

Chapter 7: Design Craft

Contents

Introduction	4
Model Structure	4
Figure 7.01: Key Models.....	4
Processing Sequence.....	5
Figure 7.02a: Overall Processing Cycle.....	5
Figure 7.02b: Period Processing Cycle.....	5
Initialise Simulation	5
Initialise Period	5
Decision Checking	5
Market/Economic	5
Accounting/Operations.....	5
Business Analysis.....	5
Ending Analysis	5
Subsidiary Models.....	6
Figure 7.03: Some subsidiary models (SEED Simulation)	6
Simulator Structure	6
Tutor Mediated Simulator Structure	6
Group Name	6
Figure 7.04: Group Name.....	6
Number of Teams	6
Figure 7.05: Decide Number of Teams	6
Select Version.....	6
Master Menu – Tutor Mediated Simulations	6
Quit Simulator	6
Figure 7.06: Master Menu	6
Enter Decisions & Simulate.....	7
Figure 7.07a: Printing Options.....	7
Figure 7.07b: Adjust Economy.....	7
Figure 7.07c: Decision Menu.....	7
Decision Entry for a team.....	7
Figure 7.07d: Team Name.....	7
Figure 7.07e: Decision Template.....	7
Figure 7.07f: Problem Decisions.....	7
Figure 7.07g: Print Decisions?.....	8
Simulation Process and Printing	8
Display Latest Results	8
Recover to Past Period.....	8
Special Options.....	8
Advice & Help	8
Direct Use Simulator Simulations.....	8
Processing Structure – Direct Use Simulations	8
Team Name	8
(Budget Entry).....	8
Decision Entry.....	8
Simulate.....	8
Results Print and Display.....	8
Ending Review.....	9
Tutor's Options – Direct Use Simulations	9
Quit Simulator	9
Reconfigure Simulator	9

Figure 7.8: Tutor's Menu	9
Delete Team Files.....	9
Copy Team Files.....	9
Tutor's Review.....	9
End and Review.....	9
Reinstate	9
Recover	9
Enable Full Version.....	9
Change Versions	9
Data Structures.....	9
Figure 7.09: Parameter and Team Data Structures	10
Figure 7.10: Modern Banking Data Structures.....	10
Variable Structures	10
Figure 7.11: Modern Banking Variable Structure	11
Other Data	11
Design Process Example.....	11
Figure 7.12a: Operational Mode	11
Figure 7.12b: Design Mode	11
Chunking and Refactoring.....	12
Modelling Language.....	12
Cartoon 7.01: Media.....	12
Cartoon 7.02: My early days.....	13
Flavour of the Month.....	13
Language Elements.....	13
Figure 7.13: Language Sub-set	13
1. Assignment	13
Figure 7.14: Examples of assignment.....	14
2. Conditional Statements	14
a. If...Then.....	14
Figure 7.15a: Example of If...Then structure	14
Figure 7.15b: Example of If...Then...Else structure	14
Figure 7.15c: Example of If ...Then...Elseif structure	14
b. Select Case	14
Figure 7.16: Example of Select Case statement.....	15
3. Loops	15
a. For...Next.....	15
Figure 7.17: Examples of For...Next statements	15
b. Do...Loop	15
Figure 7.18: Examples of Do...Loop statements.....	15
c. While...Wend.....	15
Variable Types, Operators and Functions	15
1. Variable Types	15
Figure 7.19: Variable Types	16
Figure 7.20: Market Research Flags.....	16
2. Arithmetic Operators	16
Figure 7.21: Arithmetic operators	16
Integer Division (\).....	16
Figure 7.22: Examples of Integer Division	16
Exponentiation (raising a number to a power)	17
Modular Arithmetic (the Mod function).....	17
Figure 7.23: Modulus.....	17
Figure 7.24: Examples of use of Mod function.....	17
3. Logical Operators.....	17
Figure 7.25: Logical operators	17

Figure 7.26: Logic using <i>And</i> to check for market potential research	17
4. Functions	17
Standard Functions.....	18
Figure 7.27: Standard Functions	18
Special Functions	18
Description.....	18
Figure 7.28: Special Functions	18
Figure 7.29a: Generating the <i>same</i> random sequence	18
Figure 7.29b: Generating a <i>different series</i> of random numbers	18
Figure 7.30: Generating project sizes.....	19
Figure 7.31: Using the Int function to round numbers.....	19
Figure 7.32: Using the Sin function to make a saturating response	19
Figure 7.33: Using the Sgn function.....	19
Figure 7.34: Function to calculate Log base 10	20
Documentation	20
Learner Documentation	20
Learners' Manual.....	21
Figure 7.35: Learner Manual lengths in A4/Letter size pages	21
Introduction	21
Figure 7.36: Product Launch Introduction	21
The Business.....	22
Figure 7.37: Information about the Business (Product Launch)	22
The Decisions	22
Figure 7.38: Product Launch Decisions	22
The Results	22
(Business History).....	22
Figure 7.39: Product Launch Business History	23
Briefing Slides	23
Figure 7.40: Number of Briefing Slides	23
(Decision Forms).....	23
Figure 7.41: Sample Decision form	24
Cartoon 7.03: Life-Long Writing Skills.....	25
(Worksheets).....	25
Figure 7.42: DISTRAIN Worksheet.....	25
Figure 7.43: Modern Banking – Decision Record.....	26
(New Information).....	26
Figure 7.44: Example of New Information.....	26
(Additional Tasks).....	26
Report on Organisational Structure	26
Figure 7.45: Example of Additional Tasks.....	26
(Definitions).....	27
(Business Theory)	27
Figure 7.46: Introduction to booklet discussing financial measures	27
Online Documentation.....	27
Figure 7.47: Help Database Size.....	27
Software Help	28
Decision Entry.....	28
Figure 7.48: Example of Software Help	28
Task/Report Help.....	28
Pre-qualify the Project.....	28
Figure 7.49: Example of Task Help	28
Explanations/Definitions.....	28
Tender Price	28
Figure 7.50: Example explaining a decision.....	28
Manual.....	28

Tutoring Documentation.....	28
Background to the Simulation.....	28
Figure 7.51: Number of A4/Letter size pages in Background Notes.....	29
Description of the simulation.....	29
Figure 7.52: Description of Modern Banking.....	30
Specimen Timetables.....	30
Figure 7.53: Modern Banking Timetable.....	31
Simulation Models.....	31
Figure 7.54: Processing Sequence for Modern Banking.....	31
Decisions and Issues.....	31
Figure 7.55: Example of discussion of decision and issues.....	32
Descriptions of the Reports.....	32
Run Statistics.....	32
Figure 7.56a.....	32
Figure 7.56b: Sample results (Prospector).....	32
Using Simulations for Training.....	32
Using the Software (Simulator).....	33
Documentation during design.....	33
In Software Documentation.....	33
Figure 7.57: In model comments.....	33
Suitable variable names.....	34
Formatting Code.....	34
Number of Statements per Line.....	34
Model Documentation.....	34
Figure 7.58: Report in Design Mode.....	34
Build online help database.....	35
Figure 7.59: Variable help record.....	35
Quality Assurance Documentation.....	35
Figure 7.60: Creditor and Assets cross check for SEED.....	35
Calibration Documentation.....	35
Figure 7.61: Innovation Diffusion Calibration for SEED.....	35

Introduction

This chapter explores the software craft aspects of simulation design and as such covers:

- § **Model Structure**
- § **Simulator Structure**
- § **Data Structures**
- § **Design Process Example**
- § **Modelling Language**
- § **Documentation**

Model Structure

I use two structures for the models. One is a general structure (Figure 7.01) that structures the main models and is common to most of my simulations. The other is for the small, subsidiary models that are used by the main models to do interim calculations (Figure 7.03). These are ones that I use for my more complex simulations.

Model	Description
Initialise Simulation	This initialises the simulation's starting parameters
Initialise Period	This initialises data at the start of the simulated period.
Decision Checking	This checks decisions as they are entered.
Market/Economic	This contains the "black" box models.
Accounting/Operations	This contains the "white box" (internal) models.
Business Analysis	This analyses results after the period is simulated.

Figure 7.01: Key Models

Processing Sequence

Figure 7.02a shows the overall processing sequence and Figure 7.02b shows the sequence for a single period.

