Discussion Paper No. 342

THE DESIGN OF A CORPORATE PLANNING SYSTEM SIMULATOR

by

Edward A. Stohr and Mohan Tanniru

Northwestern University
Graduate School of Management
Evanston, Illinois 60201

August, 1978
THE DESIGN OF A CORPORATE PLANNING SYSTEM SIMULATOR


Edward A. Stohr and Mohan Tanniru
Northwestern University
Graduate School of Management
Evanston, IL 60201

ABSTRACT

The integration of various functional activities in an organization can be accomplished by using a hierarchical data model based on the firm chart of accounts. Viewing the entire accounting system as a submodel and the information requirements of each decision-making unit as a submodule, modularity in planning can be achieved. Once the total configuration of the system and the accounting transactions are stored in a data base, a simple planning language can be developed to simulate the performance of the total system or any subsystem.

1. INTRODUCTION

In recent years there has been a great growth in the use of corporate planning models. A number of specialized planning languages have been developed to make the task of constructing these models easier (4), (5) and (13). In addition, information retrieval systems with sophisticated report writers, graphic packages and forecasting models have been constructed to assist in management in their planning function (7) and (16). Concurrently management scientists have developed models and built computerized support systems in the functional areas of management such as marketing, finance and production. While many of these individual efforts have been successful (12) there have been few attempts to provide a comprehensive and self-consistent integration of these models into the overall planning and budgeting system of the firm. Thus the accounting framework which is the common language of 'corporate' models and the various models in different functional areas remain unconnected.

In (1) the idea of using a graphical-hierarchical representation of the corporate financial system to provide an easily implementable computerized framework in which optimization and other planning models could be coordinated was introduced. This was used to classify groups of representative planning models according to their broad and depth of coverage. Various applications to common planning tasks such as budget preparation and the projection of financial statements were also described. In this paper we describe the design of a computerized planning system based on this technology. We concentrate on the methods used to support a wide variety of models and to integrate their outputs into a common accounting framework.

The organizational setting we assume is as follows. The planning system is designed to support management decision-making and to help in the coordination of the budgets for various organizational units. Model builders and systems analysts are responsible for the logical design and maintenance of the system. The planning system contains a central module called the 'central financial model' (CFM) which at any time contains the financial and budgeting information for the firm in a manner consistent with the firm chart of accounts. The CFM can be used to display the past history of various financial and economic time series. It also contains information concerning the adopted budgets and plans and has the capability of automatically projecting proforma financial statements by time-series and/or regression methods. Various simulation and operations research models can be used in conjunction with the forecasting models to arrive at a set of 'official' projections.

The planning system also contains modules which support the planning function at a more detailed level. These are categorized by functional area and organizational units. For ease of reference these modules will be called 'functional area models' (FAMS). It should be pointed out that we use the word 'model' in its broadest sense to mean any subsystem which provides computerized inputs to the CFM. This might include, for instance, subjectively estimated plans and budgets produced by lower organizational units.

The planning system is designed to assist the manager by providing: 1) retrieval of historic and projected information concerning the state of the organization, in the form of reports and simulations of the various functional units in the organization allowing both 'top-down' and 'bottom-up' planning, and 2) automatic projection of proforma financial statements. It assists the model builder (system analyst) by providing: 1) a simple method of expressing the logic of the planning system as data which can then be operated on to provide different configurations of multiple models with the output of one model being used as input to other models, 2) a comprehensive system for maintaining the interconnections between models and allowing their
In section II we briefly describe the planning sys-
tem, and in section III the hierarchical data model
on which it is based. In section IV we describe
the methods used to support the integration of
models with the central financial plan. Section V
describes aspects of the computer implementation
of these integrative methods. The logic of the con-
sistency checks provided by the system is described
to detail and illustrated using a typical financial
planning model, Known 8). In section VI we out-
line some of the methods used to help assemble the
construction of models. In particular the rela-
tionship between the linear hierarchical system of
the CPN and the Known model is described.

II. STRUCTURE OF THE DECISION SUPPORT SYSTEM

The planning system is currently being implemented
in APL. Its major software components are:

1. The System Manager (SM)

This contains the software for generating, managing,
allocating and retrieving the accounting-related logic
of the corporate planning models (both CPN and
HYMN). This module is based on the linear hierarc-
chical model, explained above and is outlined in
Section III. For a more detailed explanation see
section 11.

