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

Proceedings of the ACM Winter Simulation Conference, Miami, 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 firms chart of accounts. Viewing the entire accounting system as a schema and the information requirements of each decision-making unit as a subschema, modularity in planning can be achieved. Once the total configuration of the system and the accounting transactions are stored in a data base, a simpler planning language can be developed to simulate the performance of the total system or any subsystem.

I. INTRODUCTION

In recent years there has been a great growth in the use of corporate planning models (10). 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 modules have been constructed to assist top management in their planning function (2) 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 'breadth' 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 methodology. 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 system 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 firms 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/or 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, 2) integration of the plans of the various functional units in the organization allowing both 'top-down' and 'bottom-up' planning, and 3) 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 inputs and outputs to be checked for consistency with the overall financial plan of the organization, 3) information concerning all data elements and models in the planning system.

In section II we briefly describe the planning system 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 consistency checks provided by the system is described in detail and illustrated using a typical financial planning model, Krouse (8). In section VI we outline some of the methods used to help automate the construction of models. In particular the relationship between the linear hierarchical system of the CFM and the Krouse 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 storing and retrieving the accounting-related logic of the corporate planning models (both CFM and FAMS). This module is based on the linear hierarchical model explained above and is outlined in Section III. For a more detailed explanation see (11).

2. Data-base Management System

This is general purpose software based on the EDBS package (17). It may be accessed by the user if desired from any other module (including PMS--although PMS 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 relationships between model data inputs, model statements (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 capabilities are similar to those of a number of similar languages (5), (13).

4. Forecasting Module (Time-Series and Regression)

5. Operations Research Algorithms and Statement Generators (Mathematical Programming and Control Theory Algorithms)

6. Model Data Input and Logic Specification Subsystem

7. Report Generator.

8. Graphics Package

9. Planning System Data Dictionary

In this paper we will concentrate on describing the interface between 1, 3, 4 and 5.

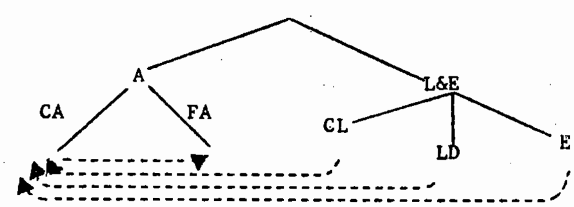
III. THE HIERARCHICAL DATA MODEL

Accounting transactions are used to report the monetary affects of different activities within the firm and in the final analysis financial statements provide the yardstick for measuring its performance. 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 arcs (3). The nodes of the tree correspond to the various accounts. The hierarchical relationships between the nodes in the tree define the accountant's system of classification and aggregation. Figure 1 provides a small example of a chart of corporate balance sheet accounts together with a list of four typical transaction types.