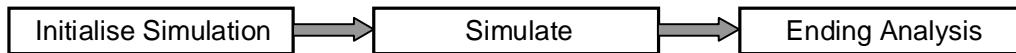


Figure 7.02a: Overall Processing Cycle

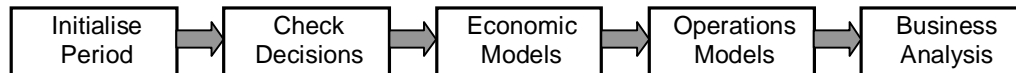


Figure 7.02b: Period Processing Cycle

Initialise Simulation

This subroutine defines simulation defaults (such as the name of the standard control file, demonstration and full use parameters). These defaults can be modified by control files to allow the simulation to have several versions.

Initialise Period

This subroutine initialises data for the period and, in particular, decisions, the starting positions. For example, in Modern Banking, cumulative incomes and costs are zeroed at the first period and closing assets, staff levels etc. at the end of the previous period are transferred to opening balances so that opening and closing balances can be seen. For DISTRAIN, although the other decisions are transferred unchanged, the Market Research decision is zeroed (as this decision would not normally be unchanged). Often it is appropriate to have additional logic that is only processed during the first simulated period. This is to initialise cumulative results and for Direct Use simulations it can be desirable to establish some initial parameters or allow the learners to choose the version to use. For example, it might be essential for learners to have to face and deal with several economic environments (with different growth rates, inflation etc.). Here the learners might select from a list of economic scenarios or the simulator might *randomly* determine the economic environment.

Decision Checking

As described in Chapter 5 (Drafting the Simulation), this subroutine checks decisions as they are entered.

Market/Economic

As described in Chapter 5, this contains the *black box* models. For interactive simulations (such as Modern Banking and DISTRAIN), this subroutine assesses across the competing teams how their decisions impact the market and sales. For Direct Use simulations (such as Product Launch), these models can be combined with the Accounting/Operations models.

Accounting/Operations

This contains the *white box* models performing the calculations need to determine the *internal* operation of a business (e.g. staffing (for Modern Banking) and materials flow (for DISTRAIN)). From these, the Profit and Loss Account (Income Statement), Balance Sheet and Cash Flows are calculated.

Business Analysis

This takes business results and calculates performance measures. For Modern Banking and DISTRAIN, this routine also compares team performance and generates data for the Market Research reports. Additionally, for all simulations, I use this to produce comments on strengths, weaknesses, etc..

Ending Analysis

This is used at the end of Direct Use simulations to analyse overall performance and where appropriate forecasts outcomes. For example, Product Launch compares

cumulative profit for the current team against some hundred previous runs and comments on relative success (or failure). Additionally it provides statistics that help identify reasons for success (or failure). For Prospector, the Ending Analysis takes the portfolio of projects created during the simulation, implements them and predicts the outcomes in terms of profit, cash flow etc.

Subsidiary Models

For complex simulations, I use these (Figure 7.03) to perform common calculations and simplify structure.

Model	Description
Asset Purchases	Costs and cash for asset purchases
Calculate NPV	Net Present Value of cash flow
Past Sales Revenue	Accumulates Sales Revenue for designated year
Sales Income	Financial income and administration costs for sales
Target Finished Inventory	Inventory weeks based on operation type

Figure 7.03: Some subsidiary models (SEED Simulation)

Simulator Structure

Chapter 6 (Design for Value) explored the architecture that I use. Here I explore the elements of the simulator, separately for Tutor Mediated and Direct Use simulations. To an extent this links with the Design for Value (Chapter 6) as the software structures are designed for easy, speedy, productive use.

Tutor Mediated Simulator Structure

Here the simulator is used by the trainer (Modern Banking, DISTRAIN and SMITE are all Tutor Mediated simulations and all have the same structure.)

Group Name

When the simulation runs, the first thing that happens is a window opens that asks for a Group File Name (Figure 7.04). The name entered is used to store the simulation data for reloading as the simulation progresses.

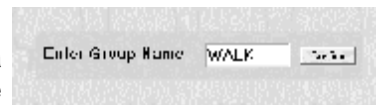


Figure 7.04: Group Name

After entering the Group Name, the simulator searches for the appropriate data file. If this does not exist (and assuming that you have not mistyped the Group Name) you are asked if you are starting the simulation.

Number of Teams

On clicking Yes, you are asked the number of competing teams (Figure 7.05). (For my simulations, the number of teams can range between two and eight. Although, arguably, two is too few to get good competition and more than eight is too dynamic.)



Figure 7.05: Decide Number of Teams

Select Version

Following this, if there is more than versions, I display a menu to choose which to use.

Master Menu – Tutor Mediated Simulations

Now the simulation is set up and the master menu is displayed (Figure 7.06).

Quit Simulator

This finishes the current run of the simulator and when rerun and the same Group Name is entered, the simulator loads the current data allowing you to continue.

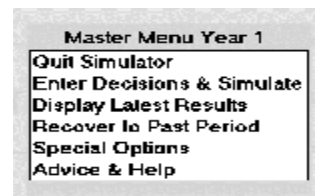


Figure 7.06: Master Menu

Enter Decisions & Simulate

This initiates the normal processing cycle (decision entry, simulation and automatic printing of team and tutor's reports. Initially, an (optional) page can be displayed to remind the novice trainer of possible period issues.

Next the trainer decides printing options (Figure 7.07a). When used in the classroom, the normal choice is **Print after Simulation**. (I.E. After all the calculations have been completed and the data saved.) **Manually choose reports** is useful when demonstrating the simulation or becoming familiar. **Print after Simulation** is used where the simulation is very complex and the bottleneck is the computer rather than the printer.

Next it is possible to (optionally) adjust parameters (like market sizes, costs, interest rates, etc.). For example with DISTRAIN, it allows the tutors to simulate the impact on sales demand of natural disasters (such as a hurricane). When the trainer decides to *Adjust the Economy* a template is displayed allowing *pre-defined* parameters to be changed.

Finally the decisions for the teams are entered (Figure 7.07c). These can be entered in any order and when selected one or more decision entry templates are displayed. After decisions entered for a team this is marked as **<entered>**. Once all decisions are entered the **Simulate Year** is clicked.

Decision Entry for a team

Entering a team's decisions involves deciding a team name (first period only), followed by one or more templates, decision checking and the option to print the decisions.

I find getting teams to name themselves (Figure 7.07d) is motivating rather than using a generic name. But I limit the name to six characters as otherwise printouts are too wide.

One or more decision templates are now provided to enable decision entry. (If there are multiple decision templates, these are selected from a menu. Each decision template is pre-populated as appropriate so the tutor only needs to enter decisions that are changed. When a decision is entered it changes from black to red. Besides moving around using the mouse one can use the arrow keys to move around the template. When all the decisions are entered the end key is pressed or the confirm button clicked.

After confirming entry, the decisions are checked and if there are issues these are displayed. In the example (bottom of Figure 7.07f) the *Advertising merely reallocated* means that the learners may not be thinking deeply enough about the advertising decision and the *Production exceeds Capacity* comments means that the production requested cannot be achieved. The tutor has the opportunity to print these comments and give them to the team.

The final stage of decision entry is to (optionally) print a copy of the decisions (Figure 7.07g) that can be returned with the results to the team. (Printing the decisions to return to the

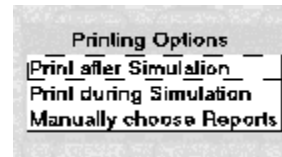


Figure 7.07a: Printing Options

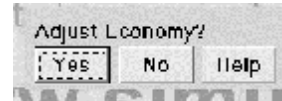


Figure 7.07b: Adjust Economy

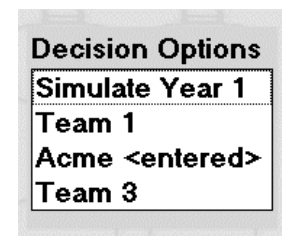


Figure 7.07c: Decision Menu

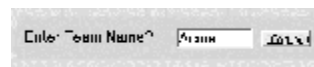


Figure 7.07d: Team Name

	Product A	Product B
Selling Price	88	48
Advertising	200	200
Production	70	210
Sales Force Size	4	
Capacity Increase	80	
Accounts Receivable Days	78	

Figure 7.07e: Decision Template

	Product A	Product B
Selling Price	88	48
Advertising	200	200
Production	70	210
Sales Force Size	4	
Capacity Increase	80	
Accounts Receivable Days	78	

Acme Annual Decision Comments Year 1
Advertising merely reallocated
Production exceeds Capacity

Figure 7.07f: Problem Decisions

results to the team. (Printing the decisions to return to the team is desirable as, commonly, they do not keep a record of their decisions and the tutor needs to keep the decision form that the team submitted to handle queries. It is helpful if the tutor checks the printed decisions against the submitted decisions.

