WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

INTERACTIVE BUDGETING MODELS: AN EXAMPLE

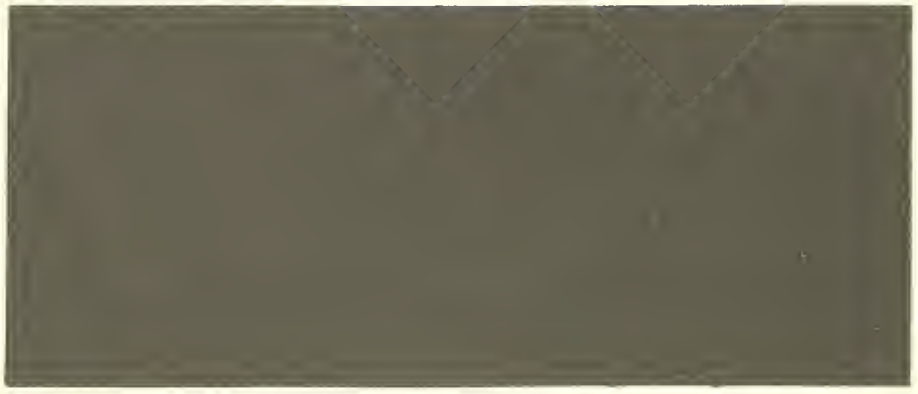
Preliminary Draft

345-68

David N. Ness

Sloan School of Management
Massachusetts Institute of Technology

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139



INTERACTIVE BUDGETING MODELS: AN EXAMPLE

Preliminary Draft

345-68

David N. Ness

Sloan School of Management
Massachusetts Institute of Technology

Acknowledgements

The program discussed in this paper was originally written (in a batch processing version) by Professor J. C. Emery of the Wharton School and Professor M. M. Jones of M.I.T. The present program differs very substantially from the earlier versions. The author profited from the comments of Professors D. C. Carroll, G. A. Gorry and Z. S. Zannetos as well as Professor Emery.

Work reported herein was supported by Project MAC, an M.I.T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(01).

INTERACTIVE BUDGETING MODELS: AN EXAMPLE

The purpose of this paper is to discuss an interactive budgeting model which has been developed on the Compatible Time Sharing System of M.I.T.'s Project MAC. While this model is not general, it does demonstrate many of the features and facilities which can be made available in an interactive environment. The paper is introduced by a transcript of a console session. The program which performs the computations is then discussed.

Introduction

The model discussed in this paper is derived from a solution to a problem in Shillinglaw's cost accounting book.[1] This example demonstrates that a computerized solution to such a problem accomplishes much more than just giving an answer to the questions posed by the problem. A computer program which solves this problem is in fact a model of the environment and thus it can be used for much more than just the conventional budgeting purposes. Such a model can be used to evaluate the effects of changes in the values of important parameters. For example, it is possible to change the advertising expense with the phrase

"MULTIPLY THE VALUES OF ADV BY 1.10".

[1] Shillinglaw, Gordon, Cost Accounting, 1961 Edition, Problem 15-25, pages 490-492.

In this way it provides the manager with a context with which he can experiment, an opportunity which is not available in the real environment. It should also be made clear that interactive budgeting does not only accomplish these ends. It also allows the process of budgeting to become sequentially adaptive. That is to say, it becomes possible to adapt the budget to the realities of the world as they become known. This means that the budget can be a much more realistic and intelligent basis for the establishment of reasonable standards.

A Session with the Model

The following section of the paper is a direct transcript of a session with the budgeting model. It is given here simply to demonstrate how the model actually works. The responses of the computer are actually typed out sequentially. Therefore, it must be remembered that given the present system it is not possible for the user to scan a report in the same way as the reader does because the report is actually being typed out character by character. This, of course, influences the design of some parts of the process.

In the transcript below, which was produced on an IBM Model 1050 terminal, the user's commands are all in lower case. The computer's responses are in upper case. The character "#" erases the previous character on the line and the character "@" erases the entire line.

r budget
W 1912.7
PLEASE INPUT REQUESTS, OR DO YOU WANT HELP

i would like som help please.
CASH BUDGET IS AVAILABLE
SO IS INVENTORY RECONCIATION
YOU MIGHT ALSO ASK FOR A PROFIT BUDGET
SINCE WE ABBREVIATE SOME IMPORTANT PARAMETERS
   YOU CAN REQUEST A LIST OF ABBREVIATIONS
YOU CAN ASK FOR A STATEMENT OF THE
   COMPUTATION OF OVERHEAD
YOU MAY REQUEST A PRINTOUT OF VARIABLES, OR
   YOU CAN REQUEST TO CHANGE THEM
AFTER CHANGING A PARAMETER, HOWEVER, YOU MUST
   REQUEST A COMPUTATION TO GET THE NEW
   DATA INCORPORATED INTO THE MODEL
AT ANY POINT YOU CAN ASK FOR A LIST OF THE ALTERNATIVES
   WHICH THE MODEL UNDERSTANDS AT THAT POINT.
YOU CAN ALSO CHANGE ANY DATUM THAT IS NOT COMPUTED
   BY THE MODEL FROM SOME OTHER DATA.