2. Data-base Management System

This is a general purpose software based on the EDBS
package (27). It may be accessed by the user if
required from any other module (including HYS
although HYS contains its own mechanism for retrieval
of historic and predicted time-series associated
with the accounting system). A network data model
is used by the planning system to maintain the rela-
tionships between model data inputs, model state-
ments (simulation, regression and mathematical
programming tableaux), and model results. The
relationships between multiple interconnected models
are also maintained (16).

3. Simulation Planning Language

This module is used for stating regression, time-
series and corporate simulation models. Its cap-
abilities are similar to those of a number of sim-
ilar languages (3), (13).

4. Forecasting Module (Time-Series and Regression)

5. Operations Research Algorithms and Statement

Generators (Mathematical Programming, and Con-

6. Model Data Input and Logic Specification Sub-

system


B. Graphics Package

9. Planning System Data Dictionary

In this paper we will concentrate on describing the
interface between 7, 3, 4 and 5.

III. THE HIERARCHICAL DATA MODEL

Accounting transactions are used to report the
monetary effects of different activities within the
firm and in the final analysis financial statements
provide the yardstick for measuring its perfor-
mance. In this section we briefly describe a model for
the firm's financial accounting structure.

Double entry bookkeeping can be described by means
of a tree diagram and a set of directed arrows (3).
The nodes of the tree correspond to the various
accounts. The hierarchical relationships between
the nodes in the tree define the accountant's sys-
tem of classification and aggregation. Figure 1 pro-
vides a sample of a part of a typical balance sheet
accounted together with a list of four typical transac-
tion types.

![FIGURE 1

ILLUSTRATION OF LINEAR HIERARCHICAL ACCOUNTING SYSTEM](image)

<table>
<thead>
<tr>
<th>Balance Sheet</th>
<th>Transaction Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Assets (C)</td>
<td>TA - investment in firm</td>
</tr>
<tr>
<td>Fixed Assets (FA)</td>
<td>TY - purchase of fixed assets for cash</td>
</tr>
<tr>
<td>Current Liabilities (CL)</td>
<td>T3 - short-term borrowing</td>
</tr>
<tr>
<td>Long-term debt (L)</td>
<td>T4 - long-term borrowing</td>
</tr>
</tbody>
</table>

Legend

Note that both balance sheet and income statement accounts can be represented in the tree (see Figure 2). Although the former represent 'stock' variables and the latter 'flows' both types of accounts will be referred to as the 'nodes' of the system. Assuming a state variable, its value at time t will be represented by the vector, b, E R^n. The various types of accounting transaction are represented by directed arcs connecting leaf-nodes of the tree; the arrow node is credited by the transaction and the leaf node is debited. A value associated with each arc represents the cumulative value of all individually recorded transactions of that type for a time period. Assuming that there are n arcs the vector of values associated with

Edward A. Nordin and Mohan Tandri

Northwestern University, Graduate School of Management

The Design of a Corporate Planning System Simulator
the arcs at time \( t \) will be denoted by the 'aggregate transaction vector', \( \mathbf{T}_t \in \mathbb{R}^n \). The states of the system (values associated with the nodes) are changed in each period by the transaction. The effect as the leaf nodes can be summarized by the conventional node-arc 'incidence matrix'. If one also wants to represent the effect on higher nodes in the treestructure, it suffices to add a row for each such node. Each transaction linking two leaf nodes defines a unique loop in the tree and will correspond to a column of the matrix. Thus a complete algebraic representation of the tree requires a \((0, +1, -1)\)-matrix with as many rows as there are nodes in the tree and as many columns as there are feasible transactions between leaf nodes. Finally we need to adopt a sign convention for the direction of the transaction. Our convention is to take debit entries (i.e., increases in the assets accounts or decreases in the equity/liability accounts) as positive and credit entries (i.e., decreases in the assets accounts or increases in the equity/liability accounts) as negative. The remaining matrix will be referred to as the \((0, +1, -1)\) 'system matrix', \( S \). Corresponding to Figure i we have:

\[
\begin{bmatrix}
T_1 & T_2 & T_3 & T_4 \\
A & +1 & 0 & -1 & +1 \\
L & -1 & 0 & -1 & 1 \\
C & +1 & -1 & +1 & -1 \\
S = FA & 0 & +1 & 0 & 0 \\
C & 0 & 0 & 0 & +1 \\
D & 0 & 0 & -1 & 0 \\
E & +1 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Since some of the states are flow items (income statement accounts) the corresponding balance must be recorded each period. Let \( x \) be an \((n \times 1)\)-vector with \( x_i \) the \( i \)-th node is a balance sheet item (stock variable) and zero otherwise. Then balance can be obtained from \( x \) by the linear 'system equation':

\[
\sum_{i=1}^{n} x_i = x_b = x_f + x_l
\]