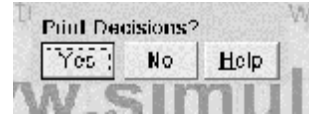


Figure 7.07g: Print Decisions?

Simulation Process and Printing

When ***Simulate Period*** is clicked the decisions are implemented, results calculated, data saved and results printed. As described in Chapter 6 (Design for Value), the results are printed as follows: Preliminary Results, Full Team Results, Business Research, Tutor's Audit and, finally, Team Commentaries

Display Latest Results

This allows the trainer to revisit results, answer questions and explore in depth learning problems and opportunities

Recover to Past Period

This allows the trainer to return to an earlier period to rerun, to correct mistakes in decisions or explore alternate strategies.

Special Options

These are things like reconfiguring the printer, deleting or copying group data or changing team names that are used infrequently and not part of the main processing cycle.

Advice & Help

This allows the novice user to explore the context sensitive help database.

Direct Use Simulator Simulations

Here the simulator is used by the learners (Product Launch, Prospector and SEED) are all Direct Use simulations and have similar structures – both Product Launch and SEED have the same structure but Prospector as a process simulation has a slightly different structure). Besides the

Processing Structure – Direct Use Simulations

This section describes how the simulator processes decisions and produces results. It is the structure used by Product Launch and SEED.

Team Name

Directly after running the simulation, the team is asked for the team name that identifies the file that stores simulation data. The first period that the team name is entered the simulator checks that the team is starting a new simulation run (rather than having typed in the wrong name).

(Budget Entry)

Optionally, depending on learning needs, learners can enter a budget and forecast results at this point

Decision Entry

Next the period's decisions are entered. As for Tutor-Mediated Simulations, decisions are entered using one or more templates. Following confirmation of the decisions they are checked and, as appropriate, comments made.

Simulate

After decisions are confirmed their impacts are assessed, results calculated and data saved.

Results Print and Display

After simulation, the results are displayed and (optionally but usually) printed.

Ending Review

As described earlier, a separate model reviews performance and produces reports after a predetermined number of simulated periods. At that time, the team can optionally, finish with the simulation or be forced to finish.

Tutor's Options – Direct Use Simulations

These are the options that I build in for the trainer. They are *password* protected to prevent learners doing inappropriate things (cheating).

Entering the right text in response to the Team Name question causes the simulator to ask for a password and if the right password is entered a menu of options is displayed (Figure 7.08).

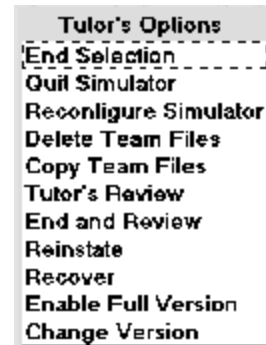


Figure 7.8: Tutor's Menu

Quit Simulator

Stops the current run of the simulator.

Reconfigure Simulator

This allows the printer, paper size, etc. to be changed.

Delete Team Files

This allows team data to be deleted.

Copy Team Files

This allows team data to be copied for transfer to other computers.

Tutor's Review

This allows access to special reports provided for the trainer so he or she can investigate results and answer questions.

End and Review

This allows the trainer to finish with a team's use of the simulation and create the reports that review progress.

Reinstate

This allows the trainer to reverse the *Ending Review* decision.

Recover

This allows the trainer to move back to an earlier period to correct mistakes in decisions or explore alternate strategies.

Enable Full Version

This is a particular feature of my simulations. It allows the trainer to switch from an evaluation version of the simulation to the full version.

Change Versions

This allows the trainer to switch to another version of the simulation.

Data Structures

In the early 1970s data processing espoused the data centric view of system design (Bachman, 1973). Similarly, for simulations, just as *Transactions* are as important as the *Model*, data and its structure is as important as the algorithms and this section discusses the data used to drive the simulation model. This consists of two classes of data – the data that drives the simulation model (parameters) and the data associated with the current business situation (team data). Except for two of my most complex simulations, I have found that I have needed a maximum of three parameter structures and three team data structures (Figure 7.09).

Description	Shape	Detail
Economic Parameters	Vector	Variable
Market or Product Parameters	2 Dimensional Matrix	Market or Product by Variable
Resource Parameters	2 Dimensional Matrix	Resource by Variable
Market or Product Data	3 Dimensional Matrix	Period or Team by Market or Product by Variable
Resource Data	3 Dimensional Matrix	Period or Team by Resource by Variable
Financial/Operating Data	2 Dimensional Matrix	Period or Team by Variable

Figure 7.09: Parameter and Team Data Structures

However, arguably, for simulations where there is interaction between teams, the Market and Resource Team Data should be four dimensional matrices (Period, Team, Market or Resource and Variable) and the Financial Operating Data should be a three dimensional matrix (Period, Team and Variable).

For Modern Banking the data file handles information about two markets (retail and corporate), two classes of staff (junior and senior), core costs, the economy (and how it changes over a twelve year period) and consolidated team results. Parameter data consists of thirty-seven economic parameters (such as depreciation, capital adequacy limits), eighty-two market parameters (such as loan life, market response parameters) and twenty-nine parameters that define how economy will develop over the years. Its team data consists of 146 Market variables (such as current number of customers, revenue and interest decisions), 56 staff variables (such as staff numbers, costs and productivity) and 193 general company variables. Thus (figure 7.10) there are in total 549 parameters and (where eight teams are competing) 4776 team data items – a total of 5325 items.

Description	Array Shape	Total Number
Economic Parameters	List	37 x 1
Market Parameters	Market by Variable	2 x 82 = 164
Annual Parameters	Year by Variable	12 x 29 = 348
Market Data	Team by Market by Variable	8 x 2 x 146 = 2336
Staff Data	Team by Staff by Variable	8 x 2 x 56 = 896
General Data	Team by Variable	8 x 193 = 1544

Figure 7.10: Modern Banking Data Structures

In fact, the actual number of items is higher because when building my data files I always include a significant number of *spare* data records. For instance, the economic parameter vector consists of 51 elements rather than the 37 elements needed. These *spares* allow for additions made during alpha and beta testing and future enhancements.

Variable Structures

Beyond the actual value, each parameter or team data variable needs to record:

- **Variable Name**
- **Variable Format**
- **Help Page Link**

Variable Name identifies the variable and can be used when producing reports.

Variable Format defines the variable time (numeric, textual etc.), is used when producing reports and, if a decision, constrains what can be entered.

Help Page Link defines the help page that should be used as part of the on-line context sensitive help.

Figure 7.11 shows an example of a variable from Modern Banking. Here variable 64 is the Short Term Loan Interest Decision. The variable format (##.##) uses the # as a number *place marker* and constrains the decision to lie between 0 and 99.99. Based on the position of the decimal point, there can be no more than two places of decimals (and,

because the format does not start with a minus, negative interest rates are not allowed. The help page is on help page 224 and the variable values for the two markets are currently both 2.40.

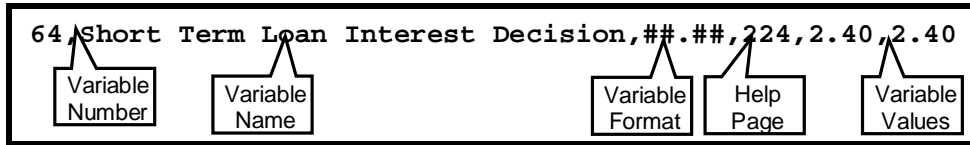


Figure 7.11: Modern Banking Variable Structure

Other Data

As described in Chapter 6, I use files to store control data, textual data, constants, comments, reporting structures and help pages. As these are specific to my platform it is not appropriate to describe them in detail.

Design Process Example

While writing this book I was in the process of updating and extending a complex simulation and this allowed me to revisit the design process and share my ideas about a typical sub-stage in the design. The simulation that I was working on was UMIK – a simulation designed to allow managers to explore the fundamental structure of a business and how money, materials and information flow between these.