compute and print a profit budget

MONTHLY PROFIT BUDGET

	JAN	FEB	MAR	APR	MAY	JUN	TOTAL
SALES REVENUE	201320	282490	415150	355830	294110	281220	1830120
COST OF GOODS SOLD	157581	221183	324804	278519	230296	220275	1432658
GROSS MARGIN	43739	61307	90346	77311	63814	60945	397462
SELLING AND ADM EXP	42051	44781	53461	43002	43549	42675	274519
OPERATING INCOME	1638	16526	36885	20309	20266	18270	122943
INTEREST EXPENSE	167	333	333	300	00	00	1133
NET INC BEFORE TAX	1521	16192	36552	29009	20266	18270	121810
PROVISION FOR TAX	456	7962	18549	14626	10080	9042	60715
NET INCOME	1065	8231	18003	14382	10186	9228	61095

MONTHLY CASH BUDGET

	JAN	FEB	MAR	APR	MAY	JUN	TOTAL
TOTAL COLLECTIONS	264546	242407	304041	367997	345026	299029	1823046
DISBURSEMENTS							
DIRECT MATERIAL	86929	77662	70993	66794	68701	61132	432211
FACTORY PAYROLLS	137407	157420	143725	134692	136139	125122	834505
MACHINE ROYALTIES	00	11288	10085	9240	8722	8945	48279
FUEL, POWER, LIGHT	45000	2437	2225	1663	1093	1027	53445
OTHER MFG OVERHEAD	00	3688	3663	3643	3630	3636	18260
FACTORY SUPPLIES	2096	1880	1708	1591	1648	1420	10343
SALES COMISSIONS	15000	10066	14124	20757	17791	14705	92445
S AND A SALARIES	9882	12352	12352	12352	12352	12352	71042
ADVERTISING	25000	16520	14380	15100	13200	12450	96650
OTHER S AND A EXP	00	1100	1100	1100	1100	1100	5500
CAPITAL ADDITIONS	5000	5000	5000	5000	5000	5000	30000
DIVIDENDS	00	00	10000	00	00	10000	20000
TOTAL DISBURSEMENTS	326313	299412	289356	271932	269377	256890	1713279
NET RECEIPTS	-61767	-57005	14685	96065	75650	42139	109767
CASH BALANCE BOM	100000	88067	80728	95080	180845	166495	
AMOUNT BORROWED	50000	50000	00	00	00	00	100000
DEBT REPAYMENT	-00	-00	-00	-10000	-90000	-00	-100000
INTEREST PAYMENT	-167	-333	-333	-300	-00	-00	-1133
ADJUSTED RECEIPTS	314546	292407	304041	367997	345026	299029	1923046
ADJ DISPURSEMENTS	326479	299745	289689	282232	359377	256890	1814412
ADJ NET RECEIPTS	-11933	-7338	14352	85765	-14350	42139	108634
CASH BALANCE EOM	88067	80728	95080	180845	166495	208634	
ENDING DEBT	50000	100000	100000	90000	00	00	00

print the overhead allocation

BUDGETED OVERHEAD COSTS

	JAN	FEB	MAR	APR	MAY	JUN	TOTAL
FACTORY OVERHEAD							
SALARIES AND WAGES	10379	9740	9604	9318	9485	8975	57502
SUPPLIES	2096	1880	1708	1591	1648	1420	10343
FUEL, POWER, LIGHT	2437	2225	1663	1093	1027	791	9236
DEPRECIATION	2470	2470	2470	2470	2470	2470	14820
OTHER FACTORY EXP	3688	3663	3643	3630	3636	3610	21869
TOTAL FAC OVERHEAD	21069	19978	19088	18102	18267	17265	113770
OH ABS AT .286/UN	23052	20678	18790	17503	18132	15616	113771
UNDER(OVER)ABSORBED	-1982	-700	298	599	134	1650	-1
SELLING AND ADMINISTRATIVE EXPENSES							
SALES COMISSIONS	10086	14124	20757	17791	14705	14091	91506
S AND A SALARIES	12352	12352	12352	12352	12352	12352	74112
ADVERTISING	16520	14380	15100	13200	12450	12350	84000
BAD DEBT	2013	2825	4152	3558	2941	2812	18301
OTHER S AND A EXP	1100	1100	1100	1100	1100	1100	6600
TOTAL S AND A EXP	42051	44781	53461	48002	43549	42675	274519

DIRECT MATERIAL	86329	77662	70993	66794	68701	61132	432211
DIRECT LABOR	161379	144096	131593	123747	127423	113200	801438
MACHINE ROYALTIES	11268	10085	9240	8722	8945	8059	56338
OVERHEAD ABSORBED	23052	20678	18790	17503	18132	15616	113771
GOODS MANUFACTURED	282647	252520	230616	216766	223202	198006	1403758

tell me the abbreviations and values of the variable parameters
PSC -- PERCENTAGE SALES PAID IN COMISSIONS
PFM -- PERCENTAGE A/R RECIEVED IN 1ST MONTH
PSM -- PERCENTAGE A/R RECEIVED IN 2ND MONTH
PTM -- PERCENTAGE A/R RECEIVED IN 3RD MONTH
(THE REMAINDER IS BAD DEBT)
PFP -- PERCENTAGE OF FACTORY WAGES PAID IN CURRENT MONTH
ROH -- OVERHEAD PER UNIT
BLIM-- MINIMUM BANK BALANCE
DTD -- MULTIPLE FOR INCREMENTS TO DEBT
VFSW-- VARIABLE FACTORY SALARIES AND WAGES (/UNIT)
VFS -- VARIABLE FACTORY SUPPLIES COST (/UNIT)
SFAC-- MULTIPLIER TO BASIC SALES DATA
PFAC-- MULTIPLIER TO BASIC PRODUCTION DATA