(1)

This completes the description of the graphical and algebraic representation of the accounting system employed by the BM. Borrowing terms from the database management field, the tree structure and list of arcs is a 'data model' and the firm's chart of accounts expressed in this way a 'schema'. Models introduced by planners to solve particular problems will be concerned with a subset of nodes and transactions. In fact, the viewpoint of a particular model may require a realignment of nodes and a redefinition (aggregation or disaggregation) of transactions. The system's tree and set of arcs used by a model corresponds to the model's 'subschema' in database terms. An essential step in the procedure used to reconcile a model's outputs with the financial accounting system is to provide a mapping between the model's subschema and the official schema. A partly automated procedure for doing this is described in Section IV. The system trees are stored internally using a 'multiattribute' tree structure (12), (13). The system matrices corresponding to the schema and subschemas are not stored but can be generated as required for use by the models. One immediate implication of (12) is that the BM stores all relevant accounting identities. If the initial state of the system is known and the aggregate transactions vector \( \mathbf{T}_0 \) can be estimated, then the projected financial statements can be immediately computed using (1). As another application, in simulation models the model's usually has to write a large number of statements describing accounting identities. Since these are already stored in the schema this task can be eliminated.

IV. INTEGRATION OF MODELS WITH THE CENTRAL FINANCIAL PLANNER

In this section we describe the design features of the planning system that allows the integration of diverse models (FM's). Referring to the system equation (1) our general approach is to use the transaction vector, \( \mathbf{T}_t \), as the means of communication: the inputs and outputs of the model are expressed in terms of the aggregate transaction vector, \( \mathbf{s}^\alpha \), associated with the model subschema and the vector \( \mathbf{s}^\alpha \) is then mapped to the aggregate transaction vector \( \mathbf{s} \) associated with the CFM. Since we wish to accommodate both the 'top-down' and 'bottom-up' approaches to planning we need to provide for a mapping in both directions. We use transactions instead of states as the medium of communication since these are most closely related to decisions and their effect on the state of the system as described by the accounting identities stored in the system. It is possible we wish to reduce the model-builder's model statement task by automatically supplying accounting identities. Our main purpose in this part of the paper is to specify the processing logic required to implement the mapping's between the schema and various subschemas. This will be done algebraically rather than algorithmically. We first clarify the terminology and establish some basic notation. In the next section we describe the planning system logic involved and illustrate its use in the specification of a financial-planning model. The CFM consists of the following software and data:

1. Logical elements of models: these 'models' are used to derive a set of corporate plans, budgets and predicted financial statements. Many different algorithms may be employed (e.g., linear regression, simulation, subjective forecasts). The use of linear regression is illustrated below.

2. The corporate schema \( \mathbf{T}_0 \), where \( \mathbf{T}_0 \) is the set of transaction types (arc) and \( \mathbf{f} \) is the set of state variables (nodes in the tree).
3. Data base for corporate and economic data.

4. Retrieval and display facilities.

An example of a typical CP schema is shown in Figure 2. A FAM consists of:

1. A logical statement of model: e.g., a production, financial, or marketing model using a variety of interconnected operations research algorithms and/or subjective estimates.

2. The submodels (T, B, P) where T is the set of transaction types and B the set of state variables which the FAM "affects".

3. FAM database

4. Retrieval and display facilities.

5. Mapping functions: \( N^T \times N_B \rightarrow N^P \) and \( N^T \times N_B \rightarrow N^P \) where \( P \) is the set of all variables used in the model.

Note that there may be more than one FAM-related to a given organizational function such as marketing and that the model may consist of a number of related submodels employing different algorithms as long as these submodels are operated as a unit.