FIGURE 1

ILLUSTRATION OF LINEAR HIERARCHICAL ACCOUNTING SYSTEM



Legend

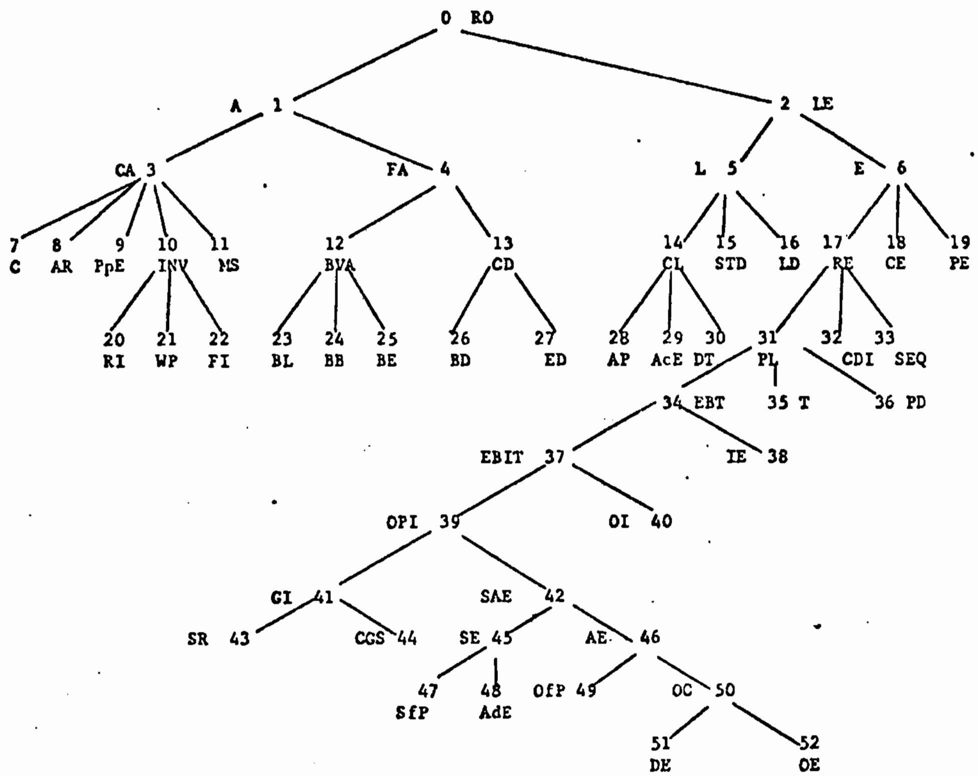
Balance Sheet	Transaction Types
Current Assets (CA)	T1 - investment in firm by owners
Fixed Assets (FA)	T2 - purchase of fixed assets for cash
Current Liabilities (CL)	T3 - long-term borrowing
Long-term Debt (L)	T4 - short-term borrowing
Equity (E)	

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 'states' of the system. Assuming n state variables, their values at time t will be represented by the vector, $b_t \in R^n$.

The various types of accounting transaction are represented by directed arcs connecting leaf-nodes of the tree; the source 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 m arcs the vector of values associated with

FIGURE 2

CORPORATE FINANCIAL SCHEMA



TREE REPRESENTATION OF A CORPORATE FINANCIAL REPORTING STRUCTURE

NO.	Description of Transactions	Node No. (from-to)	Acct.Entries (Cr. /Db.)
1.	Cash Sales	43,7	SR/C
2.	Sales on Account	43,8	SR/AR
3.	Sales Force Expense	7,47	C/SfP
4.	Advertising Expense	7,48	C/AdE
5.	Cost of Goods Sold	22,44	FI/CCS
6.	Collections of Receivables	8,7	AR/C
7.	Office Salary Expense	7,49	C/OfP
8.	Labor Costs Paid In Cash	7,21	C/WP
9.	Material Used in this Period	20,21	RI/WP
10.	Direct Overhead	7,21	C/WP
11.	Pre-paid Expenses Adjusted For	9,52	PpE/OE
12.	Depreciation Expense	26,51	BD/DE
13.	Inventory Increase	21,22	WP/FI
14.	Cash Purchase of Inventory	7,20	C/RI
15.	Purchase of Inventory on Acct.	28,20	AP/RI
16.	Payments of Accounts Payable	7,28	C/AP
17.	Equipment Bought for Cash	7,25	C/BE
18.	Purchase of Equipment on Acct.	28,25	AP/BE
19.	Indirect Expenses	7,52	C/OE
20.	Repayment of Accrued Expense	7,29	C/AcE
21.	Payment of Deferred Taxes	7,30	C/DT
22.	Revenue from Marketable Sec.	40,7	OI/C
23.	Interest Expense	7,38	C/IE
24.	Taxes paid in Cash	7,35	C/T
25.	Taxes Deferred	30,35	DT/T
26.	Other Expenses Deferred	29,52	AcE/OE
27.	Preferred Div. paid in cash	7,36	C/PD
28.	Common Div. paid in cash	7,32	C/CDI
29.	Expenses Pre-Paid (Rent etc.)	7,9	C/PpE
30.	Repayment of Long-term Debt	7,16	C/LD
31.	Repayment of Short-term Debt	7,15	C/STD
32.	Proceeds from Short-term Loan	15,7	STD/C
33.	Purchase of Marketable Securit	7,11	C/MS
34.	Proc. from Sale of Mkt. Sec.	11,7	MS/C
35.	Proc. from Issue of Common Stk.	18,7	CE/C
36.	Proc. from C.E. in Excess of par	18,33	CE/SEQ
37.	Proc. from Issue of Pref. Stk.	19,7	PE/C
38.	Proc. from Issue of L.T. Debt	16,7	LD/C