THE VALUES OF THE VARIABLE PARAMETERS ARE AS FOLLOWS.
PSC = .05
PFM = .30
PSM = .50
PTM = .19
PFP = .80
ROH = .286/UNIT
BLIM = 80000
DTD = 10000
VFSW = .052/UNIT
VFS = .026/UNIT
ICOR = 1

ok, change vfsw to 6.50 and tell me keywords after tax *end*
TAX -- TAXES
PTIN -- PRETAX INCOME
ATIN -- AFTER TAX INCOME
output the values of atin and output total atin
VALUES OF ATIN ARE 1065 8231 18003 14392 10196 9226

```
save curr1 for later use. now compute and output the values of atin and output total atin
VALUES OF ATIN  ARE          995      8119     17875     14238     10010      9116
alternatives
THERE ARE 26 ALTERNATIVES CALLED MAIN COMMANDS.
TO SEE THEM TYPE 'YES'.
no
output total atin
VALUE OF ATIN  IN TOTAL IS      60353

compare curr1 and tell me alternatives
THERE ARE  4 ALTERNATIVES CALLED KINDS OF COMPARISON.
TO SEE THEM TYPE 'YES'.



yes
ALTERNATIVES ARE
ABSOLU RATIO  DIFFER D-RATI
use the absolute ratio and print the maximum 5 values
UOA     TOTAL            148486.8281
UOA     MAY                 212.3335
UOA     MAR                  99.7042
UOA     APR                  46.7570
UOA     FEB                  45.2588
                              \
keyword uoa end### *end*
UOA -- UNABSORBED OVERHEAD

print the overload allocation
```

BUDGETED OVERHEAD COSTS

	JAN	FEB	MAR	APR	MAY	JUN	TOTAL
FACTORY OVERHEAD							
SALARIES AND WAGES	46488	42130	39038	36736	37888	33436	235716
SUPPLIES	2096	1880	1708	1591	1648	1420	10343
FUEL, POWER, LIGHT	2437	2225	1663	1093	1027	791	9236
DEPRECIATION	2470	2470	2470	2470	2470	2470	14820
OTHER FACTORY EXP	3688	3663	3643	3630	3636	3610	21869
TOTAL FAC OVERHEAD	57178	52368	48522	45520	46670	41726	291984
OH ABS AT .286/UN	23052	20678	18790	17503	18132	15616	113771
UNDER(OVER)ABSORBED	34126	31690	29732	28017	28538	26110	178213

SELLING AN alNI. 0

ok, spread and compute then print the overhead allocation

BUDGETED OVERHEAD COSTS

	JAN	FEB	MAR	APR	MAY	JUN	TOTAL
FACTORY OVERHEAD							
SALARIES AND WAGES	46488	42130	39038	36736	37888	33436	235716
SUPPLIES	2096	1880	1708	1591	1648	1420	10343
FUEL, POWER, LIGHT	2437	2225	1663	1093	1027	791	9236
DEPRECIATION	2470	2470	2470	2470	2470	2470	14820
OTHER FACTORY EXP	3688	3663	3643	3630	3636	3610	21869
TOTAL FAC OVERHEAD	57178	52368	48522	45520	46670	41726	291984
OH ABS AT .734/UN	59160	53068	48224	44921	46535	40076	291984

UNDER(OVER)ASSORTED	-1982	-700	298	599	135	1650	0

SELLING AND ADn-INT. 0

now output the values of atin and output total of atin
VALUES OF ATIN ARE -18649 -12310 -4707 -6997 -9624 -10097
VALUE OF ATIN IN TOTAL IS -62384

restore curr1 and multiply 1.25 times the values of the sales of product a
then compute and print an inventory reconcilation
INVENTORY FOR PRODUCT A GOES NEGATIVE IN MAY
INVENTORY FOR PRODUCT A GOES NEGATIVE IN JUN

RECONCILIATION FOR PRODUCT A

	JAN	FEB	MAR	APR	MAY	JUN	TOTAL
INVENTORY B.O.M.	5200	6725	5800	2675	275	-1700	
PRODUCTION	6400	6200	6000	5600	5400	4700	
SALES	4875	7125	9125	8000	7375	6625	
INVENTORY E.O.M.	6725	5800	2675	275	-1700	-3625	

INT. C
restore curr1 then and print the inventory te##reconcilation.

