A Multiple Criteria Approach for Evaluating Information Systems*

By: John S. Chandler

Abstract

An information system can be viewed as a symbiotic relationship between the users of the system and the system itself. Ideally, an information system should be evaluated with equal consideration given to both user constraints and to system constraints. The approach described in this article provides the analyst with a framework for gaining insight into information system performance from both user and system viewpoints by establishing a causal relationship between user goal attainment and system activity. This approach produces not only measures of current performance, but also predictive measures of future performance.

The approach is based on a multiple goal programming formulation of the information system design evaluation problem. This article presents an overview of the formulation and its interpretation. The focus is on the analysis of an example system facilitated by this approach. A discussion of the applicability of the approach concludes this article.

Keywords: MIS evaluation, goal programming, multiple criteria evaluation, simulation

ACM Categories: 2.44, 3.50, 3.72, 8.2

Introduction

Information systems can be evaluated from two different perspectives. One focuses on the computer system domain and the other on user domain. Each has its own goals and measures. In the computer system domain, performance is measured in terms of resource utilization, cost, and efficiency while in the user domain, throughput, reliability, and response time are common measures. A workshop sponsored by ACM and NBS [1] 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 evaluation problem for modern computer based information systems has increased significantly over its predecessors. This is due to many factors such as (a) an expanding range of users and applications with a corresponding expanding set of diverse performance goals and resource requirements, (b) an increased dependence of system behavior on subtle design decisions and changes in user load mix, and (c) a growing demand to achieve conflicting performance objectives, e.g., time vs. cost vs. effectiveness.

In this operating environment, it is quite possible to design a system such that good performance for one or more users is gained at the expense of others. Furthermore, because system resources may be used by different users, improving the performance characteristics of one or more resources for the benefit of specific users may have a detrimental effect on overall performance. The problem presented to the analyst is to configure a system which satisfies the users' effectiveness criteria while simultaneously achieving system performance criteria.

The approach presented here provides a mechanism for the analyst to gain insight into information system performance with respect to both user and system criteria. The causal liaison established between user goal attainment and system activity produces measures of current performance and predictive measures of future performance. The approach is based on three stages: System Evaluation, User Goal Evaluation, and Design Evaluation. Total system evaluation is viewed as being iterative, with each iteration...
Evaluating Information Systems

involving the invocation of these three stages to improve system performance.

The System Evaluation Stage evaluates the behavior of a specific information system with respect to system measures. It produces performance statistics for the resources in the aggregate and for their behavior with respect to identified users. Stage Two, User Goal Evaluation, first ascertains the degree of user goal achievement and then determines guidelines for altering the current system configuration for performance improvement. Multiple Goal Programming (MGP) [5] is used as a base for this stage. The Design Evaluation Stage ascertains the satisfaction of the current design with respect to both user and system criteria. If the design is not satisfactory, then a new system is defined based upon the current design, prior alterations, and the results of Stage One and Stage Two analyses.

The focus of this article is on the application of this approach to information system evaluation. The approach is briefly described and then applied to a specific example, stage by stage. A discussion of the benefits of this approach concludes this article.

Overview of the Approach

The approach takes the following view of information system activity. When a user request enters the information system, it invokes a sequence of system services. The specific services in this sequence perform the information system processes necessary to satisfy the user request, such as compilation, record retrieval, sorting, and report writing. The current implementation of the approach characterizes the system's response linearly. Since most computer based systems have concurrent operations, the sequence of system services, which may have such concurrent services, must be linearized. This is accomplished by activities, conceptual constructs corresponding to a service or set of services. For a given request, appropriate definitions of activities can partition the sequence of services into a non-overlapping linear sequence of activities.

Assume that a given user request \( i \) is characterized by a sequence of activities, \( A_1, A_2, \ldots, A_N \). Let the average performance of one of those activities, \( A_j \), with respect to the associated invoking user request be denoted as \( R_j(i) \). For example, if the user evaluates performance in terms of total system response time, then \( R_j(i) \) is the amount of time spent in activity \( A_j \) while responding to user request \( i \). If the user evaluates performance in terms of cost per run, the \( R_j(i) \) is the contribution to the total cost of the run attributable to activity \( A_j \). The approach allows for a given user request to be measured (evaluated) by several measures concurrently.