A TYPICAL LIST OF TRANSACTION TYPES

Acct. (node)	Symbol	Description	Acct. (node)	Symbol	Description
1.	A	Assets	2.	LE	Liab. & Equities
3.	CA	Current Assets	4.	FA	Fixed Assets
5.	L	Liabilities	6.	E	Equities
7.	C	Cash	8.	AR	Accounts Rec.
9.	PpE	Pre-Paid Exp.	10.	INV	Inventory
11.	MS	Marketable Securities	12.	BVA	Bk. Value Of Assets
13.	CD	Cumulative Depreciation	14.	CL	Current Liab.
15.	STD	Short-term Debt	16.	LD	Long-term Debt
17.	RE	Retained Earnings	18.	CE	Common Equity
19.	PE	Preferred Equity	20.	RI	Raw Mat. Inv.
21.	WP	Work-in-Process	22.	FI	Finished Inv.
23.	BL	Bk. Value of Land	24.	BB	Bk. Value of Bldg.
25.	BE	Bk. Value of Equip.	26.	BD	Dep. on Land&Bldg.
27.	ED	Equip. Dep.	28.	AP	Accts. Payable
29.	AcE	Accrued Expense	30.	DT	Deferred Taxes
31.	PL	Profit & Loss	32.	CDI	Common Div.
33.	SEQ	Equity Sale	34.	EBT	Earnings Before Tax
35.	T	Taxes	36.	PD	Preferred Div.
37.	EBIT	Ear. Before Int. & Tax	38.	IE	Interest Expense
39.	OPI	Operating Income	40.	OI	Other Income
41.	GI	Gross Income	42.	SAE	Sales & Adm. Exp.
43.	SR	Sales Revenue	44.	CGS	Cost of Goods Sold
45.	SE	Sales Expense	46.	AE	Administrative Exp.
47.	SfP	Sales Force Expense	48.	AdE	Advertising Exp.
49.	OfP	Office Payroll	50.	OC	Other costs
51.	DE	Depreciation Exp.	52.	OE	Other Expenses

(Legend)

the arcs at time t will be denoted by the 'aggregate transactions vector', $\tau_t \in R^m$. The states of the system (values associated with the nodes) are changed in each period by the transactions. The effect on the leaf nodes can be summarized by the conventional node-arc 'incidence matrix'. If one also wants to represent the effect on higher order nodes in the tree structure, 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 resulting matrix will be referred to as the (n X n) 'systems matrix', S. Corresponding to Figure 1 we have:

$$
S = \begin{array}{c} \\ A \\ L\&E \\ CA \\ FA \\ CL \\ LD \\ E \end{array}
\begin{array}{c} \begin{array}{cccc} T_1 & T_2 & T_3 & T_4 \end{array} \\
\begin{bmatrix} +1 & 0 & +1 & +1 \\ -1 & 0 & -1 & -1 \\ +1 & -1 & +1 & +1 \\ 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \end{array}
$$

Since some of the states are flow items (income statement accounts) the corresponding balances must be zeroed each period. Let E be an (n X n) matrix with $E_{ii} = 1$ if the ith node is a balance sheet item (stock variable) and zero otherwise. Then b_{t+1} can be obtained from b_t by the linear 'systems equation':

$$b_{t+1} = Eb_t + S\tau_t \qquad (1)$$

This completes the description of the graphical and algebraic representation of the accounting system employed by the SM. 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 rearrangement 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 sub-schema and the official schema. A partly automated procedure for doing this is described in Section V. The system trees are stored internally using a 'multi-attribute' tree structure (11), (15). The systems 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 (1) is that it contains all relevant accounting identities. If the initial state of the system is known and the aggregate transactions vector τ_t can be estimated, then the projected financial statements can be immediately computed using (1). As another application, in simulation models the modeler 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 PLAN