When I developed my platform in the late 1990s I provided a development mode designed to allow reports, variables and help pages to be explicitly revealed so that I could immediately identify records in the various databases (figure 7.12a and 12b).

Fabrication Cost Analysis April	
Raw Material Used	0
Direct Fabrication Wages	9600
Fabrication Overheads	10800
Actual Fabrication Cost	20400
Standard Fabrication Cost	9374
Fabrication Cost Variance	11026

Figure 7.12a: Operational Mode

89 Fabrication Cost Analysis April [384]	
153 Raw Material Used [0]	0
102 Direct Fabrication Cost [194]	9600
101 Fabrication Overheads [192]	10800
131 Total Fabrication Cost [391]	20400
151 Standard Fabrication Cost [0]	9374
152 Fabrication Cost Variance [0]	11026

Figure 7.12b: Design Mode

The Fabrication Cost Report was originally to show how the costs of the fabrication unit were derived from raw material use, labour costs and overheads (treating the fabrication as a cost centre). However, while working on it I thought that it would be useful to use it to analyse how actual costs compared with *standard costs* (allowing an analysis of actual against budget (standard cost) and so added the bottom two lines. Figure 7.11a is the way the operational version looked. Figure 7.11b is the way the design version looked after about halfway through adding the analysis of actual against budget.

The design version reveals information about the sources of data - the report record number (the top line to the left of the report #89) in the reporting database.

Next the design version reveals information about the variables used- to the left of each variable is the variable number in the variable database (#153, #102, #101, #131, #151 and #152). These are not in order because the of the creative, iterative design process means that variables cannot be pre-planned but are introduced as the design proceeds. Here, to add in the budgetary analysis, I needed to add three variables (#151, #152 and #153). Initially (as shown in figure 7.11b) I just totalled direct Fabrication Cost and Fabrication Overheads to calculate Total Fabrication Cost but then realised that I had to add in Raw Materials Used – Figure 7.11b shows this at the stage where the Raw Material Used line has been added to the top of the report but the calculation has not been added to the simulation model (and so the raw material use is zero).

In square brackets, to the right of the report and variable names are the page numbers of their help records [#384 etc.]. Here Raw Materials Used, Standard Fabrication Cost and

Fabrication Cost Variance all show #0 indicating that a help page has not yet been created for these variables.

While working on this report I decided that it would be clearer if I changed Direct Fabrication Cost to Direct Fabrication Wages and Total Fabrication Cost to Actual Fabrication Cost.

What happened during this stage of development?

I added the two variables (Standard Fabrication Cost and Fabrication Cost Variance) and built and inserted the models that calculated both.

I then realised that I needed to include the Raw Materials Used variable and needed to create and insert this model and create a report to display these results in the *Tutor's Audit*.

On running the simulation and displaying this report it was apparent that I needed to create help records for the three new variables, enter these in to the help database and link via the variable database. Additionally it would be necessary to check and modify if necessary the report help record (#89) and the Total Fabrication Cost help record (#131).

Next I needed to use Excel to use information from other reports to duplicate and check the calculations.

As I looked at the report, I decided that (in the interests of clarity) I should rename the Direct Fabrication Cost and Total Fabrication Cost and this involved changing the variable name in the variable file and the help record in the help file.

Finally, I replicated the report (Figure 7.11a) in the paper-based Tutor's Manual, copied and reformatted the help records from the help database to the Tutor's Manual and wrote an explanation of the issues and learning addressed in the Tutor's Manual.

Chunking and Refactoring

The work that I have just described illustrates the iterative, refactoring process implicit to building simulation models. It shows how ideas are revisited and how the complete model is built in steps as a series of small, manageable sub-models. In terms of quality this approach is important as it prevents short-term memory being overloaded. Spolsky (2004) explains that programming "*productivity depends on being able to juggle a lot of little detail in short-term memory all at one time*". There is considerable research into the capacity limit of short-term memory (Miller, 1956). So building the full model in as many small sub-models improves both quality (Deacon, 2004) and productivity.

Additionally, I have found that revisiting a sub-model (refactoring) allows one to use the subconscious to continue considering it and improving it. But this necessitates documenting the model as you build it (using comments in the code and the help database).

Modelling Language

Just as for painting the choice of medium is controversial, the choice of modelling language can be a controversial issue. There are several choices – high-level languages (such as BASIC or C) or spreadsheets (such as Excel™).

Painting	Simulation
Oil	C
Acrylic	BASIC
Water Colour	Flash & Java
Crayon	Spreadsheet

Cartoon 7.01: Media

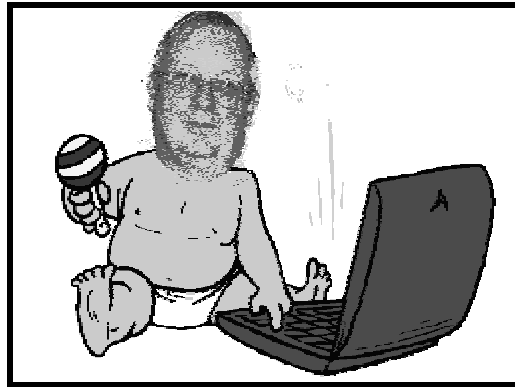
If you have knowledge of painting then Cartoon 7.1: Media resonates. I develop my simulations using BASIC. There are several reasons for this.

1. There is the longevity of the language. When I moved to microcomputers (from Computer Timesharing) in 1980, I had several simulations that I wished to transfer. Happily they were written in BASIC and so the transfer took very little time. By the late

1990s when I moved to Windows, I had some three-dozen simulations to transfer from MSDOS. For these, all that was needed was to copy the simulation models across and update - typically this took half a day.

2. There is the appropriateness of the computer language to modelling. BASIC was derived partly from FORTRAN (FORMula TRANslation) and so is very suited to building mathematical models.

3. Having used BASIC for over forty years I am totally fluent. (As Cartoon 7.2 shows, I was very, very, very young when I started and leaning out of the pram or baby buggy to type was a problem.) As a result I can model in BASIC very, very, very quickly.



Cartoon 7.02: My early days

This is not to say that I have not used other languages to build simulation models. In the early days I used both ALGOL and FORTRAN (one of which has disappeared and the other has virtually disappeared).

4. In the chapter on Design for Quality (Chapter 8) I explain why *spreadsheets* are inappropriate and quality assurance is better with a high-level language.

The purists will argue that BASIC is unstructured and this is true if you allow it to be so. But so too is the English Language. English does not have the rigorous structure of say French. Yet of any western language, English has the greatest communication (Baud) rate and, so it is remarkably efficient at communication, as is BASIC when it comes to the rapid creation of business simulation models. (This is because model creation is an iterative, recursive process.)

Flavour of the Month

In the late 1990s I spoke to the UK's University for Industry (a government sponsored and funded initiative) about developing a simulation to be used to promote the University for Industry to business. At the meeting I was informed that it was policy for **all** software to be developed using Java! Such a move would mean that I could not use my existing platform and, in any case, Java is not a good *modelling* language. But, obviously to some politician somewhere, Java was the flavour of the month even though it would probably increase simulation development costs fifty-fold. Although, I guess it was better than trying to build a website using COBOL.

Language Elements

A high level language such as Visual Basic has a huge range of statement types. But, if you are only concerned with developing the simulation model part of the simulator you only need a small subset (Figure 7.13).

I go through these language elements detail because if the SME's understand them they should be able to use them to define, explain and refine their model needs.

- 1) Assignment
- 2) Conditionals
 - a) If Then
 - b) Select Case
- 3) Loops
 - a) For Next
 - b) Do Loop

Figure 7.13: Language Sub-set

1. Assignment

This is used to assign a variable from data, using an expression or one variable to another (figure 7.14).

TotalMarket = 100	'assigns the value of 100 to the variable
SalesDemand = MarketShare*TotalMarket	'assigning from an expression
ActualSales = SalesDemand	'assigning one variable to another

Figure 7.14: Examples of assignment

Although as shown it is possible to assign a number in your model, I avoid doing this and hold all numbers in the database rather than embedding them in the model. This allows me to update parameters and data without having to search through the model and hope that I find all the numeric assignments. Another issues associated with assignments are discussed in the *Quality Assurance Issues* section.