As an example of a FAM consider the CPNI, Nuthe, Modigliani and Sten (NNSP) aggregate workforce and inventory smoothing model (7). The model variable set, \( V \) is:

- \( P_t \) = aggregate production rate at time \( t \)
- \( W_t \) = workforce level at time \( t \)
- \( I_t \) = net inventory on-hand at time \( t \)
- \( S_t \) = sales revenue at time \( t \)

The submodels and mapping are shown in Figure 3:

FIGURE 3

NNS P MODEL

[Diagram of NNS model showing CPA, PI, CA, OK, and other elements]

FIGURE 4

RELATIONSHIP BETWEEN CPI AND A FAM

[Diagram illustrating the relationship between CPI and a FAM]

The coefficients, \( c_{ij} \), are parameters to be estimated. Referring to Figure 3 it can be seen that, relative to the schema, some nodes in the submodels are aggregation points and some are disaggregation points representing a finer classification. Similarly, some transactions are aggregated and some disaggregated while the transaction CPI/SE occurs in both the schema and submodel.

The mappings \((T, B) \rightarrow (T, B)\) and \((T, B) \rightarrow (T, B)\) will require further rules to be added to the mapping system and these will depend on the accounting system chosen by the firm. One purpose of linking the FAM and CPI in this way is to assist the model-builder in estimating the parameters and exogenous variables required by the FAM. In the NNS case past values of the cost transactions (at least CPI/INV and some aggregation of the others) might be made available for parameter estimation and forecast sales could be supplied whenever it is necessary to run the model.

The relationship between the schema, submodels, and FAM model is depicted in Figure 4.

This provides for the communication necessary in both top-down and bottom-up planning activities. In top-down planning plans and budgets are formulated at the highest level in the organization using a CPI. Budget levels or goals are then sent (via \( \sigma \)) to the FAM’s where more detailed planning is carried out. For example, plans at the top level may be in annual or quarterly terms, these at the FAM level may be for daily, weekly or monthly periods and may take into account the more

Edward A. Scarf and Mohan Tambe
Northwestern University, Graduate School of Management
The Design of a Corporate Planning System
In bottom-up planning on the other hand, the results of the FAN models are made available (via $f^1$) to the CFP which consolidates them into the overall financial projections for the company. Generally an iterative approach employing both top-down and bottom-up planning may be implemented. Finally it should be pointed out that it may not be necessary to implement all the maps shown in the Figure for all FAN's.

The major aspect in building and running a FAN which are of interest to this discussion are as follows:

1. Input of Model sub-hulls ($T^1$, $S^1$)

2. Pre-reconciliation Stage (Mapping ($T_0$, $B$) to ($T^1$, $S^1$))

   The objective of the systems manager (SM) at this stage is to maintain and project time series data stored in the CFP compatible with the variables of the user's model. The model builder can then use the historical and projected data to estimate model parameters and examine historical relationships which may provide guidance in the specification of the model.

3. Model Construction

   The construction of the model consists of making the task of model construction simpler by automatically generating the matrices $E$ of $T$ and $S$ in equation (1) (which together contain all accounting identities) and providing other information such as an integer, linear and goal programming tableau generator which works directly from an algebraic ("identified statement") version of the problem (14, 16).

4. Running the Model

   Data base techniques are used to manage the input and storage of data, model specification, the results of runs and the interrelationships between algorithms employed by the models (16).

5. Post-reconciliation Stage (Mapping ($T^1$, $S^1$) to ($T$, $B$))

   The objective of this stage is to retranslate the model's results into a form compatible with the data stored in the CFP. This allows the user to examine corporate financial statements revised according to the results of his model. Since the model will be concerned only with a subset of the CFP transaction types the others have to be automatically retranslated by the CFP to complete the financial statements.

In this section we outline the method used to establish the relationship between ($T$, $B$) and ($T^1$, $S^1$) and illustrate it using the financial planning model given in Kruse (8). This model is used to determine a firm's short-term cash budget, long-term capital budget and related financial mix. In common with a number of other financial models (8), (9) as well as the HQS aggregate projection model given above, this is a linear quadratic stochastic (LQS) stochastic control model with the familiar "linear decision rule". The model sub-blocks, variables and parameters are shown in Figure 5.

FIGURE 1

Edward A. Short and Mohan Tamhane
Northwestern University, Graduate School of Management
The Design of a Corporate Planning System Simulator
In this section we explain the stages 1, 2 and 5 outlined in the previous section in some detail, since these are the primary means for integration of planning activities. It should be pointed out that either on both phases 2 and 5 may be irrelevant to some planning tasks. Also that in general these stages can only be partly automated since the translation is based on the semantics of the models involved. The role of the system manager is to provide a first approximation of the relationship between the CMF and FAM. The FAN model builder, perhaps working jointly with the CMF model builders and accounting staff, will be required to resolve ambiguities. They are assisted in this task by an automated data dictionary.