In this section we describe the design features of the planning system that allows the integration of diverse models (FAM's) with the central financial model (CFM). Referring to the systems equation (1) our general approach is to use the transaction vector, τ, as the means of communication: the inputs and outputs of the model are expressed in terms of the aggregate transaction vector, τ^s, associated with the model subschema and the vector τ^s is then mapped to the aggregate transaction vector τ 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 affect on the state of the system is described by the accounting identities stored in the system tree. As far as 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 our 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 statements 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. time-series, regression, simulation, subjective).

2. The corporate schema (T,B) where T is the set of transaction types (arcs) and B is the set of state variables (nodes in the tree).

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

3. Data base for corporate and economic data.

4. Retrieval and display facilities.

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

1. 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 subschema (T^S, B^S) where T^S is the set of transaction types and B^S the set of state variables which the FAM 'affects'.

3. FAM database

4. Retrieval and display facilities.

5. Mapping functions: $(T^S, B^S) \rightarrow V^S$ and $V^S \rightarrow (T^S, B^S)$ where V^S 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 Holt, Muth, Modigliani and Simon (HMMS) aggregate workforce and inventory smoothing model (7). The model variable set, V^S is:

P_t = aggregate production rate at time t

w_t = workforce level at time t

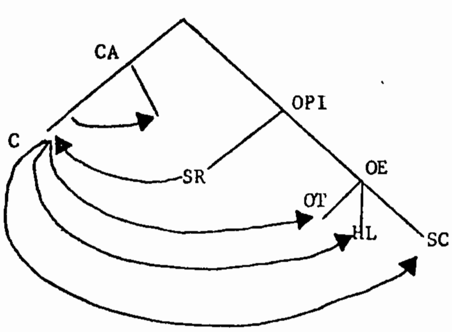
I_t = net inventory-on-hand at time t

S_t = sales revenue at time t

The subschema and mapping are shown in Figure 3:

FIGURE 3

HMMS MODEL
(only those nodes affected by transactions in the model are shown)



Transaction Type	
C/HL	Hiring and layoff costs
C/SC	Inventory back order & set up costs
C/OT	Expected overtime costs
C/INV	Regulat payroll costs
SR/C	Sales Revenue (exogenous)

Mapping rule, $V^S \rightarrow T^S$	
C/HL -	$c_2(w_t - w_{t-1} - c_{11})^2 + c_{13}$
C/SC -	$c_7(I_t - (c_8 - c_9 S_t))^2$
C/OT -	$c_3(P_t - c_4 w_t)^2 + c_5 P_t - c_6 w_t + c_{12} P_t w_t$
SR/C -	$c_1 w_t + c_{13}$

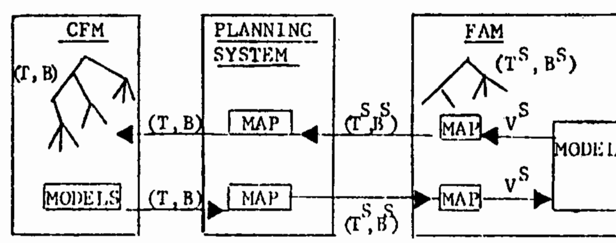
The coefficients, c_i, are parameters to be estimated. Referring to Figure 3 it can be seen that, relative to the schema, some nodes in the subschema are aggregation points and some are disaggregation points representing a finer classification. Similarly, some transactions are aggregated and some disaggregated while the transaction C/SR occurs in both the schema and subschema.

The mappings $(T,B) \rightarrow (T^S,B^S)$ and $(T^S,B^S) \rightarrow (T,B)$ will require further rules to be stored by the planning system and these will depend on the accounting system chosen by the firm. One purpose of linking the FAM and CFM in this way is to assist the model-builder in estimating the parameters and exogenous variables required by the FAM. In the HMMS case past values of the cost transactions (at least C/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, subschema and FAM model is depicted in Figure 4.