2. Conditional Statements

These allow different calculations to occur depending on the value of variables.

a. If...Then

This is the most commonly used conditional statement and there are three forms:

- If...Then (Figure 7.15a)
- If...Then...Else (Figure 7.15b)
- If...Then...Elseif (Figure 7.15c)

ActualSales = SalesDemand	'assign demand to actual sales
If SalesDemand > AvailableInventory Then	'not enough inventory
ActualSales = AvailableInventory	'actual sales limited to inventory
End If	

Figure 7.15a: Example of If...Then structure

If SalesDemand > AvailableInventory Then	'not enough inventory
ActualSales = AvailableInventory	'actual sales limited
Else	'sufficient inventory
ActualSales = SalesDemand	'assign demand to actual sales
End If	

Figure 7.15b: Example of If...Then...Else structure

If Production > OvertimeCapacity Then	'not enough inventory
ActualProduction = OvertimeCapacity	'actual production limited
Overtime = ActualProduction – Capacity	'amount produced in over time
Elseif Production > Capacity Then	'need to work overtime
ActualProduction = Production	'actual production = decision
Overtime = ActualProduction – Capacity	'amount produced in over time
Else	'sufficient inventory
ActualProduction = Production	'actual production = decision
Overtime = 0	'no overtime
End If	

Figure 7.15c: Example of If ...Then...Elseif structure

Figures 15a and 15b are logically the same but figure 7.15b is clearer. Figure 7.15c shows how a production decision is modified to take into account capacity and overtime.

b. Select Case

This is particularly useful if a variable or expression is used to decide which of several different actions need to be taken (figure 7.16)

```

Select Case SalesDemand / AvailableInventory 'sales demand relative to inventory
  Case > 1.2 'demand well above inventory
    Comment = "Significant sales lost"
  Case > 1.0 'demand above inventory
    Comment = "Some sales lost"
  Case < 0.8 'demand well below inventory
    Comment = "Significant excess inventory"
  Case < 0.95 'demand below inventory
    Comment = "Some excess inventory"
  Case Else 'demand and inventory balanced
    Comment = "Inventory about right"
End Select

```

Figure 7.16: Example of Select Case statement

For clarity in figure 7.16 the checks are shown as actual numbers and the comments are shown as actual text. In practice both would be held in a data file and would be assigned to variable that would be used in the model. An alternative to the Select Case statement is to use multiple Elseif...then sections in an If...Then structure. But, for multiple conditions, I find the Select Case structure is preferable as it is clearer.

3. Loops

These allow blocks of calculations to be repeated several times.

a. For...Next

These are used when the calculation will be repeated a set number of times (as might be the case when there are several market sectors or resources).

```

For market = 1 to NumberofMarkets 'loop through all markets
  'individual market calculation here
Next Market 'end of repeated calculations

```

Figure 7.17: Examples of For...Next statements

b. Do...Loop

This also allows a calculation to be repeated several times but only while a condition is true. In the Prospector simulation learners searched for project opportunities that met criteria that they set. Once a project was found that met all the criteria then the search was over.

```

Do 'start of loop
  ProjectType = 1 + 2 * Rnd 'select project type randomly
  SearchTime = SearchTime + 1 'accumulate search cost
Loop Until ProjectType = RequiredType 'end of repeated calculations

```

Figure 7.18: Examples of Do...Loop statements

In the example, the do loop is used to randomly generate a project type that matches a predefined project type (RequiredType). While a suitable project is being searched for, the time taken to do the search is being incremented.

c. While...Wend

This is similar to the Do...Loop statement.

Variable Types, Operators and Functions

1. Variable Types

A high level language such as Visual Basic has several variable types. Typically I make use of two types of numbers - integers (whole numbers) and single (decimal number) and, on occasion, I use long integers and strings (to store text).

Number Type		
VB.net	VB6	Range (Microsoft® Visual Basic.net)
Short	Integer	-32,768 to 32,767
Single	Single	-3.402823E38 to -1.401298E-45 (negative numbers) 1.401298E-45 to 3.402823E38 (positive numbers)
Integer	Long	-2,147,483,648 to 2,147,483,647
Strings	String	Used to store text (alphanumeric characters)

Figure 7.19: Variable Types

I find **short** (whole numbers) useful for things like resource numbers, time, small quantities of inventory and staff numbers. I find **single precision numbers** suitable for most other variables. I use **integers** to store a list of flags and **strings** to store text information.

When developing SEED, I needed to keep a record of whether or not learners had done some market research or not and used integers to store a list of these flags (Figure 7.20).

Flag	Market Research
1	Market Potential Research
2	Selling for Early Learning
4	Selling as Companions
262144	Staff Cost Research
524288	Manufacturing Cost Research
1048576	Promotion Cost Research

Figure 7.20: Market Research Flags

By making the flags increase in powers of 2 it is possible to test for them using the AND operator (see Figure 7.26 later).

I use **strings** to store text such as comments that need to be fed back to learners. (This is illustrated earlier in figure 7.16 (the Select Case example).)

2. Arithmetic Operators

These do the calculations.

Operator	Description	Example where A=5 & B=2
+	Sum two variables	A + B (result = 7)
-	Subtract two variables	A - B (result = 3)
/	Divide one variable by another	A / B (result = 2.5)
*	Multiply two variables	A * B (result = 10)
\	Integer division	A \ B (result =2)
^	Exponentiation	A ^ B (result = 25)
Mod	Modulus Arithmetic	A Mod B (result = 1)

Figure 7.21: Arithmetic operators

The first four operators need no real explanation. However, the last three do.

Integer Division (\)

This returns an integer when a two numbers are divided. I use it rarely but did use it in the SEED simulation where it was only possible for retailers to order full cases of the Cuddl-Etoy (Figure 7.22)

```
NumberofCases = PurchaseNeed\CaseSize 'get cases purchased
AmountPurchased = NumberofCases*CaseSize 'get actual purchase amount
```

Figure 7.22: Examples of Integer Division

Exponentiation (raising a number to a power)

This is useful in statistical and D.C.F. calculations and to create a non-linear response to a factor. I used this function in a simulation where I wanted to forecast perceived skills from actual skills. I could have used the Rnd function but felt that it would be better to assume that at low actual skills levels the perceived knowledge would be less than the actual. As the actual skills increased the perceived skills would move closer to the actual. I used the exponentiation function to link the two with a factor less than one. This meant that as the actual moved closer to the ultimate the forecast (perceived) skills became more accurate.

Modular Arithmetic (the Mod function)

This calculates the integer remainder when dividing two integers (Figure 7.23)

At first glance you may ask “Where might I use such a function?” I use it regularly where the simulation period is a quarter (or a month) to initialise year start data (Figure 7.24).

1 Mod 4 = 1
2 Mod 4 = 2
3 Mod 4 = 3
4 Mod 4 = 0
5 Mod 4 = 1
6 Mod 4 = 2

Figure 7.23: Modulus

```
Quarter = Period Mod 4 'convert period being simulated into quarter
If Quarter = 1 Then 'first quarter in the year
    'year to date variables initialised here
End If 'end of initialisation routine
```

Figure 7.24: Examples of use of Mod function

3. Logical Operators

These do the logical tests and calculations. Figure 7.25 is not a complete list of logical operators, rather it shows the ones that I use regularly.

Operator	Description	Where A=5 & B=2
=	Checks for equality	A = B (result = false)
<	Checks for first variable smallest	A < B (result = false)
>	Checks for first variable largest	A > B (result = true)
<=	Checks for first variable smallest or equal	A <= B (result = false)
>=	Checks for first variable largest or equal	A >= B (result = true)
<>	Checks to see if different.	A <> B (result = true)
And	Checks to see <i>both</i> are true	A And B (result = 0)
Or	Checks to see <i>either</i> is true	A Or B (result = 7)

Figure 7.25: Logical operators

The **And** operator is very useful to check to see if a flag has been set in a group of flags. For example the SEED simulation checks to see if market potential research has been done (Figure 7.26). If so the market size is known but, if the research has not been done, then the market size for planning purposes is set to zero and a message generated advising the learners of this.

```
If (ResearchFlag And 1) > 0 then      'check to see if research done
    MarketSize = MarketPotential      'assign potential market size
Else                                  'no research done
    MarketSize = 0                    'zero potential market size
    Message = "No knowledge of potential" 'warn of problem
End if                                'end of repeated calculations
```

Figure 7.26: Logic using And to check for market potential research

4. Functions

High-level language and spreadsheets have a wide range of standard functions. However, I only use of few of these (Figure 7.27), plus several that I have created (Figure

7.28). In VB6 these are functions but in VB.Net many of the standard functions are Math Methods.