RECONCILIATION FOR PRODUCT A

	JAN	FEB	MAR	APR	MAY	JUN	TOTAL
INVENTORY B.O.M.	5200	7700	8200	6900	6100	5600	
PRODUCTION	6400	6200	6000	5600	5400	4700	
SALES	3900	5700	7300	6400	5900	5300	
INVENTORY E.O.M.	7700	8200	6900	6100	5600	5000	

INT. 0


```
give me keyword advertising *end*
ADV -- ADVERTISING

output the values of advert######adv ad#nd output the mean s l and dev of adv
VALUES OF ADV   ARE       16520     14380     15100     13200     12450     12350
MEAN OF ADV   IS      14000.00
SD OF ADV   IS      1498.32
DEVIATIONS OF ADV   ARE       1.682      .254      .734     -.534    -1.034    -1.101

change @multiply@  add 10000. to the values of adv , then output the values of adv
VALUES OF ADV   ARE       26520     24380     25100     23200     22450     22350
```

```
to...multiply 1.03 times the sa...es of product a
    (now
        multiply 1.03 times the values of the sales of product a
compute and output the values of atin
VALUES OF ATIN  ARE      -28551    -22166    -14590    -16902    -19576    -20054
output total atin
VALUE OF ATIN  IN TOTAL IS    -121849

change vfsw back to 0.052 then compute and output the values of atin
VALUES OF ATIN  ARE       -8328      3520     13322      9669      5463      4526

output total atin over the period
VALUE OF ATIN  IN TOTAL IS      28173

compare curr1 using ratio@

compare curr1 using ratio and then print the maximum 5 and the minimum 5
ADV     JUN                  1.8097
ADV     MAY                  1.8032
ADV     APR                  1.7576
ADV     TOTAL                1.7143
ADV     FEB                  1.6954
ATIN    JAN                 -7.8215
BTIN    JAN                 -5.4750
OPIN    JAN                 -4.8356
UOA     TOTAL                -.0016
PAYN    APR                  0.

keyword btin a:opin *end*
OPIN -- OPERATING INCOME
BTIN -- BEFORE TAX INCOME

thats all thanks, your work is complete


WHEW, THANK GOD IM DONE.  I WAS GETTING TIRED.
R 22.333+17.200
```

The Structure of the Program

The original version of the program accepted a complete specification of the values of the parameters as its input and produced all of the appropriate management reports as its output. Among the input parameters were such things as initial values of inventories, sales and production figures, materials and labor costs and interest cost. The reports produced were profit budgets, cash budgets, allocation of overhead and reconcilliations of inventory. This original version of the program was, in fact, written for a batch processing environment and worked quite effectively in that capacity. The realities of a batch processing environment, however, produce substantial delays between successive runs of the model. This makes it impossible for the manager to interact with the model. It was, therefore, decided to make the model available on-line using the facilities of Project MAC. When such a program is transferred bodily from a batch-processing environment to an interactive one, several things become obvious. First, it is not desirable to require that all parameter values be specified every time a parameter is changed. Similarly, while the person who will eventually read the reports is not required to watch them as they are printed on the printer in the batch-processing mode, he is required to watch them be typed out on the console if, in fact, he decides to interact with that console.[2] Thus it becomes important to be able to selectively specify inputs and also the outputs. In the first round of implementation, although all of the

[2]As display consoles become available this will speed up this response very significantly.

specified reports were printed out, one could decide to look at the profit budget, the cash budget, etc., or any combinations of these things. In the current version of the program, it is possible to limit output to individual items or arrays of items, as well as whole reports.

After a little experimentation with such an interactive model, it became clear that it was desirable to be able to use the mathematical capacities of the computer to perform calculations which were not normally required in the batch-processing environment. These calculations are not normally required for two reasons. First, in conventional situations it is difficult to obtain such computations without substantial modifications to already existing (and apparently working) programs. This is a practice clearly frowned upon in most situations. Second, it may simply be that the users of these reports have not realized the potential value of such computations. In an interactive situation the usefulness of such computations becomes obvious. For example, when attempting to understand the effects of a certain change in one of the parameter values, it may be very useful to be able to focus on such things as which budget figures change the most. Such data allow the user of the model to direct his attention to important changes. Where appropriate, even more complicated measures, such as variances, can be used. All of this is simply left up to the ingenuity and creativity of the user of the model.

Program Structure

The original batch processing version of this program was written in FORTRAN II Language. The current version of the program is in MAD, with a few subroutines in assembly language for the IBM 7094 computer. The present

program occupies about 1/3 of the 32,768 available words of computer memory. The program also allows the user to make use of the rather extensive volume of secondary storage (disk) available so that the state of several budgets may be maintained. The program, as will be made clear when we discuss the command language, also allows two budgets to be maintained in core. It is through this facility that it is possible to compare one budget with another.

The program operates through what might be called a command language. Commands, which will be discussed in detail in the next section, are simply key words typed by the user at his console. The program scans lines of input looking for key words, and as they are found certain actions are taken. The rules for scanning the input line are quite simple. The basic interface is through an input analyzer which separates the input line according to some elementary rules. First, the basic break[3] character in the input line is the blank. The fields of the line, as defined by the break character, are handed one by one to the central control of the program. A field is regarded as an alphabetic field if it contains any alphabetic character. It is regarded as a numeric if it consists only of digits with perhaps a decimal point or a plus or minus sign. The central control of the process handles the interpretation of the fields. The contents of each field starting from the left are looked up in a table of legal commands. If the field is present in that table, then it is a legal

---

[3]A "break character" is the character which separates one field from another. In standard English break characters are blanks and several of the punctuation characters. In the current version of this model, however, the only break character is blank.

command and control is transferred to the appropriate section of the program. If the field is not in the command table, and therefore does not represent a command, it is simply neglected. This means that the user is free to type any words which are not commands in the command line and the system will neglect them. This allows the commands to look much more like normal English than might otherwise be the case. Some of the commands expect further qualification, others do not. For example, the command COMPUTE (which tells the model to proceed with incorporation of any new parameter values) stands alone. It does not need to be qualified by any further statement. On the other hand, the command CHANGE expects to be followed by the name of a parameter and then a new value for the parameter. We might say for example:

"change the vfsw to 0.06".