For each type of user request, user performance goals, \( G(i) \)'s, are identified. If there is only one measure of performance for a given user request, e.g., \( R_j(i) \) is only measured in time, then only one \( G(i) \) is identified, in the same dimension of time. If more than one measure is used, then for each measure there is a corresponding \( G(i) \). For example, if a user request type is to be evaluated on both time and cost, then one set of \( R_j(i) \)'s will be collected and evaluated in units of time, and one set of \( R_j(i) \)'s will be collected in dollars and evaluated against a \( G(i) \) expressed in dollars. For the purposes of discussion, however, all measures will be identified generically as \( R_j(i) \) with an associated goal, \( G(i) \), with the understanding that a given user request may have several performance measures.

The user's measure of the achievement of this goal is the discrepancy, \( D(i) \), between the user's goal expectation and the actual system performance

\[
D(i) = G(i) - \sum_{j=1}^{N} R_j(i) \tag{1}
\]

This formulation establishes a direct, measurable relationship between the user-oriented evaluation, \( D(i) \), and system-oriented performance, \( R_j(i) \). \( R_j(i) \) is not only a measure of the activity's performance, but also a measure of the activity's associated system services' performance, over which the analyst is assumed to have control.

The MGP formulation is based on the general measure in equation (1). We can replace \( D(i) \) with \( (D^- (i) - D^+(i)) \), where \( D^- (i) \) indicates the level
Evaluating Information Systems

of overachievement and $D^+(i)$ indicates the level of underachievement of goal $G(i)$. This yields

$$\sum_{j=1}^{N} R(j(i)) + D^-(i) - D^+(i) = G(i)$$

(2)

The objective of MGP, like that of the user, is to minimize total system discrepancy, $\lambda$, over all $M$ goals, i.e., assuming equal worth among goals

$$\text{Minimize } \lambda = \sum_{i=1}^{M} (D^+(i) + D^-(i))$$

(3)

Note that $\lambda$ is a minimization from both above and below the $G(i)$'s.

Let $B_j$ be the decision variable associated with each activity $A_j$. It is interpreted as the performance change indicated for activity $A_j$ to reduce total system discrepancy $\lambda$ (e.g., $B_j = .75$ implies that a 25% reduction is required). The values of the $B_j$'s are determined by the MGP procedure as it attempts to minimize $(D^-(i) - D^+(i))$. Thus, the basic MGP constraint equation for one user is

$$\sum_{j=1}^{N} (R_j(i)B_j) + D^-(i) - D^+(i) = G(i)$$

(4)

Evaluation of a total information system is for all users, each of whom may have more than one goal, and furthermore, a given activity may be in more than one user request sequence. Thus, equation (4) must be expanded to cover $M$ user goals and $N$ activities. Furthermore, MGP requires a goal hierarchy of the discrepancies be built based on priority levels and weightings within priority levels. To accomplish this, each discrepancy, $D(i)$, is associated with a priority level, $F(i)$, and weight within that level, $P(i)$. The priority level is an ordinal ranking scheme such that a discrepancy (goal) at $F(i) = 1$ is immeasurably more important to minimize than a discrepancy (goal) at $F(i) = 2$.

Within a given priority level, discrepancies are weighted according to their importance, relative to other discrepancies at that level. The value of $P(i)$ may be any positive real number and may be normalized to the maximum weighting per level if desired. It should be noted that at a given priority level, all discrepancies must have the same unit of measure, but different levels can have different measures. The general formulation of the information system evaluation problem is, thus

$$\text{Minimize } \lambda = F^+P^+D^+ + F^-P^-D^-$$

(5)

subject to

$$R\beta + D^- - D^+ = G$$

where

- $R$ = array of current activity performance levels
- $\beta$ = array of $B_j$'s
- $D^+, D^-$ = arrays of positive (negative) discrepancies
- $G$ = array of user target goals
- $F, P$ = arrays of priority factors and weights

In order to complete an operational formulation of the information system design problem, several additional constraints were incorporated to limit the range of the $B_j$'s. These constraints, called feasibility constraints, provide upper and lower bounds on the $B_j$'s, which are interpreted as bounds on the degree of feasible or allowable change in performance for $A_j$. For example, if $A_j$ can be only performed twice as fast, i.e., its performance level reduced 50%, $B_j$'s lower bounds would be .5. An initial default value of 1.0 for each $B_j$ is also specified. The final formulation of the information system evaluation problem in MGP form is presented in Appendix A.

