Journal of the Operational Research Society (1999) 50, 70–74 0160-5682/99 $12.00
http://www.stockton-press.co.uk/jor

A framework for building spreadsheet based decision models

D Mather
Monash University, Australia

Spreadsheet based decision modelling is widely used in business today. The functionality provided within the current integrated packages allows users to develop very complex business models. Unfortunately, most spreadsheet users do not follow any particular methodology when building spreadsheet based models. This in turn leads to many wasted hours of debugging and re-programming and often produces models that look more complex than they need to be and are also tedious to extend or modify. This paper presents an example of the type of framework that can be used when developing spreadsheet based business models. The framework encourages the efficient development of more effective models that are easy to use and easy to modify.

Keywords: decision support systems; model building; OR education; risk analysis; spreadsheets

Introduction

Most business organisations collect and manage data on a MS Windows platform. With integrated spreadsheet packages providing users with business modelling tools that are both easy to use and effective, there is an increasing trend in the current business environment to use a spreadsheet platform to build decision models. The main benefits of developing spreadsheet based business models are their transparency and portability characteristics.

One of the factors that usually gets in the way of a smooth model development process is that most spreadsheet users do not follow a framework for model development. Often, people start at the top left hand corner of the first worksheet and cut, move and copy cells and introduce variables and formulae as they need while developing various aspects of the model. Whilst these approaches invariably lead to a completed model, the efficiency of the model development task itself and the quality of the model produced does suffer. Although most business related graduates learn how to use a spreadsheet and how to model business problems, few learn any framework that is useful specifically for building spreadsheet based business models. A recent study of a sample of organisations that used spreadsheet modelling reported some alarming statistics relating to the poor quality of the models developed.[1] Several other researchers in this area have also documented the very high usage of spreadsheets in business and noted in addition the problems associated with the open and unstructured format of spreadsheets. [2,3]

Correspondence: Dr D Mather, School of Business Systems, Faculty of Information Technology, Monash University, Clayton, Victoria 3168, Australia.
E-Mail: DMather@infotech.monash.edu.au

As in all aspects of programming, the way one chooses to develop a spreadsheet model is very subjective. Building decision models on spreadsheets using a structured and systematic approach that encapsulates the whole problem produces a model that is not only easy to use but easy to extend, and is usually more efficient in the generation of meaningful output. The framework presented here incorporates a mixture of existing, modified and new techniques and has been found to be extremely effective for quantitative risk analysis models and other business models that contain a large number of variables. This is mainly due to the fact that the approach encourages the modeller to categorise model variables according to their function and use a structured spreadsheet layout, which not only makes the task of finding key inputs and outputs easier but creates a logical platform for sensitivity and risk analysis.[5]

The framework

A framework for building spreadsheet based decision models is presented below. The use of the key aspects of the method will then be illustrated with the use of a fictitious case study. The process is as follows:

Identify inputs and outputs

The first task is to define one or many outputs that are required from the model in order to achieve the objective of the task. The inputs are data and information that are required to produce those outputs.

Document the logic flow

Link inputs and outputs using a series of user-defined intermediate or calculation variables. Document the flow of information between all the variables using a graphical representation method such as a bubble diagram or an influence diagram.[4,5]

Categorise the variables into the following

- *Variable inputs*:
  quantities that are likely to change during the timeframe of the project. It may be that, within the scope of the project, some quantities that may appear to be variable should be considered as constants. Some of the variable inputs will be decision variables. Separate the variable inputs into:
  —deterministic inputs.
  —stochastic inputs.
- *Constant inputs*:
  quantities that can be considered to be constant for the scope of the project.
- *Intermediate variables*:
  variables that have been introduced into the model to link inputs and outputs. The choice of intermediate variables is subjective, but, as long as the processing is accurate, the actual number of these variables is not critical, and it is always better practice to use more rather than less. This is because breaking down the processes into smaller steps makes the programming easier to follow, easier to debug and overall reduces the need to use complex formulae, thus reducing possible sources for errors.
- *Performance measures/key outputs*:
  model outputs central to the objective of the task.

Some general rules for programming the spreadsheet are

- There should only be a single point of entry for the values of all variables. Other cells that require this quantity should copy the cell reference. This ensures that there is a single location for updating values and there is no resulting inconsistency.
- Constant values should not be used in formulae. A formula should contain functions and cell references to where the constant value is stored. Although this may seem rather tedious it is a safer method to use as it makes it easier to debug and modify the model.