Standard Functions	Description
Rnd	Generates a random number between 0 and 1
Int	Truncates to an integer
Sin	Sine of an angle
Sign	Sign of a number
Abs	Absolute value of the number
Val	Converts a string into a number
Str	Converts a number to text (a string)

Figure 7.27: Standard Functions

Special Functions	Description
Max	Selects the maximum of a pair of numbers
Min	Selects the minimum of a pair of numbers
Log10	Logarithm base 10
Swap	Swap two numbers or strings
NormalRandom	Generates a normal random number

Figure 7.28: Special Functions

The **Rnd** function generates a random number from a uniform distribution between 0 and 1. It is a key function for any stochastic simulation. However, there are several issues – the way the number is generated, debugging a stochastic simulation and generating random numbers from other distributions.

The computer generates a random number based on a *seed* that is normally the previous random number. This means that actually, the random number is not truly random (although, in practice it is sufficiently random to be used). However, this means that it is possible to generate the *same sequence* of random by starting the sequence with the same *seed* (Figure 7.29a) or a different sequence of random numbers each time the simulation is run (Figure 7.29b). (Note: Rnd with a negative argument is necessary to ensure that the random sequence is repeated.)

Seed = 100	'initialises a predefined seed
X = Rnd(-1)	'see note
Randomize Seed	'initialises random number generator

Figure 7.29a: Generating the *same* random sequence.

Seed = Timer	'initialises to the current time
Randomize Seed	'initialises random number generator

Figure 7.29b: Generating a *different series* of random numbers

It is useful to generate the same sequence of random numbers when debugging the simulation model as this means that the results are unaffected by changes in the sequence of random numbers. This is also useful where you want to compare teams. This was the situation for my TEAMSkill simulation. This simulated the operation of a factory and was used as a national contest in the UK. To ensure fairness each machine and day's operation selected the same random number.

The standard Rnd function produces a random number from a uniform distribution. However, you often require a different distribution. One example of this (for the Prospector simulation) is the distribution of project size. Here I wanted project sizes to be generated randomly with a few large projects, several middle sized projects and many

small projects. To do this I took the random number, raised it to a power and multiplied it to get the project size (Figure 7.30)

```

MinProject = 100           'smallest project is 100
ProjectSpread = 1900      'largest project is 2000 (1900+100)
SkewFactor = 2.5         'skew distribution towards 0
ProjectSize = ProjectSpread * (Rnd ^ SkewFactor) 'get random part of project size
ProjectSize = MinProject + ProjectSize          'add in minimum size

```

Figure 7.30: Generating project sizes.

Besides the uniform distribution, a second important distribution is the Normal distribution. Because of the importance of this distribution, I have created my own function (NormalRandom) that generates a normal random number based on a desired standard deviation.

The second function that I use regularly is the Int function. This is useful to convert a decimal number to an integer (whole number) or to convert a number to a fixed number of decimal places (Figure 7.31)

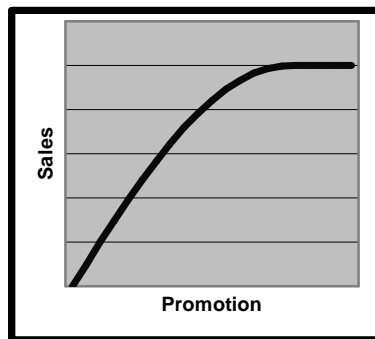
```

Sales=Int(demand + .5)           'round to nearest number
Stock=Int(Production)           'round to whole number below
Income=Int(Price * Sales * 100 + .5) / 100 'round to two decimal places

```

Figure 7.31: Using the Int function to round numbers.

The third function that I use regularly is the Sin function (a math method in VB.Net). This may seem odd except that it provides a neat way to represent a saturating relationship. (Figure 7.32)



$$\text{Response} = \text{Sin}(\text{Min}(1.57, \text{Factor})) \quad \text{'limit factor to 1.57}$$

Figure 7.32: Using the Sin function to make a saturating response

As the Sine curve will drop after the factor passes 1.57 ($\pi/2$) the value of the factor is constrained to 1.57. In the SMITE simulation I used the Sine curve to generate the impact of coaching on skills improvement.

The **Sign** function allows on to determine whether a number is positive, negative or zero. If a number is positive, Sign returns +1, if negative -1 and if zero it returns zero. For example in SEED the learners could lease a factory or use outworkers. If the factory was leased then there was a banking cost associated with paying for the factory. So the model checked to see if a factory lease payment was made and if so a bank charge was incurred.

```

BankCost = BankCost + Sign(FactoryCost) * PaymentCost

```

Figure 7.33: Using the Sgn function.

(I could have done the same thing using an If Then statement but here there were several other things that could be leased and his meant the one statement could be used to calculate BankCost rather than many.)

The **Abs** function always returns a positive number and so can be used to convert a negative number into a positive one while leaving positive numbers as positive.

Max and **Min** functions are ones that I used regularly when I used PowerBASIC to develop MSDOS simulations and although VB6 has a function that could be used (IIf function), I found this unwieldy and so built two separate functions. But VB.Net now has Max and Min maths methods!

The standard Logarithm in Visual Basic is a natural logarithm (to base e). However, the base 10 logarithm is more useful as it provides a way of determining how big a number is and because of this I created a special **Log10** function (Figure 7.35). (Note: the MSDN website has an example of how to program a Log10 function. However, this does not check for zero or negative numbers and hence will cause an error if the number (X) is zero or negative! And, although VB6 did not have a Log10 function VB.Net now has!)

```
Function Log10#(ByVal X)
  If X > 0 Then           'only calculate for positive values
    Log10# = Log(X) / Log(10#) 'divide natural logarithms
  Else                   'negative or zero
    Log10# = 0           'set to zero (if negative)
  End If
End Function
```

Figure 7.34: Function to calculate Log base 10

Documentation

This section explores the documentation needed when using a business simulation and discusses how this should be created as the simulation is designed. Throughout this section, whenever a document type is shown in brackets this means that it is optional and may not be needed for your simulation. Documentation when the simulation is used falls into these classes:

- Learner Documentation**
- Online Documentation**
- Tutoring Documentation**

And documentation used when developing a simulation involves:

- In Software Documentation**
- Model Documentation**
- Quality Assurance Documentation**
- Calibration Documentation**

Learner Documentation

This comprises the paper-based information provided to learners thus:

- Learners' Manual**
- Briefing Slides**
- (Decision Forms)**
- (Worksheets)**
- (New Information)**
- (Additional Tasks)**
- (Definitions)**
- (Business Theory)**

Learners' Manual

This consists of the information provided to the learners and is likely to consist of descriptions of:

Introduction
The Business
The Decisions
The Results
(The Business History)

For adult learners, the learners' manual should be short. Experience suggests that even the shortest, most succinct manuals, are not fully read or understood (and you have no chance with a long manual). Further, if you provide a manual beforehand to be read before the event, a few will read it and remember, a few will read it and forget about it and the vast majority will not read it.

The manuals of the case study simulations range in length from two to eighteen (A4 or Letter size) pages. This compares with simulations used on academic programs (Keys and Biggs, 1990) where a review of nine simulations shows a page count of between 166 and 256 pages. However, included in Keys and Biggs' page count were pages devoted to theory.

Where the learners are making direct use of the simulator it is better to provide context sensitive help. Where the simulator is used by the trainer you can often rely on she or he answering team questions. Figure 7.35 shows the lengths of the learner manuals for the case study simulations.

Further, there is the degree of ambiguity. Instructions (rules) should be totally unambiguous. However, the description of the business that is to be run should be appropriately ambiguous. Thus some information will be unambiguous and other will be ambiguous. Often, information about cost and resources will be unambiguous but information about the marketplace and economy would be ambiguous. However, this is not universally true. For Product Launch the total market size is stated

Simulation	Product Launch	Modern Banking	DISTRAIN	SMITE	SEED	Prospector
Manual Length	2 pages	12 pages	8 pages	18 pages	18 pages	18 pages

Figure 7.35: Learner Manual lengths in A4/Letter size pages

(Although the DISTRAIN manual is only eight pages, additional decisions are introduced as the simulation progresses and each of these add pages.)