FIGURE 4

RELATIONSHIP BETWEEN CFM AND A FAM



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 t.e CFM. Budget levels or goals are then sent (via T) 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, those at the FAM level may be for daily, weekly or monthly periods and may take into account the more

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

detailed constraints and requirements of the lower organizational entity involved. Since the top level plans should be constrained by the feasibility requirements of the FAM's certain parameters, bounds and constraint conditions should be maintained and stored in the FAM for use by the higher level planning system when required (not shown in the Figure).

In bottom-up planning on the other hand, the results of the FAM models are made available (via T^S) to the CFM which consolidates them into the overall financial projections for the company. Obviously 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 FAM's.

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

1. Input of Model Subschema (T^S, B^S).

2. Prereconciliation Stage (Mapping $(T,B) \rightarrow (T^S, B^S)$)

   The objective of the systems manager (SM) at this stage is to make the historical and projected time series data stored in the CFM compatible with the variables of the user's model. The model builder can then use the previous historic 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 planning system attempts to make the task of model specification simpler by automatically generating the matrices E and S in equation (1) (which together contain all accounting identities) and providing other aids such as an integer, linear and goal programming tableau generator which works directly from an algebraic ('sigma notation') statement 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 $(T^S, B^S) \rightarrow (T,B)$)

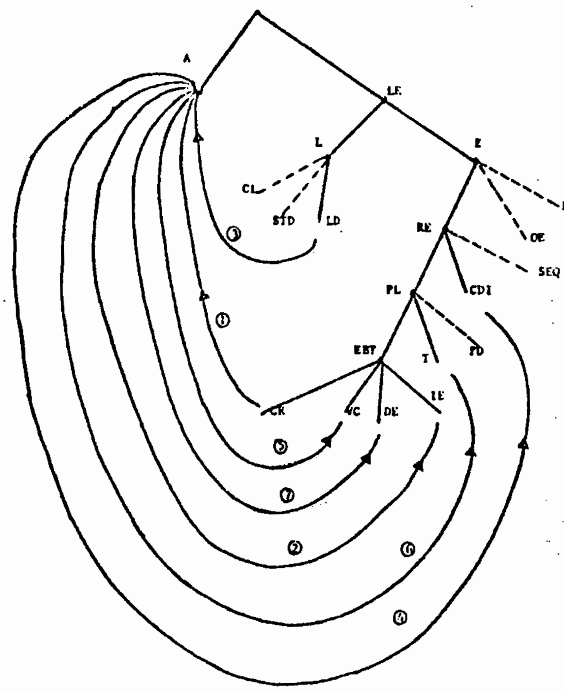
   The objective of this stage is to translate the model results into a form compatible with the data stored in the CFM. This allows the user to examine corporate financial statements revised according to the results of his model. Since the model will be concerned only with a subset of the CFM transaction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

V. PRE- AND POST RECONCILIATION

In this section we outline the method used to establish the relationship between (T,B) and (T^S, B^S) and illustrate it using the financial planning model given in Krouse (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 (6), (9) (as well as the HMMS aggregate production model given above), this is a linear quadratic Gaussian (LQG) stochastic control model with the familiar 'linear decision rule'. The model subschema, variables and transactions are shown in Figure 5.

FIGURE 5

SUBSCHEMA FOR KROUSE MODEL



Input By User			Krouse Model	
No.	Affected Nodes Cr/Db)	Description	Nature of these Variable	Variable Definition
1	SR/A	Sales Revenue	exogenous	R
2	A/IE	Interest Expense	endogenous	$k_5 \cdot D$
3	D/A	Debt Acquired	decision variable	b
4	A/CDI	Dividends Paid	decision variable	d
5	A/VC	Variable Expense	decision variable	a
6	A/T	Tax Expense	endogenous	$(1-k_6)P$
7	A/DE	Depreciation Expense	endogenous	$k_4 \cdot A$

The objective of the Krouse model is to determine optimal values for the debt acquired, d_t, the dividends paid, b_t, and the expenses incurred, a_t, subject to achieving a set of financial goals--earnings growth and a number of balance sheet ratios (ratios of elements in B^S).