This command would direct the system to change the variable cost of factory salaries and wages (vfsw) to $0.06/unit. It is the responsibility of the part of the program dealing with the execution of a given command to obtain the appropriate parameters from the command line.

As the above rules suggest, the interpretation of the command line is sensitive to the order of the command words. The word COMPUTE will not be interpreted the same way if it occurs immediately after the word CHANGE as it would be if that word had not occurred. In other words, when the program is looking for the name of a variable to change, it will not recognize other command words. Therefore, in order to make sure that the user is not confused by these rules, one command word which is recognized at any point is "ALTERNATIVES". If the user gets confused at any point he may say

"give me a list of alternatives"

and the system will respond first with a statement of the kind and how many alternatives are available and then, should the user desire, an actual list of the alternatives themselves.[4] It should be noted that this accomplishes two objectives. First, it frees the user from dependence upon manuals and directions which he may not have with him at the time he is actually sitting at the console. Second, and somewhat more importantly, it allows the program to be developed, to a certain extent, independent of its documentation. In a sense, the program is self-documenting. Alternatives can be added and if the user is not aware of them they will not interfere with his normal computation. Should, on the other hand, he find that the program is changing in certain dimensions, he may ask for a list of alternatives and find out about the new one. As will be noted below, there are also other facilities which help support the user in this regard. One may comment at this point that such facilities are characteristic of a change from batch processing to interactive environments. The user in an interactive environment typically demands much more flexibility in available alternatives and much more availability of supporting information than does the batch processing user. This is simply because the interactive user is at the time actually using the computational power, whereas the batch processing user can have recourse to manuals and discussions with other people without interrupting, in any effective sense, the course of his computation.

---

[4] For example in a COMPARE instruction the response would be:

THERE ARE 4 ALTERNATIVES CALLED KINDS OF COMPARISON.
DO YOU WANT TO SEE THEM.

Using the Model

In Appendix C the actual procedure for logging in to CTSS and activating the model will be discussed. In this section we will presume that the user has successfully logged in and has activated the model. We will discuss the commands available inside the model. When the model is activated, all parameters have initial values. These are the values specified in the original statement of the problem.

The basic command of the model is the verb COMPUTE. This command tells the model to take all the changes in the parameters and compute the effects of these changes in the actual output reports. Since computation of all the relations in the model is rather time-consuming, it would be very expensive to recompute the model every time a single parameter is changed. This is particularly clear when on considers the fact that often several parameters will be changed. By making the compute verb explicit it is thus possible to change several of the parameters without necessitating any computation, and then when all the changes have been made compute the affects of the changes. The verb COMPUTE does not produce any output. That is left for other commands of the language.

There are four basic commands for obtaining the reports. The command CASH causes a six-month cash budget to be printed on line. Similarly, the command PROFIT causes a profit budget to be produced, the command OVERHEAD, an overhead budget, and the command INVENTORY an inventory reconciliation. All of these reports are produced at the console and are reasonably lengthy, therefore they are typically used only when the user desires all the information in the report. Otherwise the command OUTPUT is used to

selectively look at pieces of information. The exact specification of the OUTPUT command will be discussed in Appendix B, which describes the details of all of the commands.

Several commands can be used to effect changes in values of basic data and parameters. The command CHANGE allows the values of the financial parameters listed in Table One of Appendix B to be changed. The basic form of the command is "CHANGE" followed by the name of the thing to be changed followed by a numerical value. Of course since most words are acceptable in a line, one can say

"change the value of vfsw to 0.06".

The commands SET, ADD and MULTIPLY allow changes to variables or whole classes of variables. For example, one can use the command ADD to increment, in a consistent pattern, the production or sales figures of a given product.

The commands SPREAD and DO NOT SPREAD may be given and the system will set a switch which will cause it to spread all overhead costs or not to spread them the next time a compute is requested. If overhead costs are spread then all the variance accounts will be forced to zero by the end of the six-month period covered by the model. On the other hand, if the command DO NOT SPREAD is given then the system will use the previously established burden rate and allow variances to accumulate.

There are several commands available to allow the user to interrogate various files of information. The command VARIABLES produces a list of the current values of the financial parameters. The commands ITEMS and

ABBREVIATIONS each produce lists of some of the abbreviations used in the system. The command HELP gives the user some indication of the kind of options at his disposal.

There are also a series of commands dealing with saving and restoring budgets and comparing one budget with another. The basic commands of this set are SAVE and RESTORE. The command SAVE causes the current state of the model to be saved in a named file in disk storage which later can be restored by use of the RESTORE command. It can also be used by the COMPARE command which allows comparison of one budget to another. The arguments of COMPARE are a statement of what budget is to be compared and what type of comparison is to be performed. Among the options available for kinds of comparison are such things as differences between the budgets, the absolute value of those differences, the ratio of items in one budget to another, the absolute value of such ratios, etc. When a comparison is performed, the system makes a list of all of the values of the compared variables in sorted order. Thus the commands MAXIMUM, MINIMUM and NEXT can be used to print the variables which have changed the most or the least from one budget to the next. This set of commands is extremely useful in allowing the user to selectively interrogate parts of the model and to find out what parts of the model are most sensitive to changes in the values of the basic parameters.