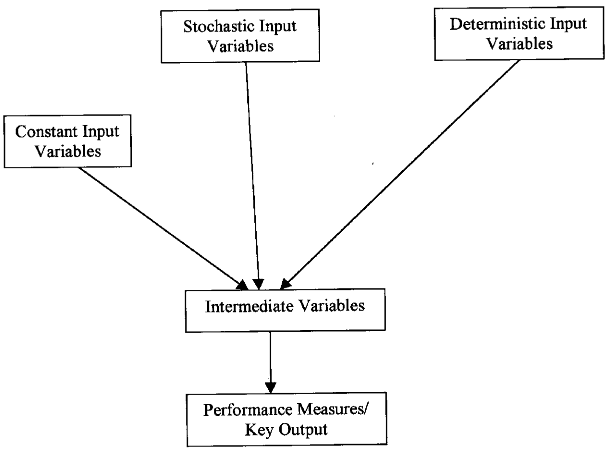

Figure 1 The flow of information between the categories of variables.

Divide the worksheets

The modeller can make best use of the multiple worksheet (page) format of spreadsheet workbooks by allocating specific tasks to separate pages. The basic idea is to allocate variables to workbook pages in order to create a shopfront-backroom system.

- *The Manager Page*:
  The Manager Page serves as an interface to the decision model. The primary source of the decision variables of the model should be on this page. The other types of variable that should appear on the Manager Page are the performance measures/key outputs. The objective of this method is to provide a specially designed area of the worksheet that can be used to view results and carry out sensitivity analysis. This ensures that the user can change the values of decision variables and immediately view the effect of that change on the performance measures or bottom-line, without needing to wade through rows of intermediate calculations. By containing only the key inputs and outputs, the Manager Page contains only, and all of, the information that needs to be displayed. The Manager Page can also be used for buttons that activate macros that either carry out some processing or display worksheet objects such as graphs and dialog boxes.
- *The Constants Page*
  This page is used for constant inputs. This is the only point of entry for constant inputs. In a risk analysis model, empirical probability distributions and (constant) parameters for theoretical distributions will also be stored in this page.
- *The Calculation Page*
  The Calculation Page contains all the intermediate variables, usually in the form of formulae. Input variables used in the calculations are copied from either the Manager Page or the Constants Page. In a risk model, stochastic inputs will also be placed in the Calculation Page. If the values are sampled from theoretical distributions, the parameters will be copied from the Constants Page. Alternatively, if empirical distributions are used to generate the variables, the formulae will sample directly from the (cumulative distribution) tables in the Constants Page. The Manager Page extracts or copies output values generated in this page to display the key outputs. In a very

large model, it may be useful to use several calculation pages and allocate submodels to each. The Calculation Page(s) should be documented using a rules table[5] or a similar object.

- *The Graphs Page (optional)*
  In business models it is often useful to have dynamic graphs, with the chart input range responding to any changes in the worksheet simultaneously. A series of pointers in the form of button driven macros can be used to display specific graphs on request. These buttons should be placed on the Manager Page.

Test and audit the model

Debugging spreadsheets can be rather monotonous and laborious. Testing and auditing the model regularly at various stages of its development is usually the only way to avoid having a long-winded debugging session at the end. The following types of tests provide a reasonable foundation for identifying possible programming errors.

- *Zero tests*
  Set single or groups of inputs to zero and ensure the output values are as should be.
- *Logic tests*
  With each input variable, increase and then decrease the values to ensure the changes to the outputs are in the logical direction as well as of appropriate magnitudes.
- *Manual tests*
  Substitute a series of small, easy to manipulate, values for the inputs and compare a manual calculation of the outputs to those generated by the spreadsheet.

The auditing function provided on spreadsheets is a useful tool to identify precedents and dependents of formulae in order to trace the source of an error.

A worked example

The case study used below to illustrate the method is a typical textbook example of an inventory model. A firm that sells a particular product needs to make a decision about the optimal reorder quantity, Q, and reorder point, r, over a period of 10 weeks. The weekly demand for the product as well as the lead time to delivery, once an order is placed, are uncertain. Historical data has been used to construct frequency distributions for each of the variables.

Defining and classifying inputs and outputs

Constant inputs

- Unit purchase cost
- Unit storage cost
- Ordering cost (fixed)
- Unit shortage cost (unmet demand)
- Empirical distributions of the weekly demand
- Empirical distributions of the lead time

All these inputs will be stored in the Constants Page.

Variable inputs

- Weekly demand (stochastic)
- Lead time (stochastic)
- Maximum order size (decision variable)
- Re-order point (decision variable)

The two decision variables will be stored in the Manager Page. The stochastic inputs will be placed in the Calculation Page.