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 or both of 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 depends on the semantics of the models involved. The role of the system manager is to provide a first approximation of the relationship between the CFM and FAM. The FAM model builder, perhaps working jointly with the CFM 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 subclassification nodes where necessary (e.g. the nodes OT, HL and SC in the HMMS model).

Rule 2: If the model subschema contains nodenames which also appear in the schema then the associated nodes are assumed by the SM 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 A and LE. Subject to the above rules the model builder can redefine state variables and their hierarchical relationships.

The subschema tree generated and stored by the SM (the 'internal' subschema) may differ from that input by the user. It will have to satisfy the following properties:

Property 1: Each of the transactions defined in the user subschema will be associated with two leaf nodes. All the schema nodes on the paths from this set of subschema leaf nodes to the root of the tree are included. If some of the leaf notes and/or aggregate points are not known to the schema, the first recognizable node in this path is taken as the starting point for this expansion. The starting point is always ensured 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 set of new nodes not known to the schema and/or revises the structure of the existing ones.

Let $L_S(X)$ represent the set of leaf nodes of the sub-trees for which the nodes in X are parents and let $P_S(x)$ define the nodes in the path between $x \in X$ and the root node of the subschema. The SM can generate $L_S(x)$ and $P_S(x)$ for each aggregation point of the subschema input by the user including the root node. Let us also define $I_S(x)$ as a measure function, which is 0 if $L_S(x) \subset L(x)$ and is n if there are n elements in $L_S(x)$ that are not in $L(x)$. Here $L(x)$ is the set of leaf nodes in the schema for which x is the parent. If x, itself, is not known to the schema, then $I(x)$ is set to -1 to avoid its consideration.

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

(i) $x \in X$ where X is the set of all aggregation points in the subschema.

(ii) $I(x) = k > 0$.

(iii) $\{\nexists\, y \in X | P_S(x) \subset P_S(y)$ and $I(y) = k\}$

(iv) $\{\nexists\, y \in X | P_S(y) \subset P_S(x)$ and $I(y) > k\}$

Determination of LAP's will partition the tree such that the tree above a LAP is similar to the one 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' above 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 HMMS model does not satisfy property 1 because, for example, schema node A is not included; it does not satisfy property 2 because schema nodes AR, PPE and MS are omitted from the subtree for CA. Note that the LAP here is OE. If 'OE' was redefined by the user in his subschema as, say, 'IC', then the LAP would have been 'OPI'. The subschema for the 'Krouse' model (Figure 5) satisfies property 1, but not property 2. The LAP here is EBT.

For future reference letus define B^S, as the set of subschema nodes input by the user and revised by property 1 with the revision occuring above the LAP's. Let us define B_2^S to include nodes added to satisfy property 2. B^S is then $B_1^S \cup B_2^S$.

STAGE 1: INPUT OF MODEL SUB-SCHEMA (T^S, B^S)

(a) Input of State Variables, B^S

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. Sample interactions between the model builder and the system manager are shown in (T) and (L). The nodes used directly by the Krouse model are

$B_1^S = \{A, LE, L, E, LD, RE, PL, CDI, EBT, T, SR, VC, DE, IE\}$

Note that all these nodes also appear in the schema with the exception of VC. After the subtree definition has been completed by the user, the system manager checks for adherence to rules 2 and 3, and develops the internal subschema that satisfies properties 1 and 2. The nodes added to satisfy property 2 are:

$$B_2^S = \{CL, STD, CE, PE, SEQ, PD\} .$$

The nodes in set B_2^S (and their associated historic or projected schema transactions) are required during the post-reconciliation stage so that their parent nodes are properly evaluated. During the pre-reconciliation stage on the other hand the associated subtrees and transactions must be subtracted from the schema.