The command KEYWORD can be used to determine the abbreviation for a given variable name or to find out what the given abbreviation means.

In Appendix B we recapitulate statements of all legal commands and further give an exact statement of the kind of parameter sequences which are acceptable to those commands.

Future Work

Experimentation with the current version of the budgeting program has made some points clear. First, it is difficult to use the present model to study the effects of parameter variation some sub-structure of the whole model. For example, if one wants to determine the effect of a change in factory salaries on production costs, it is necessary (with the present model) to compute the effects of this change throughout the entire model. While one need only look at the results with respect to production costs, the present version of the model performs a computation of the values of all the variables. Thus it makes no use whatsoever of what is the final object of computation, and every time that the COMPUTE command is executed the entire set of values of all variables in the model are re-computed. Instead if we were to allow the desired output to determine which intermediate variables are calculated, we may be able to avoid orders of magnitude of the comparison.

A second deficiency that experience makes clear is the inability of the present model to accept procedures written in the command language. It seems to be a characteristic of interactive systems that as soon as one begins to gain experience in using them, it becomes desirable to build procedures in the language that is being used. For example, one might define a procedure which would change all the basic cost figures by a given amount. This procedure could then be invoked to study the effects of a change of a given percentage in the cost of materials and labor.

Third, the present program does not record information about its own use. In other words, it does not keep track of what kinds of requests people make. Therefore, it does not fulfill the function of adapting to their usage patterns in any manner. One of the most useful characteristics of many human systems is the ability to adapt over a period of time to their users so that if one, for example, always asks for something on a Monday morning a reasonably intelligent system after a while may cause it to appear on Monday morning even without have been asked for it. It is very difficult to know in general how to allow such adaptation, but it is quite clear that it is impossible to perform such adaptation given a model constructed in the present fashion. If, however, information about nature of requests over a course of time were maintained by the system then at least it would be possible to look for such characteristics. Needless to say, there is no guarantee that a pattern of consistent usage would develop, but there seems to be good reason to suspect that most people use such systems in a fairly regular fashion.

Finally, it is clear that the present system does not allow modification of the structure in even the most trivial of ways. For example, it is a difficult task to add to the present program a new product, or even a new grade of a product. The present program is constructed with the idea in mind that only four grades are to be produced and therefore accounts for only four different categories of products are allowed. It would seem quite easy to design the model in such a fashion that simple changes in the structure would be quite easy, in fact, to adapt and could be done even by the user on-line. This is, of course, impossible given the present

structure where the data are all incorporated in data files and the procedures are all incorporated into a program which is compiled before execution and then run under the user's control. It is thus impossible, while operating the program, to request a recompilation because that would necessitate restructuring of all data files. If, however, the structure were flexible enough it would be possible to define new classes of data.

Work is already under way to provide models which satisfy the full requirements specified above along with those which were discussed earlier in this chapter.

APPENDIX A

Paraphrase of the Problem from Shillinglaw

Assume a company which produces and sells four grades of shoes. In this problem we will not be concerned with further style or with size variation. The sales budget January-June is as follows:

Grade	Jan.	Feb.	March	April	May	June	Total
A	3,900	5,700	7,300	6,400	5,900	5,300	34,500
B	9,400	12,700	21,000	16,800	13,100	13,000	86,000
C	14,300	20,200	29,300	25,500	21,100	20,600	131,000
D	17,200	24,500	33,900	31,000	25,800	23,900	156,300
Total	44,800	63,100	91,500	79,700	65,900	62,800	407,800

The production budget over the same period is:

Grade	Inventory Jan. 1	Jan.	Feb.	March	April	May	June	Total
A	5,200	6,400	6,200	6,000	5,600	5,400	4,700	39,500
B	8,700	17,300	14,900	13,800	13,300	13,500	12,300	93,800
C	16,300	26,100	22,500	19,400	18,600	20,300	20,900	144,100
D	18,700	30,800	28,700	26,500	23,700	24,200	16,700	169,300
Total	48,900	80,600	72,300	65,700	61,200	63,400	54,600	446,700

The following data are also available for each grade:

Grade	Selling Price	Direct Materials	Direct Labor	Machine Royalties	Factory Overhead	S and A Expense	Profit
A	$7.90	$2.04	$3.48	$0.334	$0.286	$1.185	$0.447
B	6.90	1.67	3.06	0.232	0.286	1.035	0.617
C	3.90	0.98	1.85	0.126	0.286	0.585	0.073
D	2.90	0.63	1.23	0.060	0.286	0.435	0.259

The overhead budget is as follows:

	Fixed Charges per Month						Variable Costs
	Jan.	Feb.	March	April	May	June	
Factory overhead:							
Salaries & wages	$ 6,188	$ 5,980	$ 6,188	$ 6,136	$ 6,188	$ 6,136	$0.052/pr.
Supplies	....	....	....	....	....	....	0.026/pr.
Fuel, power, & light.....	1,550	1,430	940	420	330	190	0.011/pr.
Depreciation	2,470	2,470	2,470	2,470	2,470	2,470	....
Other	3,446	3,446	3,446	3,446	3,446	3,446	0.003/pr.
Total	$13,654	$13,326	$13,044	$12,472	$12,434	$12,242	$0.092/pr.
S and A expense:							
Commissions	....	....	....	....	....	....	5.0% of sal
Salaries	$12,352	$12,352	$12,352	$12,352	$12,352	$12,352	....
Advertising	16,520	14,380	15,100	13,200	12,450	12,350	....
Bad debts	....	....	....	....	....	....	1.0% of sal
Other	1,100	1,100	1,100	1,100	1,100	1,100	....
Total	$29,972	$27,832	$28,552	$26,652	$25,902	$25,802	6.0% of sal

For budgeting purposes the following assumptions are made:

(1) Wages and salaries are paid 80 percent in the month in which they are earned and 20 percent in the following month.