Outputs

- Total cost over 10 weeks
- Average weekly costs (storage, shortage, order and total)
- Total shortage cost
- Total holding cost

All these outputs will be copied back to the Manager Page.

Documenting the flow of logic/information

This documentation is done using a representation similar to an influence diagram. Four different shapes of 'bubbles' are used in order to differentiate between constants, variable inputs, intermediate variables and key outputs. A 'D' is used to identify a decision input, a tilda or '~' to identify stochastic inputs and an asterisk '*' to indicate a link to a previous or future time period. The shapes are shown in Figure 2.

These conventions are used in the influence diagram of Figure 3. This diagram depicts the logic flow for the example in a typical week n of the weeks covered by the model. The intermediate variables that calculate the total weekly costs for week n, are then used as inputs to calculate the average weekly costs.

Note the following assumptions:

(a) The deliveries are received and accounted for at the start of a week.

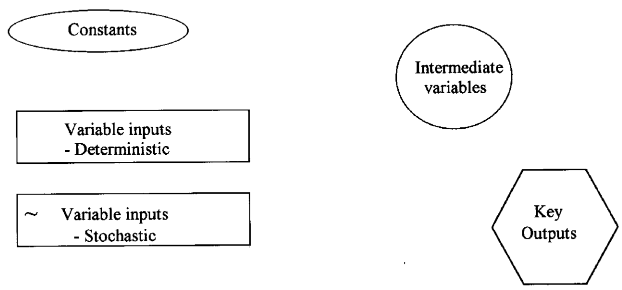

Figure 2 Shapes used for classes of variables.

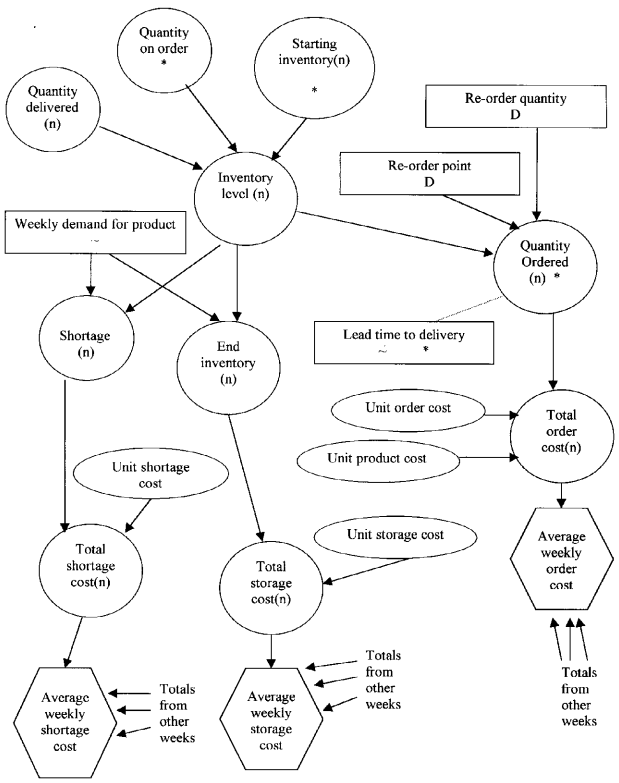

Figure 3 Documentation of the flow of information for the n^{th} week.

(b) The orders are placed at the start of a week, based on the inventory after deliveries are received, including outstanding orders. An order is placed if this inventory level is below the order point.
(c) The lead time to delivery is not influenced by the order quantity.

The model documentation looks at a typical week in the ten-week period. The spreadsheet calculations will consist of ten columns corresponding to the ten weeks. The link(s) between the columns are identified in the diagram with an asterisk. If the model had links that went beyond one time period, or links between different sub-models, these could be flagged using appropriate user defined symbols. Once again, the shapes and symbols used are entirely subjective; the only requirements are clarity and consistency.

The spreadsheet layout

The variables are distributed between the Manager, Calculation and Constants Pages as described earlier. This is quite easily done using the flow diagram as all variable types are clearly identified. The extracts below show these pages with some arbitrary values assigned to the inputs.

In the Manager Page, the values of the deterministic inputs (re-order quantity, re-order point) are changed manually by the user. The stochastic inputs (weekly demand, lead time) are generated by the spreadsheet using built-in statistical functions.

The deterministic variables and, parameters and probability distribution functions of the stochastic variables, are listed in the Constants Page (see Figure 4). All these values are copied onto the Calculation Page using their cell references.