(b) Input of Transactions

The user is then asked to input the transaction types (defined by the associated leaf nodes in from-to or credit/debit order) and their descriptions for his model. In the Krouse example there are seven transactions as shown in the table in Figure 5. Let this transaction set be called T^S. Note that all transactions from the asset side of the balance sheet have been aggregated under the total assets node, A. After the input of the transactions the planning system checks that rule (1) is satisfied.

STAGE 2: PRE-RECONCILIATION

For comparative evaluation of the subschema with the proforma schema one needs to establish the relationships between the nodes and transactions in both. The system manager takes the following steps:

(a) Identify, T_0, the set of all transactions in the schema that affect two leaf nodes in the subschema. This eliminates all transactions which have a zero affect on leaf nodes in the subschema. (These are the schema transactions that form a leaf in the schema and sub-tree associated with a subschema leaf node, e.g. the collection transaction AR/C).

(b) Develop a Mapping, M, between leaf nodes in the Model Subschema B_1^S and the Schema, B. Now $L(B) - L(B_2^S)$ represents the leaf nodes in the schema which are of interest and $L(B_1^S)$ represents the leaf nodes in the model subschema. To assist the user in the identification process the system manager: (i) scans the list $L(B_1^S)$ and identifies nodes defined in the schema. If a non-leaf node is identified it is mapped to its children in L(B); if a leaf-node is identified it is associated directly with the node of the same name in L(B). A list of the identified associations and nodes with no associations is then printed. For the Krouse example we have:

Subschema Leaf Nodes	≡	Schema Leaf Nodes
LD		LD
SR		SR
DE		DE
IE		IE
T		T
CDI		CDI
A		C,AR,PPE,RI,WI,FI,MS,BL BB,BE,BD,ED.

Nodes with no associations: CGS,SFP,ADE,OFP,OE,OI.

If the planning system can not 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 ≡ CGS,SFP,ADE,OFP,OE,OI
2. VC ≡ CGS,SFP,ADE,OFP,OE
   SR ≡ SR,OI
3. VC ≡ CGS,SFP,ADE,OFP,OE
   OI ≡ $\emptyset$

In the first specification 'variable costs' is defined as an aggregation of the various cost accounts and Other Income, OI. In the second OI is combined with sales revenue to produce a redefinition of the subschema node SR as 'total income'. In the third specification OI is not relevant to the model. This node is then included in the set B_2^S for use in post-reconciliation. During the post-reconciliation phase it will be recognized by the system as a child of EBT.

More generally, if the model subschema involves disaggregation as well as aggregation as many-to-many mapping is obtained. Thus in the HMMS model we have: OT, HL, SC ≡ OE.

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 $L(B_2^S)$ and the sets defined by the above mapping.

(c) Develop the association between the Model Transaction Set, T^S, and the Schema Transaction Set, T^0.

The system manager scans T^S. Each subschema transaction identifies a pair of subschema node sets in the mapping, M, defined in the previous step. Let S_1 be the 'from' node set (contains the 'from' leaf node of the subschema transaction) and S_2 the 'to' node set (contains the 'to' node of the transaction). All schema transactions with a 'from' node in $M(S_1)$ and a 'to' node in $M(S_2)$ are then associated with the subschema transaction. For the Krouse example with VC ≡ CGS,SFP,ADE,OFP,OE we obtain:

Edward A. Stohr
Northwestern University, Graduate School of Management

Subschema Transactions ≡ Schema Transactions

No.	From/To	
1	SR/A	SR/C, SR/AR
2	A/IE	C/IE
3	LD/A	LD/C
4	A/CDI	C/CDI
5	A/VC	FI/CGS,C/SFP,C/ADE, C/OFP,PPE/OE
6	A/T	C/T
7	A/DE	BD/DE

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

C/OT,C/HL,C/SC ≡ C/OE

The model builder is then given an opportunity to redefine the transactions as required. Note that in the Krouse case 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 HMMS 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, B^S, associated with his model; or (2) for any subset of the schema which includes B^S.

Use of property 2 facilitates the projection of financial statements for the subschema, B^S, associated with a users 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 SM 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 to 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 nodes based on their historical relationship to the nodes/transactions of the schema used by the corporate planner.