To simplify the design of the planning system certain rules are employed which must be followed during the specification of the subschema:

Rule 1: The model subschema can have at most one directed arc in each direction between any two leaf nodes. This can always be achieved by adding sub-classification nodes where necessary (e.g. the nodes GT, IL and SC in the 1985 model).

Rule 2: If the model subschema contains modules which also appear in the schema then the associated nodes are assumed by the DM to represent the same state variables. The meanings of the nodes can be altered at a later stage by the model builder if necessary.

Rule 3: Every path from a leaf node to the root of the model subschema must contain a node which also appears in the schema. This rule can be easily satisfied by including nodes 1 and 1S. Subjects to the above rules the model builder can redefine state variables and their hierarchical relationships.

the subschemas tree generated and scored by the DM (the 'internal' subschemas) may differ from that input by the user but will have the following properties:

Property 1: Each of the transactions defined in the user subschemas will be associated with two leaf nodes, all the schema nodes on the path from this set of subschema leaf nodes to the root of the tree are included. If some of the leaf nodes and/or aggregate points are not known to the schema, the first recognizable node to this path is taken as the starting point for this expansion. The starting point is always assumed by rule 3. Before going to property 2, let us define the least aggregation points (LAP's). LAP's are necessary whenever a user defines a number of new nodes not known to the schema and/or overstates the structure of the existing ones.

Let L(x) represent the set of leaf nodes of the subschema for which the nodes to X are parents and let P(x) define the nodes in the path between X and the root node of the subschema.

The DM can generate L(x) and P(x) for each aggregation point of the subschema input by the user including the root meta. Let us also define L(x) as a measure function, which is 0 if L(x) C L(x) and is 1 if there are no elements in L(x) that are not in L(x). Here L(x) is the set of leaf nodes in the schema for which X is a parent. If X, itself, is not known to the schema, then L(x) is set to 1 to avoid its consideration.

Let us now define a least aggregate point (LAP) as a node x that satisfies the following properties:

1. L(x) = L > 0.
2. [y Y x] P(y) C P(x) and (y) = k
3. [y Y x] P(y) C P(x) and (y) = k

Determination of LAP's will partition the tree such that no flow variable. This is followed by the tree given in the original schema and below the LAP it requires a revision through user interaction.

Property 2: The subschema modified according to property 1 has to be 'balanced' below the LAP's. The 'balancing' here is defined as the inclusion of all the immediate children of a schema node that appears in the modified subschema.

Figure 3 depicting the 1985 model does not satisfy property 1 because, for example, schema node A is not included; it does not satisfy property 2 because schema nodes AN, PPE and NS are omitted from the subschema for CA. Note that the LAP here is G. If 'GT' was defined by the user in his subschema as, say, "(C)", then the LAP would have been 'GT'. The subschema for the 'browse' model (Figure 5) satisfies property 1, but not property 2. The LAP here is SC.

For future reference let us define Y2, as the set of subschema nodes input by the user and revised by property 1 with the revision occurring above the LAP's. Let us define N2 to include nodes added to satisfy property 2. Y2 is then N2 U Y2.

STAGE 1: INPUT OF MODEL SUBSCHEMAS (Y2, Y2). (Input of State Variables, Y2)

The user first inputs the names of the nodes used by the model and indicates if they are stock or flow variables. This is followed by the hierarchical relationships as depicted in Figure 5. Simple interactions between the model builder and the system manager are shown in (T) and (L). The nodes used directly by the browse model are


Note that all these nodes also appear in the schema with the exception of VC. After the subroutines
Subschema | Schema
---|---
ID | ID
SK | SK
DE | DE
IF | IF
g1 | g1

Nodes with no associations: g5, g6, A, D, E, F, G, O1, O2.

If the planning system cannot identify a subschema node it requests that the relationship be defined by the user. In the example VC is such a node.

The model builder then has the following specification alternatives:

1. VC = g5, g6, A, D, E, F, G, O1
2. VC = g5, g6, A, D, E, F, G, O1
3. VC = g5, g6, A, D, E, F, G, O1

In the first specification 'variable costs' is defined as an aggregation of the various cost accounts and Other Income, O1. In the second O1 is combined with sales revenue to produce a redistribution of the subschema node SK as 'total income'. In the third specification O1 is not relevant to the model.

