "Multitype Multiprogramming: Probability Models and Numerical Procedures", Proc. of CPMME, (1976), pp. 38-43.

GON76 Gonzalez, G., "Using Covariance Analysis as an Aid to Interpret the Results of a Performance Measurement", Proc. of CPMME, (1976), pp. 179-186.

HAL74 Hall, W. A., "A Simulation Model to Aid in the Design and Tuning of Hierarchical Databases", Winter Simulation Conference, (1974), pp. 277-284.

HEL70 Hellerman, H. R., and Smith, H. J., "Throughput Analysis of Some Idealized Input, Output and Computer Overlap Configurations", Computing Surveys, vol. 2, no. 2, (1970), pp. 111-118.

IJI65 Ijiri, Y., Management Goals and Accounting for Control, Chicago: Rand McNally, 1965.

KLE72 Kleinrock, L., and Muntz, R. R., "Processor-Sharing Queueing Models of Mixed Scheduling Disciplines for Time-Shared Systems", JACM, vol. 19, no. 3, (1972), pp. 464-482.

LEE72 Lee, S. H., Goal Programming for Decision Analysis, Philadelphia: Auerbach, 1972.

LEW71 Lewis, P. A. W., and Shedler, G. C., "A Cyclic-Queue Model of System Overhead in Multiprogrammed Computer Systems", JACM, vol. 18, no. 2, (1971), pp. 199-220.

LIN76 Lindsay, D. S., "A Hardware Monitor Study of a CDC KRONOS System", Proc. of CPMME, (1976), pp. 179-186.

LUM70 Lum, V. Y., Ling, H., and Senko, M. E., "Analysis of a Complex Data Management Access Method by Simulation Modeling", FJCC, (1970), pp. 211-222.

MAC73 MacCrimmon, K. R., "An Overview of Multiple Objective Decision Making", Multiple Criteria Decision Making, eds., J. L. Cochrane and M. Zeleny, Columbia, S. C., (1973), pp. 18-44.

MAD74 Madnick, S., and Donovan, J., Operating Systems, N. Y., McGraw-Hill, 1974.

NAH73 Nahourii, E., "Direct Access Device Simulation", IBM Systems Journal, vol. 13, no. 1, (1973), pp. 19-31.

NEI76 Neilson, J. E., "An Analytic Performance Model of a Multiprogrammed Batch Time-Shared Computer", Proc. of CPMME, (1976), pp. 59-70.

NOR71 Norland, K. E., and Bulgren, W. C., "A Simulation Model of GECOS III", Proc. of ACM, (1971), pp. 596-612.

REE75 Reeves, T. E., and Pooch, U. W., "A Multiple Subsystem Simulation of Processor Scheduling", Symposium on the Simulation of Computer Systems III, (1975), pp. 129-135.

ROE70 Roehrkasse, R. C., and Smith, D., "Simulation of Operating Systems", Tech. Report GITIS-70-11, (1970), Georgia Inst. of Tech.

SHE76 Sherman, S. W., and Bric, R. C., "I/O Buffer Performance in a Virtual Memory System", Symposium on the Simulation of Computer Systems, (1976), pp. 24-35.

SUT75 Sutherland, J. W., Systems: Analysis, Administration, and Architecture, Van Nortrand Reinhold, New York, (1975).

ZUR69 Zurcher, F. W., and Randall, B., "Iterative Multi-Level Modelling: A Methodology for Computer System Design", IFIP 68, pp. 867-871.