The formulae contained in the Calculation Page (see Figure 5) take values from both the Manager and the Constants Page. Examples of such formulae are

cell F8 ‘IF(F6 + F5<Manager!$D$5, manager!$D$4,0)

and

cell C20 ‘IF(C15 = 0,0,Constants!$D$8*C15)

The outputs displayed in the Manager Page extract information from the Calculation Page. For example, the average shortage cost is calculated using row 18 of the Calculation Page as ‘AVERAGE(Calculations!C18:L18)’. To make best use of a model with stochastic variables, it is necessary to simulate through a number of repetitions and sumarise the bahaviour of the output measures, as in a typical risk analysis model. The Manager Page in Figure 6 is shown displaying results from a single simulation.

Conclusion

The case study presented above illustrates the general use of the proposed framework. Dividing the variables into different parts of the workbook, with no duplication and having these linked through formulae, makes the model a very efficient sensitivity analysis tool for the user. Any changes made to the input values, take place on the Manager Page and the resulting change in the key outputs is evident immediately without needing to move around looking for relevant variables. The Manager Page effectively acts like the ‘shop-front’ whilst the Constants and Calculation Pages provide the ‘back-room’ support.

As stated previously, there is no right or wrong way to build a spreadsheet-based model. Any model that works, in so far as that it provides the required output, is obviously an effective model. Frameworks, of the type presented in this paper, play an important role in making the model building

	B	C	D	E	F	G	H	I	J
2		DETERMINISTIC				STOCHASTIC			
3									
4		Ordering Cost (fixed	$300.00		WEEKLY DEMAND			LEAD TIME (Weeks)	
5					Mean	Std. Dev.		CPDF	Time
6		Unit Product Cost	$ 5.00		73	8		0	1
7								0.3	2
8		Unit Storage Cost	$ 8.00					0.8	3
9								0.95	4
10		Unit Shortage Cost	$ 12.00						
11									
12									

Figure 4 The Constants Page.

	B	C	D	E	F	G
2	Week	1	2	3	4	5
3	Starting Inventory	250	178	121	55	0
4	Quantity Received	0	0	0	0	250
5	Quantity on Order	0	0	0	0	0
6	Total Inventory Level	250	178	121	55	250
8	Quantity Ordered	0	0	0	250	0
9	Lead Time (weeks)	0	0	0	1	0
10	Next Order received	0	0	0	5	0
11	Total Product cost	$ -	$ -	$ -	$ 1,250.00	$ -
12	Total Order cost	$ -	$ -	$ -	$ 300.00	$ -
14	Weekly demand	72	57	66	74	76
15	End Inventory	178	121	55	0	174
17	Shortage	0	0	0	19	0
18	Total Shortage Cost	$ -	$ -	$ -	$ 228.00	$ -
20	Total Storage Cost	$ 1,424.00	$ 968.00	$ 440.00	$ -	$ 1,392.00
22	Total Cost	$ 1,424.00	$ 968.00	$ 440.00	$ 528.00	$ 1,392.00

Figure 5 An extract from The Calculation Page.

	B	C	D	E
2				
3		DECISION VARIABLES		
4		Re-Order Quantity	250	
5		Re-Order Point	100	
6				
7				
8				
9		OUTPUTS		
10		Average weekly shortage cost	$ 570.00	
11		Average weekly order cost	$ 120.00	
12		Average weekly storage cost	$ 251.20	
13		Average weekly total cost	$ 941.20	
14		Total cost	$ 9,412.00	
15				

Figure 6 The Manager Page with results from a single simulation

process more efficient and guard against programming disasters. However, it should be noted that the framework assumes that the underlying model is a good enough representation of the situation. The process of classifying the variables and documenting the logical structure of the model is, itself, often a very rewarding exercise. The systematic approach used here is an excellent way of identifying missing information, loopholes and other problems early in a project. This is because the development of the spreadsheet model and the resulting programming does not begin until the complete model structure, listing all the information that is required for the exercise, is identified and documented.

References

1 Cragg PB and King M (1993). Spreadsheet Modelling Abuse:An Opportunity for OR? *J Opl Res Soc* **44**: 8, 743-752.
2 Cornford T and Doukidis GI (1991). An investigation of the use of computers within OR. *Eur J Inf Sys* **1**: 131–140.
3 Mason D and Keane D (1989). Spreadsheets: solution or problem?, Interfaces **19**: October, 82–84.
4 Bodily SE (1988). Modern Decision Making ± A guide to modeling with decision support systems. McGraw-Hill: USA.
5 Sherwood D (1983). Financial Modelling — A Practical Guide. Gee & Co. (Publishers) Ltd.: USA.

Received September 1997;
accepted August 1998 after two revisions