This node is then included in the set (B) for use in post-reconciliation. During the post-reconciliation phase it will be recognized by the system as a child of O1.

More generally, if the model subschema involves disaggregation as well as aggregation as many-to-one mapping is obtained. Thus in the HUMS model we have: O1, O2, SK = O1.

To test that the association is complete the system manager determines (i) that all model subschema leaf nodes have been assigned and that (ii) an exact partition of L(B) has been obtained consisting of L(E) and the sets defined by the above mapping.

(c) Develop the association between the Model Transaction Set T1 and the Schema Transaction Set T1.

The system manager scans T1. Each subschema transaction identifies a pair of subschema node sets in the mapping, T1, defined in the previous step. Let S1 be the 'from' node set (contains the 'from' node leaf node of the subschema transaction) and S2 the 'to' node set (contains the 'to' node of the transac-
tion). All schema transactions with a 'from' node in S1 and a 'to' node in S2 are then associated with the subschema transaction. For the example with VC = g5, g6, A, D, E, F, G, O1 we obtain:
Subschema Transactions = Schema Transaction

Eq. From/Cto
1 SR/A SR/C, SR/A
2 A/E C/E
3 LD/A LD/C
4 A/CO C/CO
5 A/NC T/C, T/NC, C/IN, C/AD, C/DP, C/E, T/E
6 A/T C/T
7 A/DE B/DE

Here generally, this mapping is also many-to-many if the model is more disaggregated than the schema. For example, in the NOS model we obtain:

C/NC, C/IN, C/AD ♦ C/E

The model builder is then given an opportunity to redefine the transactions as required. Note that in the know base the user can obtain exact historic data concerning the transactions in the model via an aggregation process. In particular, the exogenous transaction SR/A is immediately available. On the other hand, in the NOS model the model builder obtains only an aggregate value for three of the variables in his model.

STAGE 5: POST-RECONCILIATION

After the model has been tested and run its results can be integrated with the overall financial plan. The model builder may require projection of the financial statements: (1) for only the subschema, Π, associated with his model; or (2) for any subset of the schema which includes Π.

Use of property 2 facilitates the projection of financial statements for the subschema, Π, associated with a user model. However, if the projection includes a subset of the schema provided by the user and if this involves a disaggregation of a user's schema node, then estimation procedures such as least squares have to be used. As an example, if the projection of the subset below 'A' in Figure 5 is desired, then the user has to determine how the leaf nodes below A are related to the transactions estimated. Here the transaction SR/A is the aggregation of SR/C and SR/AR, and thus the value of SR/A has to be disaggregated in SR/C and SR/AR appropriately. This procedure is widely used in top-down planning where an aggregate financial plan determines transaction values which then have to be disaggregated to the lower levels.

V. MODEL FORMULATION

Given that a user's subschema has been tested by SN for its compatibility with the schema, the planning system will facilitate the input of the model using the Model Data Input and Logic Specification (MDIL). The subschema accepts the interrelationships between transaction types, and between transactions and the current state of the system and/or exogenous variables. It then constructs the following system of equations:

\[ L_A = L_C^{0} + L_C^{1} + L_C^{2} + L_C^{3} \]

where the vector \( A \) corresponds to the exogenous variables at time \( t \), and \( L_C \) is the input of objectives/goals as a function of the states of the system. This is expressed as:

\[ L_C = L_C^{0} + L_C^{1} \]

Using the system equation in (1), it relates the transaction types to the state vector \( T \) (after some rearrangement) as:

\[ B^{\dagger} = A^{\dagger} + B^{\ddagger} + L_C \]

A modeler will be able to use any of the available algorithms to estimate \( T \). In the example discussed by Brouse, a control theory algorithm is used and by inputting the cost matrix \( Q \) that operates on the deviations of \( K \), we have:

\[ T_{1-} = \Sigma \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} \]

All the elements of (3) can be automatically generated by the MDIL and by providing this as input to the control theory algorithm it can evaluate \( T \). For more detailed discussion on MIDL see [10].

The support system, thus, uses the features of the SN and MIDL to develop an integrated planning framework that facilitates linkage of multi-period (or long-term) planning to single-period (or short-term) planning using the set of transaction types, \( T \), and the states of the system, \( B \).

BIBLIOGRAPHY


Edward A. Stob and Wun Nen Tawas
Northwestern University, Graduate School of Management
The Design of a Corporate Planning System Simulator