The solution to this formulation produces values for the $B_j$'s that indicate what the performance level of the associated activities should have been to minimize total system discrepancy, i.e., achieve the user goals. Because of the basic conflict between user and system goals, and the likelihood of internally conflicting goals, an optimal solution is remote. MGP, however, is designed to characterize such situations and produce a satisfactory solution, i.e., a non-optimal yet feasible solution. It should be emphasized that these interpretations of individual $B_j$'s should be taken in the context of the entire set of $B_j$ values.

**Application**

The following application, although hypothetical, is typical of the information system evaluation
problem faced by analysts. Specifics of an operational hardware and software environment are assumed to provide a reasonable setting. Performance data was obtained from a simulation model of the example. The simulative tool employed was the Information Processing System Simulator (IPSS).\(^1\)

The information system to be analyzed is a disk-based online system, which accesses data such as bibliographic abstracts, credit data, or inventory levels. A user of this system submits a request to the system, it is analyzed by the system, the appropriate record is retrieved from the database, and then the record is sent back to the user. It is also assumed that this database system requires an internal control feature of a front-end security screening to prevent unauthorized access to the database. The hardware characteristics assumed are IBM 2314-like disk drives and a teleprocessing line of 3600 baud. Since I/O is the primary time constraint, CPU time was considered negligible.

The data accessed in this example, credit data, resides in a hierarchic database (see Figure 1). The software hierarchy for this example is as follows. The top level index file (INDEX1) contains an entry for each individual credit item in the database, referenced by individual name/credit item, e.g., Jones/loan. The second level index (INDEX2) has a corresponding entry for each item in INDEX1 but it references an internal document number. This number points to the bottom level of the database which contains the actual credit records (DATABASE). Another file is required for the security function, a list of authorized users (AUTH). The assumed characteristics of all these files are given in the table below.

The model of this system follows the breakdown of these files and indices. Accessing the AUTH file is characterized by a security function (Service S1), and by a safeguarded, non-shared disk access routine solely for the AUTH file (Service S5). Accessing the other layers of the hierarchy are characterized by Services S2, S3, and S4.

There are types of users of this information system. The first user type, USER1, needs to traverse all levels of the database in search of an individual credit item (Services S2, S3, S4). As an example, a bank loan officer may need to examine a prospective applicant's bank balance. The second user class, USER2, only has to reference the highest index, INDEX1, to determine the existence of a given credit item in the database (Service S2). For example, an information specialist may be examining the growth of the database and needs to know only the number, not content, of items in the database. The third user class, USER3, requires the actual credit record, but already has the internal document number, such as during an update, and can go directly to INDEX2 (Services S3, S4). All users must go through the security function (Services S1, S5).

\(^1\)The Information Processing System Simulator (IPSS) is a special purpose, discrete event simulator the development of which was conducted with the support of the National Science Foundation, initially under Grant No. GN-36622 and was continued under STS-5-21643. (See [3, 4]).

### Table 1. File Characteristics of Example System

<table>
<thead>
<tr>
<th>File Name</th>
<th>Number of Records</th>
<th>Logical Record Length In Bytes</th>
<th>Blocking Factor</th>
<th>Physical Record Length In Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVCY</td>
<td>1000</td>
<td>100</td>
<td>7</td>
<td>700</td>
</tr>
<tr>
<td>LOG</td>
<td>10000</td>
<td>25</td>
<td>28</td>
<td>700</td>
</tr>
<tr>
<td>TITLE</td>
<td>10000</td>
<td>12</td>
<td>180</td>
<td>2160</td>
</tr>
<tr>
<td>ID</td>
<td>10000</td>
<td>12</td>
<td>180</td>
<td>2160</td>
</tr>
<tr>
<td>BIBLIO</td>
<td>10000</td>
<td>400</td>
<td>15</td>
<td>6000</td>
</tr>
</tbody>
</table>

64 MIS Quarterly/March 1982
Figure 1. Model of Example System
Using the activity creation process of this approach, four activities can be identified. Since the security disk access function (S5) is performed only at the request of the global security function (S1), one activity, A1, can be defined for the complete security function (S1 + S5). The other invocations of services are sequential due to the logic of the hierarchy. Thus, each service S2, S3, and S4 can be associated with a unique activity A2, A3, and A4. The result is the following functional definition of each type of user request:

\[
\text{USER}_1 = f(A1, A2, A3, A4) 
\]

\[
\text{USER}_2 = f(A1, A2) 
\]

\[
\text{USER}_3 = f(A1, A3, A4) 
\]