V. MODEL FORMULATION

Given that a user's subschema has been tested by SM for its comparability with the schema, the planning system will facilitate the input of the model using the Model Data Input and Logic Specification Subsystem (MILS). This subsystem 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:

$$L_1\tau^t = L_2b^t + L_3z^t \qquad (2)$$

where the vector z^t corresponds to the exogenous variables at time t. The MILS also facilitates input of objectives/goals as function of the states of the system. This is expressed as

$$K_1b^{t+1} = K_3 \qquad (3)$$

Using the system equation in (1), it relates the transaction types to the state vector b^{t+1} (after some rearrangement) as:

$$b^{t+1} = Ab^t + B\tau^t + Cz^t \qquad (4)$$

A modeler will be able to use any of the available algorithms to estimate τ^t. In the example discussed by Krouse, a control theory algorithm is used and by inputting the cost matrix Q that operates on the deviations of K_3, we have

$$\text{Min:} \quad \sum_{t=1}^{T-1} b^{t+1}K_1'QK_1b^{t+1} \qquad (5)$$

$$\text{s.t.:} \quad b^{t+1} = Ab^t + S\tau^t + Cz^t, \; t = 1,2,\ldots,T-1.$$

All the elements of (5) can be automatically generated by the MILS and by providing this as input to the control theory algorithm it can evaluate τ^t, t = 1,...,T. For more detailed discussion on MILS see (16).

The support system, thus, uses the features of the SM and MILS 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

1. Blin, J.M., E.A. Stohr and Mohan Tanniru. Proceedings IEEE COMPSAC Conference, Chicago, 1977.
2. Buneman, Peter O., Howard L. Morgan and Michael D. Zisman. "Display Facilities for DSS Support: The Daisy Approach," in DATABASE, Vol. 8, No. 1, Winter, 1977.
3. Butterworth, John. "The Accounting System as an Information Function," Journal of Accounting Research (Spring, 1972), pp. 1-27.
4. CALL/370: PROPHIT II Reference Manual, The Service Bureau Company, February, 1977.
5. FPS User's Guide, Common Library 701, Scientific Time Sharing Corporation, December, 1973.

6. Hamilton, William F. and Michael A. Moses. "An Optimization Model for Corporate Financial Planning," Operations Research, Vol. 21, No. 3, pp. 677-792, May-June 1973.
7. Holt, Charles C., Franco Modigliani, John Muth and Herbert Simon. Planning, Production, Inventories and Workforce, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1960.
8. Krouse, Clement G. "A Model for Aggregate Financial Planning," Management Science, Vol. 18, No. 10, pp. B555-B566, June 1972.
9. Lev, Baruch and Dov Peckelman. "A Multiperiod Adjustment Model for the Firm's Capital Structure," The Journal of Finance, Vol. XXX, No. 1, pp. 75-91, March 1975.
10. Naylor, Thomas H. and Horst Schanland. "A Survey of Users of Corporate Planning Models," Management Science, Vol. 22, No. 9, May 1976.
11. Ormancioglu, Levent. "Process Managerial System," Ph.D. Dissertation, Northwestern University, unpublished.
12. Scott Morton, Michael S. "Some Perspectives on Computerized Management Decision-Making Systems," Proceedings IEEE COMPSAC Conference, Chicago 1977.
13. SIMPLAN User's Manual, Social Systems, Inc., Chapel Hill, N.C., 1976.
14. Stohr, Edward A. "A Mathematical Programming Tableau Generator" (working paper).
15. Subas, S.K.C. and R.L. Kashyap. "Data Base Organization--An Access and Storage Method," Ph.D. Dissertation, School of Electrical Engineering, Purdue University (September 1975).
16. Tanniru, Mohan R. "A Decision Support System for Planning," Ph.D. Dissertation, Northwestern University, 1978.
17. Tsichritzis, C. Dionysios and Frederick H. Lochovsky. Data Base Management Systems, 111 Fifth Avenue, New York, NY, Academic Press, Inc., 1977.