Introduction

This briefly describes the simulation and the task facing the learners. Figure 7.36 shows the introduction for the Product Launch simulation. Typically the length of the introduction ranges from a few lines (Product Launch) to a single page (DISTRAIN, Modern Banking).

This simulation involves you introducing a new consumer product - the *SOUPER HOT* range of self-heating soups. These are soups in a wide-necked container. On opening, you pull a tab and a moment later the soup is hot and ready to drink. *SOUPER HOT* can be used by the busy cook, by the sportsman, on picnics - anywhere.

You launch *SOUPER HOT* making decisions about price, promotion and how much to produce. You enter these in to a computer that simulates their effect and produces results for you to analyse before making more decisions. You should do this for at least eight periods.

Figure 7.36: Product Launch Introduction

The Business

This describes the business that the learners are to manage. Figure 7.37 illustrates this information for the Product Launch simulation.

Initially, SOUPER HOT is unique and you have no competitors. However, the technology of the heating device (the HOTZIT) cannot be patented and so, eventually, competitors will launch similar products. So, it is important to get established in the market before the, computer simulated, competitors arrive.

Initial market research suggests that the total market for **SOUPER HOT** is one million units each period. Unfortunately these potential consumers are totally unaware of **SOUPER HOT** and must be made aware of it.

FINANCE

To fund the launch of SOUPER HOT you have £400,000 available. This cash must be used to fund initial losses, investment in factory machinery and stocks. If you spend more than the £400,000 you may go bankrupt!

Figure 7.37: Information about the Business (Product Launch)

The length of this section ranges from a few paragraphs to several pages. As appropriate, it should consist of information about business structure, products, markets, resources, costs and assets etc. Where appropriate this section coupled with it's business history, should be sufficient to allow learners to do a SWOT (Strengths, Weaknesses, Opportunities and Threats) Analysis.

The Decisions

This describes the decisions that the learners are to make. Figure 7.38 illustrates this information for the Product Launch simulation.

DECISIONS

To influence your performance in the market you must make the following decisions:

PRICE
PROMOTION
PRODUCTION

Each period you must enter these decisions into the microcomputer. The Price may be entered to two decimal places. Promotion and Production must be entered as whole numbers.

Figure 7.38: Product Launch Decisions

The Results

This describes the reports produced by the simulation and, if appropriate, how they are grouped. Additionally, you may wish to include definitions for the terms where the learners would not have the requisite knowledge. This was the case for SEED, Product Launch and Prospector. Both Modern Banking and DISTRAIN relied on the trainer defining terms when asked. However, although SMITE is a Tutor-Mediated simulation, it's complexity meant that the help database provided definitions.

(Business History)

If appropriate, this provides information about the way the business has performed in the past. Modern Banking, DISTRAIN and SMITE had business history (three years for Modern Banking, four quarters for DISTRAIN and a single quarter for SMITE). Business History was not a appropriate for Product Launch, SEED and Prospector. Except that for Product Launch a "market test" forecast was provided (Figure 7.39).

TEST MARKET RESULTS	
To help with your initial decisions the following decisions have been <i>tested</i>	
PRICE	1.9
PROMOTION	75000
PRODUCTION	70000
and would produce the following results	
PENETRATION	5
MARKET SHARE	100
UNIT SALES	50000
INVENTORY (UNITS)	10000
SALES INCOME	95000
NET PROFIT	-30000
FIXED ASSETS	140000
INVENTORY (VALUE)	5000
CASH	225000
CUM. PROFIT	-30000

Figure 7.39: Product Launch Business History

Briefing Slides

These are based on the Learners' Manual but very abbreviated. Figure 7.40 shows the number of slides in the briefing.

Simulation	Product Launch	Modern Banking	DISTRAIN	SMITE	SEED	Prospector
Number of Slides	9	17	15	23	27	22

Figure 7.40: Number of Briefing Slides

(Decision Forms)

These are required where team decisions are provided to the trainer for entry into the simulator (as was the case with Modern Banking and Distrain). Where the simulation is used directly by the learners then decision forms are not required. (Although, it may be useful to provide a record sheet.) Finally, decision forms are not required when the decisions are sent electronically to the trainer from a decision support program. Figure 7.41 shows the Modern Banking Decision Form submitted by teams to the trainer for entry into the simulation.

Modern Banking - Decision Form

Team Name		Year	
MARKET DECISIONS	Retail	Corporate	Total
Short Term Loan Interest			
Long Term Loan Interest			
Deposit Account Interest			
Promotion			
STAFF DECISIONS	Junior	Senior	Total
Operating Staff Numbers			
Quality Improvement			
Productivity Improvement			
OTHER DECISIONS			
Number of Account Managers			
Bank Borrowing			

Notes:

Please complete the form carefully and legibly.

Please deliver to the control centre on or before the time specified.

The team name can be up to six characters and cannot be changed after the first year.

Except for the Interest Rate decisions, all decisions must be whole numbers.

You take over the company at year 1.

The Interest Rates are relative to the National Bank's Base Rate and can be entered to two decimal places.

Promotion is the total amount to spend in the year in thousands of EUROS.

Total Staff is the total number of staff to employ for the year separately for the two types of staff.

Quality and Productivity Improvement are the total amounts to spend in thousands of EUROS.

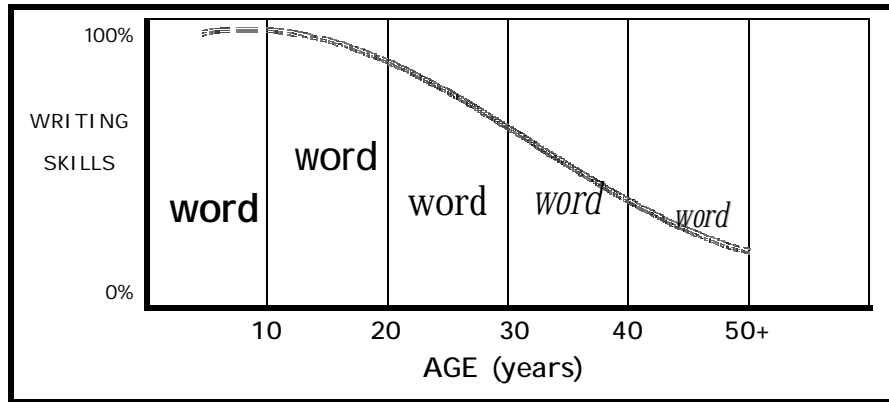
Account Managers are the total number to employ for the year to support to the corporate market.

Bank Borrowing is the amount in thousands of EUROS you wish to borrow from other banks to ensure adequate liquidity. If you wish to lend money to other banks, you should enter a negative number.

The total columns are provided as a check on your decisions.

Figure 7.41: Sample Decision form

The *Total* columns are provided to help interpret badly written decisions. At the bottom of the form are some notes explaining the entries. (Although, generally, learners do not read these notes and will find ways of entering decisions incorrectly!



Cartoon 7.03: Life-Long Writing Skills

(Although not based on rigorous, academic research I believe (as illustrated in Cartoon 7.3) that writing skills peak at about the age of nine. They then start declining slowly until the late teens and then the rate of decline accelerates so that by the time a person is fifty years old, his or her writing is unreadable (when filling out a decision form).)

(Worksheets)

These may be useful to record information or to support manual calculations. Where a worksheet is used it is useful to replicate it with reports produced by the simulation. An alternative is to allow (encourage) learners to use Spreadsheets on their own microcomputers. However, this tends to add time to the decision-making process and may mean that one or more of each team’s learners becomes engrossed with his computer and takes no real part in the simulation (and not learn).

It can be useful to print worksheets and record forms on A3 (Tabloid or ANSI B, USA) or even A2 (ANSI C, USA) sized paper so that they can be fixed to the breakout room wall to share information and be laminated for reuse and updating.

Figure 7.42 shows one of several worksheets for the DISTRAIN simulation to allow teams manually produce the basic accounting reports to allow learners to check their calculations.