In order to transform the information system model above into a form amenable to the MGP procedures, constraints on objective function and a goal structure need to be established. It is assumed only one metric is used and that each user class, USER_i, has a goal, G(i), in the same metric as R_j(i). Thus, the user constraints are:

\[
\begin{align*}
R_{11} \beta_1 + R_{21} \beta_2 + R_{31} \beta_3 + R_{41} \beta_4 + D(1)^- - D(1)^+ &= G(1) \\
R_{12} \beta_1 + R_{22} \beta_2 + D(2)^- - D(2)^+ &= G(2) \\
R_{13} \beta_1 + R_{33} \beta_3 + R_{43} \beta_4 + D(3)^- - D(3)^+ &= G(3)
\end{align*}
\]

For simplicity, the discrepancies were placed in only one priority level and were given equal weightings (P(i) = 1.0).

The measure of system response to user requests was taken to be response time. Thus, each R_j(i) is in time units and G(i) likewise. A target response time of 3 seconds (3000 milliseconds) was chosen for each user request. In an operational setting the upper and lower bounds on \( \beta \) corresponding to the acceptable range of changes to A_j would be determined by management. For this example, the \( \beta \) bounds were set at .2 and 5.0. The resulting MGP formulation is shown below in (12).

The final point about the basic information system model is the set of design alternatives. These were determined by the hardware configuration, the actual hardware devices, and the data set characteristics. Only discrete changes in these areas were employed to simplify illustration. Table 2 lists the types of design alternatives that were considered. Alternatives for both increasing and decreasing an activity's performance time were possibilities. As seen in Table 2, alternatives to increase activity time have an implicit effect of reducing costs. Thus, even in this single metric example, a second measure of cost can be taken into account. It should be re-emphasized that the objective in equation (12) is not time minimization alone, but the minimization of the time difference from the goal G, from both above and below.

\[ \text{Minimize } \lambda = F_1 \left( \sum_{i=1}^{3} (D_i^- + D_i^+) \right) + F_2 \left( \sum_{i=1}^{4} (e_i^+ + e_i^- + \lambda_i^+ + u_i^-) \right) \]

\[
\begin{align*}
\text{s.t. } & R \beta + D^- - D^+ = G \\
& \beta + E^- - E^+ = \lambda \\
& \beta + L^- - L^+ = L \\
& \beta + U^- - U^+ = U \\
\end{align*}
\]

where \( R = \begin{bmatrix} R_{11} & R_{21} & R_{31} & R_{41} \\ R_{12} & R_{22} & 0 & 0 \\ R_{13} & 0 & R_{33} & R_{43} \end{bmatrix} \)

\[
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\end{bmatrix} = \begin{bmatrix} 3000 \\
3000 \\
3000 \\
\end{bmatrix} \quad \begin{bmatrix} L \\
U \end{bmatrix} = \begin{bmatrix} 2 & 5 \\
2 & 5 \\
2 & 5 \end{bmatrix} \]
Table 2. Design Alternatives

<table>
<thead>
<tr>
<th>To Increase Activity Time</th>
<th>To Reduce Activity Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>—Move data sets to slower cheaper devices</td>
<td>—Move data sets to faster devices</td>
</tr>
<tr>
<td>—Reduce blocking factors on data sets to make required buffer size smaller</td>
<td>—Increase blocking factors on data sets</td>
</tr>
<tr>
<td>—Reduce the speed of the TP which may reduce TP cost</td>
<td>—Move data sets closer together on same disk to reduce seek times</td>
</tr>
<tr>
<td>—Move data sets to same disk which increases contention, but may save on cost of disk</td>
<td>—Increase the speed of the TP line</td>
</tr>
<tr>
<td></td>
<td>—Move data sets to different disks to reduce contention</td>
</tr>
</tbody>
</table>

Interpretation and Use

The User Goal Evaluation Stage produces three sets of measures to aid the analyst in the Design Evaluation Stage. They are current design evaluation, design alternative identification, and design alternative evaluation. Each set of measures individually provides insight into the performance of the system, and when viewed collectively, provides a systematic approach to information system evaluation.

The first set of measures produced by this approach evaluates the current information system design with respect to the individual user goals and the entire goal structure. Forcing the formulation to yield \( \beta_j \) = 1 for all activities essentially sets the formulation to the current configuration (since \( \beta_j = 1 \) implies that the required level of performance equals the current level). This results in the determination of \( D^+ (i) \) and \( D^- (i) \) for all goals in the current design.