(2) Sales commissions are paid in the month following the month in which they are earned.

(3) Materials and supplies are paid for in the month in which they are used.

(4) Federal income taxes are accrued monthly at 30 percent of any income up to $2,083 and at 52 percent for all income in excess of $2,083. The company had no income tax liability in the preceding year. No income taxes will be paid during the months of January through June.

(5) All other cash costs and expenses are paid in the month following the month in which they are accrued.

(6) Receivables are collected 30 percent in the month of shipment, 50 percent in the following month, and 19 percent in the second month after shipment. The remaining 1 percent are written off as uncollectible.

(7) A factory burden rate of $0.286 per pair is used in costing production. Manufacturing variances are closed to profit and loss semiannually at the end of June and December.

(8) Factory wages and salaries and other cash factory costs accrued in December and to be paid in January amount to $45,000; cash selling and administrative expenses, other than sales commissions, accrued in December but payable in January amount to $25,000.

(9) Accounts receivable on January 1 amount to $210,000 from December sales and $57,000 from November sales. The allowance for bad debts has a January 1 balance of $5,850.

(10) Capital expenditures of $5,000 a month are scheduled for the first six months of the year.

(11) Dividends of $10,000 are to be paid in March and in June.

(12) The company will have a cash balance of $100,000 on January 1 with no bank indebtedness. It is the company's policy to borrow from a local bank to meet peak seasonal requirements for cash. It has a line of credit permitting it to borrow at an interest rate of 6 percent per year in multiples of $10,000. Funds are borrowed and repaid on the first of the month. The company maintains a minimum balance of $80,000 or 20 percent of any outstanding bank debt, whichever is greater. Interest is paid monthly on the last of the month.

APPENDIX B

Tables of Keywords and Abbreviations

The first part of this appendix gives a list of the system commands and the various parameters of these commands. It should be remembered that this list does not contain "noise" words which may be added at the user's pleasure to improve readability.

In the discussion of commands we use the abbreviation UAR to represent any one of the items listed in Table 2, the abbreviations of the variables. The discussion of the commands is divided into several categories:

1) Basic Computation
2) Report Generation and Interrogation
3) Access to Information
4) Modification of Items
5) Miscellaneous.

The commands will be discussed by category. Table 3 contains a list of the commands in alphabetical order, cross-referencing them to each section.

1. Basic Computation

COMPUTE

This command causes the system to re-compute all of the values of the variables which are derived from the ones which can be set by the user. The appropriate procedure is to make all changes desired and then perform a computation. This saves a substantial amount of computer time over recomputing everything each time any change is made.

2. Report Generation and Interrogation

VARIABLES

This command causes a list of the current value of all of the variable parameters listed in Table 1 to be printed.

PROFIT

This command causes a profit budget to be printed.

CASH

This command causes a cash budget to be printed.

OVERHEAD

This command causes a statement of the allocation of overhead to be printed.

INVENTORY { A / B / C / D / ALL }

This command causes a reconciliation of inventory for the grade named or all grades to be printed.

The next four commands deal with the items in the model which are placed on an ordered list by the compare instruction (see Section 5).

MAXIMUM N

This command prints out the N items on the list which have maximum values.

MINIMUM N

This prints out the list of the N smallest items.

NEXT N

Continuing in the same direction (after a MAXIMUM or a MINIMUM) this causes the next N items to be printed.

SKIP N

This causes the next N items to be skipped. Thus one can say PRINT THE MAXIMUM 5 SKIP 3 AND THEN PRINT THE NEXT 4.

	VALUES	VARIABLES	
	MEAN	SALES	A
OUTPUT	DEVIATIONS	PRODUCTION	B
	SD	BOM	C
		EOM	D

This obtains a list of the values, the mean, the deviations (number of standard deviations from the mean) and the standard deviation of any items in the model.

	VARIABLES		
	SALES	A	
OUTPUT VALUE	PRODUCTION	B	MONTH
	BOM	C	
	EOM	D	

This obtains a single value and prints it out.

3. Access to Information

HELP

This causes a statement of some of the things that the model can do to be printed for the user.

ITEMS

This causes a complete list of the items (as listed in Table 2) and their abbreviations to be printed.

ABBREVIATIONS

This causes a list of the parameters (as listed in Table 1) to be printed.

ALTERNATIVES

This command causes the system to print out what the system is expecting as input, and it asks the user if he wants to see a full list of all of the alternatives available.

KEYWORD WORD1 ... WORDN *END*

The items file is searched for all occurrences of WORD1 ... WORDN and each line which contains any one of the words is printed.