This evaluation of individual goals is then used to evaluate the global objective function. This objective function can be a combination of goals at different priority levels. Thus, the result is a vector of collective goals' measures, with an entry for each priority level, \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k) \) for \( k \) priority levels. (Note: For the remainder of this article, all references to the objective function will be denoted \( \lambda \), with the understanding that the context can apply to each individual priority level, \( \lambda_k \).)

The second set of measures is the solution vector of \( \beta_j \)'s. They are called decision or control variables to emphasize the fact that they represent the aspects of the information system design under control of the analyst. Implicit in the assumption of linear performance is the further assumption that an activity is an indivisible unit. For example, one does not use half a compiler or one and a half job schedulers. The analyst has under control, however, the operating characteristics of the activity, e.g., its queue discipline, number of servers, type of function. Therefore, in order to achieve the indicated \( (\beta_j)^* \) 100% performance level for an activity, the analyst can modify the operating characteristics of that activity along the lines of the alternatives listed in Table 2.

Thus, \( \beta_j \) is interpreted as the level of performance of activity \( A_j \) required to satisfy the global objective function, relative to the current level of performance. In other words, \( \beta_j = .80 \) implies that 80% of the current level must be achieved. If time is the measure, this translates into making design modifications to the activity such that the resulting operating time is reduced by one fifth. If storage is the measure, then program size must be reduced accordingly. Similarly, for a \( \beta_j \) greater than one, the resultant change to the operating characteristics of activity \( A_j \) should increase the performance level, e.g., use up slack and take more time.
The MGP formulation permits all $\beta_j$'s to be changed simultaneously. Thus, although an activity may be associated with many different user requests, the value of $\beta_j$ produced by the MGP solution is with respect to all these associated users. The effect of changing the current level of performance for an activity on the global objective function, i.e., its effect on different activities, goals, priority levels, and weights, has been automatically accounted for by MGP procedures.

Therefore, the output of the MGP problem formulated in equation (5) is a vector of $\beta$ values, indicators of the levels of performance required to satisfy all user goals. The Design Evaluation Stage uses these values to determine which activities to change ($\beta_j \neq 1.0$), which not to change ($\beta_j = 1.0$), and the direction of that change ($\beta_j$ value itself). The determination of what specific change to make is made in conjunction with outputs from the System Evaluation Stage and analyst experience.

The third set of measures evaluates design alternatives indicated by the $\beta_j$ values of the second set. The result of this evaluation is a list of the activities ordered by their ability to improve performance through a change in their design. This ordering is based on the marginal contribution of each activity $j$ to the minimization of overall system performance $\lambda$ denoted as $\delta \lambda_j$, given that the activity was modified according to its $\beta_j$ value.

The calculation of $\delta \lambda_j$ is made for each activity defined for the current system. Based on the value of $\delta \lambda_j$ an ordered list can be assembled, ranging from the most negative value to the most positive value. The most negative value indicates that modifying the activity associated with $\beta_j$ offers the possibility for the greatest reduction in total system discrepancy. The activity at the other end of the list offers the possibility for the least reduction in $\lambda$. 

If there are no $\delta \lambda_j$ values less than 0, then there may be no activities that can be changed to reduce $\lambda$.

This is very useful to the analyst, because between iterations of a given design every desired design change cannot be made. In addition to insufficient time to implement all changes, the cost to physically make all changes will probably be prohibitive and the ability to isolate the effect of an individual change will become extremely difficult. Thus, the analyst can benefit from having a facility to rank design alternatives in the order of their potential performance improvement capabilities so that the analyst need spend time and money on only the potentially most effective alternatives.

### Analysis of Example System

As mentioned above, a simulation model of this system was built. Executions of this simulation produced not only normal system statistics (resource queueing and utilization) but also the $R$ matrix. This matrix was then entered in the Goal Evaluation Stage. Depending on the results of this analysis, the original simulation model was modified to reflect the indicated change, and then re-entered for a system evaluation, completing the iteration.

Given formulation (5) of the example model, Table 3 presents the results of the first two evaluation iterations. As described in the previous section, the application of the Goal Evaluation Stage produces three sets of measures. The evaluation of the current design for iteration 1 yields a measure of total discrepant performance, $\lambda$, (2406 time units in this case). The individual discrepancies show that for the user types 1 and 3, the current system design took longer than desired to respond to the request, whereas the system responded to user type 2, faster than desired.

For the example system in iteration 1, the values of $\beta_3 = 1.0$ and $\beta_1 = 1.005$ imply that for the current system, activities $A_1$ and $A_3$ are performing at or very close to satisfactory levels. Activities $A_2$ and $A_4$ on the other hand, are performing at very unsatisfactory levels. Since the individual $\beta_j$ value indicated in which direction a change should be made, $A_2$ should be lengthened (i.e., it currently contains slack) and $A_4$ needs to be reduced. The relative distance a particular $\beta_j$ is from 1.0 does not alone provide enough information to determine which activity to modify first.

The values of $\delta \lambda_j$, however, provide such information. $A_3$, because it is performing at a satisfactory level, need not be changed, and there would...
Table 3. Raw and Calculated Data for Example System

<table>
<thead>
<tr>
<th>Input Data:*</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R = [2280 40 44 1695]</td>
<td>R = [2218 36 41 404]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R = [2770 70 0 0]</td>
<td>R = [2352 64 0 0]</td>
</tr>
<tr>
<td></td>
<td>R = [2403 0 37 1747]</td>
<td>R = [2298 0 34 424]</td>
</tr>
</tbody>
</table>

A. Current Design Evaluation

\[\lambda = 2406\] \hspace{2cm} \[\lambda = 1128\]

\[D^+ (1) = 1059\] \hspace{2cm} \[D^+ (1) = 0\]

\[D^+ (2) = 0\] \hspace{2cm} \[D^+ (2) = 0\]

\[D^+ (3) = 1187\] \hspace{2cm} \[D^+ (3) = 0\]

\[D^- (1) = 0\] \hspace{2cm} \[D^- (1) = 301\]

\[D^- (2) = 160\] \hspace{2cm} \[D^- (2) = 583\]

\[D^- (3) = 0\] \hspace{2cm} \[D^- (3) = 244\]

B. Design Alternative Identification

\[\beta_1 = 1.005\] \hspace{2cm} \[\beta_1 = 1.20\]

\[\beta_2 = 3.31\] \hspace{2cm} \[\beta_2 = 2.74\]

\[\beta_3 = 1.00\] \hspace{2cm} \[\beta_3 = 1.00\]

\[\beta_4 = 0.33\] \hspace{2cm} \[\beta_4 = 0.49\]

C. Design Alternative Evaluation

\[\delta \lambda_1 = -10\] \hspace{2cm} \[\delta \lambda_1 = -657\]

\[\delta \lambda_2 = -66\] \hspace{2cm} \[\delta \lambda_2 = -174\]

\[\delta \lambda_3 = 0\] \hspace{2cm} \[\delta \lambda_3 = 0\]

\[\delta \lambda_4 = -2153\] \hspace{2cm} \[\delta \lambda_4 = 422\]

*For each and all iterations, several, not one, simulation runs were made in order to give more consistent, reliable \(R_{ij}(i)\) values. Thus, each \(R_{ij}(i)\) value is a mean value across simulation executions, not a point-estimate of only one particular execution.
Figure 2. Results of Design Decision Sequence
not be any marginal contribution to reducing \( \lambda \) (hence, \( \delta \lambda_3 = 0.0 \)). Similarly, \( \beta_1 \) for \( A_1 \) implies only a small change and, thus, \( \delta \lambda_1 \) is also small. The main decision then for the analyst in iteration 1 is to decide which of the remaining activities to modify, \( A_2 \) or \( A_4 \). Comparing \( \delta \lambda_2 \) and \( \delta \lambda_4 \) provides guidance. \( A_4 \) should be modified, and since \( \beta_4 < 1 \), it should reduce its current performance level.

This guidance was followed in the model in iteration 2. Activity \( A_4 \) was modified to reduce its performance level (processing time) by moving the database from the current slow disk device (IBM 2314-like) to a faster device (IBM 3330-like). The effect can initially be seen by comparing the \( R \) matrices for iterations 1 and 2. The column of \( R_4(i) \) is greatly reduced in iteration 2 from that in iteration 1. \( \lambda \) has also been greatly reduced which implies that the overall system evaluation was indeed improved by following the \( \beta_j \) and \( \delta \lambda_j \) interpretations. The individual discrepancies show a shift to a design which satisfies all responses too fast (on the average). Analysis of the \( \beta_j \) and \( \delta \lambda_j \) measures now reveals that the burden of performance improvement has been shifted to activity \( A_1 \).