4. Modification of Items

CHANGE T1ITEM NUMB

This command allows the user to change any item listed in Table 1 (T1ITEM) to a new value NUMB. Notice that, as with all of the modification commands, the change is not effective until a COMPUTE command is given.

{SET / ADD / MULTIPLY} {VALUE / VALUES} {VARIABLES / SALES / PRODUCTION / BOM} {A / B / C / D} {MONTH} {NUMB}

These commands allows the user to set, add to or multiply by an individual in a given month or in all months a given amount.

{NOT} SPREAD

This causes a switch to be set such that the next time a computation is performed, either overhead will be spread and a new burden rate established, or it will not be spread and the old burden rate used.

5. Miscellaneous

SAVE
RESTORE NAME

These commands cause the current state of the model to be saved in disk storage for later use, or restored from disk storage. The file created is NAME BUDGET. There cannot be any "noise" words between the command and the name. Different budgets can, of course, be saved under different names.

COMPARE NAME { ABSOLUTE } { RATIO / DIFFERENCE / D-RATIOS }

This command causes the current state of the model to be compared with the state recorded in file NAME BUDGET. The comparisons are NEW/OLD, NEW-OLD or (NEW-OLD)/OLD or the absolute value of any of these. After the comparison is performed, the list is sorted into descending order and the items on the comparison list can be obtained by the MAXIMUM, MINIMUM, NEXT and SKIP commands.

COMPLETE

This terminates the model.

Table 1

Abbreviations for Parameters

PSC = Percentage Sales Paid in Comissions

PFM = Percentage of A/R Received in First Month

PSM = Percentage of A/R Received in Second Month

PTM = Percentage of A/R Received in Third Month

(The remainder is bad debt.)

PFP = Percentage of Factory Wages Paid in Current Months

ROH = Overhead/Unit

BLIM = Minimum Bank Balance

DTD = Multiple for Increments to Debt

VFSW = Variable Factory Salaries and Wages (per Unit)

VFS = Variable Factory Supplies Cost (per Unit)

Table 2

Abbreviations for Variables

(Only those items marked * can be changed by the user, as the other items are computed from basic data in the model.)

ADIS		Adjusted Disbursements
ADV	*	Advertising
ANR		Adjusted Net Receipts
ARCT		Adjusted Receipts
ATIN		After Tax Income
BAD		Bad Debt
BOR		Borrowing
BTIN		Before Tax Income
CAP	*	Capital Additions
CGF		Cost of Goods at the Factory
CGS		Cost of Goods Sold
DIV	*	Dividends
DL		Direct Labor Expense
DM		Direct Materials Expense
EBAL		Bank Balance at End of Period
EDT		Debt at End of Period
EINN		Interest (Cash Flow)
EINT		Interest Expense
FD	*	Factory Depreciation
FFFPL	*	Fixed Factory Fuel Power and Light
FFOE	*	Fixed Other Factory Expenses
FFPL		Total Factory Fuel Power and Light
FFSW	*	Fixed Factory Salaries and Wages
FOE		Total Factory Other Expenses
FS	*	Factory Supplies
FSW		Total Factory Salaries and Wages
GMAR		Gross Margin
OA		Overhead Allocated
OPIN		Operating Income
PAY		Payments
PAYN		Net Payments
REV		Revenues
RY		Royalties
SC		Sales Commissions
SUE		Other Sales Administrative Expenses
SR		Sales Revenue
SS		Sales Administrative Salaries
TAX		Taxes
TCOL		Total Collections

Table 2 (Continued)

TDIS		Total Disbursements
TF		Total Factory Expenses
TSA		Total Sales/Administrative Expenses
UC		Unit Cost
UDL	*	Unit Direct Labor Expense (for each grade)
UDM	*	Unit Direct Materials Expense (for each grade)
UMR	*	Unit Machine Royalties (for each grade)
UOA		Unabsorbed Overhead
UP		Unit Production (for each grade)
US		Total Unit Sales
USP	*	Unit Sales Price (for each grade)

Table 3

Alphabetical List of Commands

ADD	4
ALTERN	3
CASH	2
CHANGE	4
COMPAR	5
COMPLE	5
COMPUT	1
HELP	3
INVENT	2
ITEMS	3
KEYWOR	3
MAXIMU	2
MINIMU	2
MULTIP	4
NEXT	2
NOT	4
OUTPUT	2
OVERHE	2
PROFIT	2
RESTOR	5
SAVE	5
SET	4
SKIP	2
SPREAD	4
VARIAB	2

1 -- Basic Computation

2 -- Report Generation and Interrogation

3 -- Access to Information

4 -- Modification of Items

5 -- Miscellaneous

APPENDIX C

Logging in to CTSS

To activate the model described in this paper the user must first log in to CTSS and then activate the model.

1. To log in to CTSS

Dial 8 (MAC) or 0 (Comp. Center) from a 1050 or 2741 (typewriter)

9 (MAC) or 7 (Comp. Center) from a teletype

LOGIN NUMBER NAME

The number and name will be different in each circumstance.

The machine will respond:

W TIME (giving the time of day)

PASSWORD

At this point type in the password that will also be given to you. After a long message the system will finally type:

R N.NNN + N.NNN (giving some numbers). At this point type:

R BUDGET

and you will have activated the budgeting model.