This process of basing decisions on \( \delta \lambda_j \) and \( \beta_j \) was continued for several more iterations. The design modifications indicated for \( A_1 \) were followed in iteration 2 (make \( A_1 \) longer). Activity \( A_1 \) was modified by allowing the AUTH file to be non-contiguous, thereby increasing response time. Overall \( \lambda \) was reduced to 500 time units from 1128. (Remember \( \lambda \) includes both over and underachievement.) Figure 2 shows how over the entire set of iterations for this model, use of \( \delta \lambda_j \) and \( \lambda_j \) consistently reduced \( \lambda \). Two other information systems were simulated and analyzed using this approach. Applying the same \( \delta \lambda_j \) and \( \lambda_j \) based decision rules, they achieved markedly similar effects on \( \lambda \) over a sequence of design decisions. The results of these analyses are presented in full in [2].

The freedom of direction in design represented by the \( \delta \lambda_j \) and \( \beta_j \) analysis is an advantage. This approach allows the entire set of design alternatives to be available throughout the design and evaluation process. Through appropriate analysis of the \( \beta_j \)'s and \( \lambda \), on a given iteration, this set is reduced to the most beneficial subset for consideration by the analyst. On subsequent iterations, however, the complete set of alternatives is available again. Such flexibility provides for continued creativity for the analyst and lessens the chance of locking the analyst into an unfruitful path because of errors early in the sequence of design iteration.

Since MGP is designed to characterize multi-dimensional situations, it provides the analyst with a more natural evaluative tool from two perspectives. First, the priority levels facilitate the construction of a ranked hierarchy of user goals instead of forcing the analyst into one evaluative level. Second, multiple performance measures do not have to be transformed into a single, aggregate measure for a solution to be possible as is the case in other approaches, particularly optimization techniques.

As an example, consider an information system that is to be measured by response time, user satisfaction, and operating cost. In other approaches, a single measure, such as cost, would have to serve as a common denominator. Thus, a response time goal may have to be expressed in terms of dollars lost per second of delay, times the time delay as opposed to a target time in seconds. Similarly, the user satisfaction goal may have to be transformed into an artificial dollar value of this intangible area instead of allowing the goal to be expressed in utiles or some other subjective scale. The current approach, however, allows each measure to remain in its appropriate dimension (unit) and allows for goals for each measure (dimension) through the priority levels.

Conclusion

The approach described in this article provides a framework for evaluation of information system from a multi-criteria perspective. It is not meant to be an optimizer, yielding "the solution." Being based on MGP, this approach accepts the infeasibility of an optimal solution for a conflicting goal structure and produces a satisfactory result. It is a tool to gain insight into the behavior of information systems. Because this approach, like any model, is an abstraction of reality, it has its limitations. These constraints, however, can be avenues of extensibility if applied appropriately.
The first area of concern for the user is the goal structure of the objective function. For a given sequence of applications of the approach, the defined goal structure remains constant, allowing the MGP mechanisms to manipulate the $B_j$'s to achieve the objective function. When compared to the dynamic decision making environment that exists in operational information systems, however, this may seem artificial. Admittedly, management cannot live with the same goal structure forever.

The approach, however, does not imply that this is the case. A sequence of evaluation iterations is only the first step in analysis. By performing sensitivity analysis on the objective goal structure the analyst can discover much about the goals of the information system user. The sensitivity of the MGP solution to changes in the goal structure is another area of investigation. Internal political considerations such as ranking applications (users) according to their importance can be characterized by the formulation through the definition of the goal structure. If the user is only beginning the evaluation process, the initial goal structure may be a guess, at best. One can then make changes in the goal structure and use the resultant feedback to refine and fine-tune an acceptable and realistic goal structure.

Another aspect of the formulation of interest to the user is the quantification of the goals, $G$. As in the construction of the goal structure, this process may seem foreign and artificial to the information system user. But, at some point, such a determination must be made, sometimes even on the intangible aspects of information systems. Performance evaluation via any method, not just the approach described here, depends on established goals for comparison.

Again, this approach offers the analyst and the user a tool to evaluate different sets of targets and gain a "feel" for the appropriate levels of performance. If system measurement data is produced from a simulation package, this process can iterate to produce a satisfactory set of goal targets for the entire set of constraints. The underlying point to be made about both the goal structure and the goal targets, however, is that the approach fixes them for only one iteration, not forever. The user of this approach can react to the dynamic changes in goals and performance levels as quickly as they occur.

The final element of the formulation is the definition of the performance constraints themselves. In the current implementation of the approach, the relationship between activities in a system response sequence is linear. Non-linear relationships and parallel operations can be linearized via activities if the level of aggregation, and the corresponding level of control, is sufficiently high. If greater detail in modeling is required in these situations, however, then other approaches may need to be investigated. For example, if parallel operations must be modeled, an integer goal programming approach may be able to be formulated where each potential path is assigned an integer variable. The formulation of constraints then becomes a summation of integer variables. The decision to model at such detail is the classical choice in modeling between greater accuracy and the marginal benefit to decision making.

An immediate extension of the current approach involves the analysis of additional measures of performance. A logical first choice is cost. Each activity $A_j$ could be considered a cost center in addition to its current identification as an operational unit. A measure $R'_j(i)$ can be defined as the mean cost charged to user $i$ for the use of activity $j$. The corresponding goals values, $G'_j(i)$, could be expressed in terms of cost per run. The approach could then be used to determine (1) the cost per user of their current workload, (2) the total revenues to the data processing department, (3) the effectiveness of a charging algorithm, or (4) the impact of changing the workload distribution or the charging scheme on the system and the user. If the target goal values are based on a user's data processing budget, then an objective function could be constructed to identify budget overruns and keep performances within budget constraints. This cost measure, $R'_j(i)$, could also be combined with the current defined measure of $R_j(i)$ to produce a model that aids the analyst in producing a satisfactory cost-effective information system design.

Thus, the approach presented in this article provides a vehicle for information system analysis and evaluation. It allows for the identification of deficiencies in current performance and the investigation of possible alternatives to these problems. It is flexible enough to be able to handle the dynamics of the information system environment and map their influence into the
Evaluating Information Systems model. It provides a tool for analysis of tradeoffs between goals, applications, and performance. Furthermore, it allows for the investigation of the impact of environmental and design policy decisions on information system performance and user goal attainment.

Appendix A

Minimize \( \lambda = \)

\[
\begin{align*}
& F^+_D \cdot P^+_D \cdot D^+ + F^-_D \cdot P^-_D \cdot D^- \\
& + F^+_L \cdot P^+_L \cdot L^- + F^-_U \cdot P^-_U \cdot U^- \\
& + F^+_E \cdot P^+_E \cdot E^- + F^-_E \cdot P^-_E \cdot E^-
\end{align*}
\]

s.t.

\[
\begin{align*}
R \cdot \beta + D^- + D^+ &= G \\
\beta + U^- - U^+ &= U \\
\beta + L^- - L^+ &= L \\
\beta + E^- - E^+ &= \bar{1}
\end{align*}
\]

Where

- \( R \) is the matrix of performance levels
- \( \beta \) is the array of performance change indicators
- \( G, U, L, \) and \( \bar{1} \) are the arrays of goals for the user criteria and the respective \( \beta \) constraints
- \( D^\pm, U^\pm, L^\pm \) and \( E^\pm \) are the arrays of positive and negative discrepancies from the respective goals
- \( P^\pm_D, P^\pm_U, P^\pm_L, \) and \( P^\pm_E \) are the arrays of penalties for the associated discrepancies
- \( F^\pm_D, F^\pm_U, F^\pm_L, \) and \( F^\pm_E \) are the arrays of priority levels for the associated discrepancies

Note:

- \( \beta + U^- - U^+ = U \) is the generalization of the upper bound constraint
  \[ \beta_j + u_j^- - u_j^+ = U_j, \text{ where } 1 \leq U_j \]
- \( \beta + L^- - L^+ = L \) is the generalization of the lower bound constraint
  \[ \beta_j + l_j^- - l_j^+ = L_j, \text{ where } 0 \leq L_j \leq 1 \]
- \( \beta + E^- - E^+ = \bar{1} \) is the generalization of the activity default value constraint
  \[ e_j^- - e_j^+ = 1 \text{ and } (1, 1, ..., 1) = \bar{1} \]
Evaluating Information Systems

References


About the Author

John Chandler is currently an Assistant Professor in the Department of Accountancy at the University of Illinois at Urbana. He completed his graduate work in Computer and Information Science at The Ohio State University in 1977. He has had several research projects with government, military, and commercial organizations. His research interests generally are applications of information systems in business and society, and, specifically, performance evaluation of information systems.