Counter Merchandise – Worksheet					Team			
Counter Merchandise - Inventory Movement								
Year	Year 0				Year 1			
Quarter	1	2	3	4	1	2	3	4
Opening Inventory	1000	4167	4267	4127				
+ Inventory Purchase	8000	5000	5000	7000				
= Available Inventory	9000	9167	9267	11174				
– Cost of Goods Sold	4833	4900	5093	5697				
– Obsolescence	0	0	0	0				
= Closing Inventory	4167	4267	4127	5477				
Counter Merchandise - Sales								
Year	Year 0				Year 1			
Quarter	1	2	3	4	1	2	3	4
Cost of Goods Sold	4833	4900	5093	5697				
x Markup	35	36	36	36				
= Gross Profit	1691	1764	1833	2050				
+ Cost of Goods Sold	4833	4900	5093	5697				
= Sales Revenue	6524	6664	6926	7747				

Figure 7.42: DISTRAIN Worksheet

For each line of the worksheet shows the calculation to perform. So *Available Inventory* is calculated by adding *Opening Inventory* to *Inventory Purchase*. Then *Cost of Goods Sold*

and *Obsolescence* are subtracted from *Available Inventory* to calculate *Closing Inventory*. Figure 7.43 shows a record sheet for Modern Banking

Modern Banking - Decision Record						Team	
Retail Market Decisions	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7
Short Term Loan Interest							
Long Term Loan Interest							
Deposit Account Interest							
Promotion							
Corporate Market Decisions	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7
Short Term Loan Interest							
Long Term Loan Interest							
Deposit Account Interest							
Promotion							
Junior Staff Decisions	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7
Operating Staff Numbers							
Quality Improvement							
Productivity Improvement							
Senior Staff Decisions	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7
Operating Staff Numbers							
Quality Improvement							
Productivity Improvement							
Other Decisions	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7
Account Managers							
Bank Borrowing							

Figure 7.43: Modern Banking – Decision Record

(New Information)

These documents are provided when the complexity of the simulation evolves over time. This was the case for DISTRAIN where new decisions were introduced as the simulation progressed and required additional handouts for the later periods (Figure 7.43).

Period	Description of New Information
Quarter 2	Staffing Decision and Reports
Quarter 3	Staff Training and Key Results Report
Quarter 4	Receivable Days and Product Range Decisions
Quarter 5	Marketing Initiatives Decisions

Figure 7.44: Example of New Information

(Additional Tasks)

These are provided when it is desired to add additional tasks. Examples of these are showed in figure 7.45.

Make Board Presentation
Produce Key Result Forecasts
Attend Management Briefing
Prepare Business Plan
Report on Organisational Structure

Figure 7.45: Example of Additional Tasks

(Definitions)

These are provided when it is felt that the learners have insufficient business knowledge. So they are appropriate for junior staff and university students if the information has not been provided as part of prior learning. However, if they are provided you may wish to provide them as a separate booklet or print on different coloured paper. This will mean that the learners know that they only need to access this information on a need to know basis. If the simulation is a Direct Use simulation, it is better to provide definitions as part of the on-line help database.

(Business Theory)

This consists of one or more white papers that discuss business thinking. For example my Financial Analysis simulation had a separate booklet that was provided that discussed financial measures of business performance (Figure 7.46 shows the introduction to this booklet).

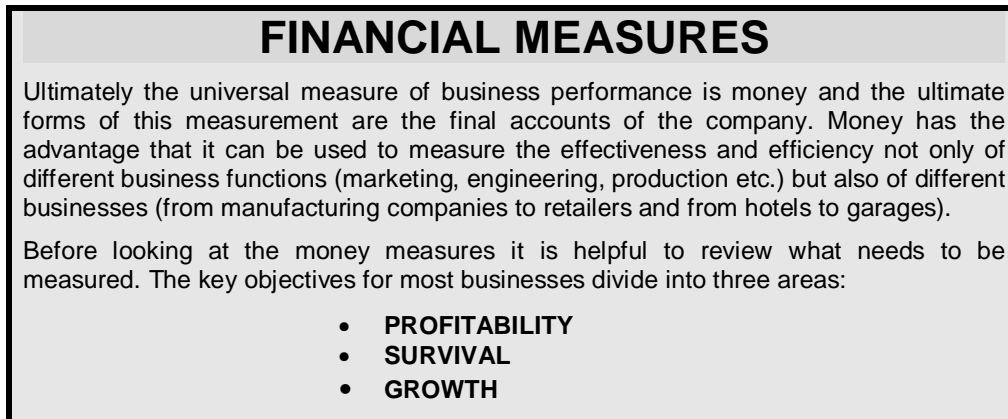


Figure 7.46: Introduction to booklet discussing financial measures

For SEED where there was a similar need, but here business theory was provided as part of the on-line help.

Online Documentation

This consists of a hypertext context-sensitive database consisting of the following information:

Software Help
Task/Report Help
Explanations/Definitions
(Manual)

In each of the examples that follow, underlined text represents a hypertext link to another help screen. The amount of online documentation varies considerably between simulations depending on who uses the simulation (learners or trainer) and their experience and knowledge and the complexity of the simulation. Both DISTRAIN and Modern Banking are similar in terms of complexity but have significantly different numbers of Help Screens. As DISTRAIN was to be facilitated by experienced, trained tutors, the need for on-line help was minimal. In contrast, I did not know who the ultimate trainers would be for Modern Banking. Because of this I provided many more help screens for Modern Banking. Likewise, Product Launch and SEED are similar in terms of user but differ in terms of complexity and this is reflected in their number of help screens.

Simulation	Product Launch	Modern Banking	DISTRAIN	SMITE	SEED	Prospector
Number of Help Screens	254	554	166	497	668	261

Figure 7.47: Help Database Size.

Software Help

This provides information about how to use the simulator (which key to press etc.). Figure 7.48 shows an example of software help.

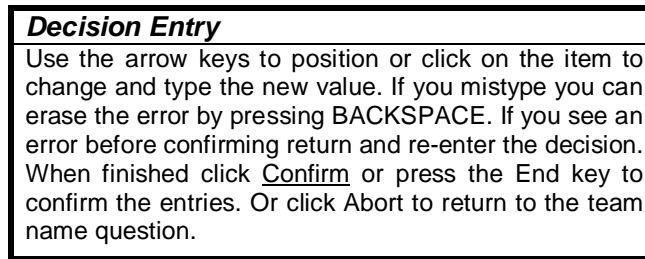


Figure 7.48: Example of Software Help

Task/Report Help

This provides help with the current task or report being displayed.

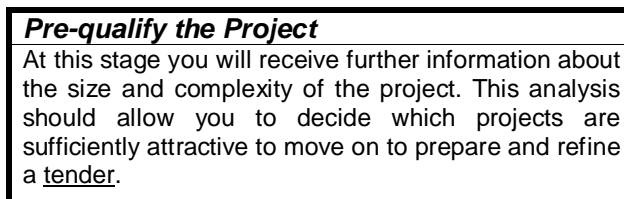


Figure 7.49: Example of Task Help

Explanations/Definitions

This provides information about an individual decision or result. Figure 7.50 shows an explanation of the tender price decision from the Prospector simulation.

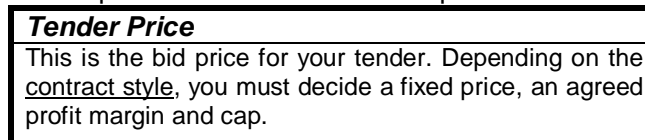


Figure 7.50: Example explaining a decision

Manual

This replicates the learner manual on-line except that it is structurally different from a printed manual as it should have a tree structure and make extensive use of hypertext links. The online manual is optional and is not required where the simulation is used by the trainer or is simple. (None of the Case Study simulations have an Online Manual.)

As I create a simulation and introduce new parameters and reports, I build the online Task/Report Help and the Explanations/Definitions in parallel.

Tutoring Documentation

This comprises the paper-based information provided to the trainer thus:

Background to the Simulation
Using Simulations for Training
Using the Software (Simulator)

Background to the Simulation

This provides simulation specific information and as such may include:

Description of the Simulation
Specimen Timetables
Simulation Models
Decisions and Issues
Description of Reports
Run Statistics

The detail in this document depends both on its purpose and on the experience and knowledge of the trainers. For example, the background notes for Product Launch were designed both to explain the simulation to the trainer and be used as a handout for the learners (after they have taken part in the simulation). Additionally, the Product Launch background notes also included topics for further discussion about marketing. Except for SEED (where currently there is no background document), all the other background notes were designed for use by the trainer. Modern Banking and DISTRRAIN are of similar complexity the difference is because Modern Banking makes use of on-line support and DISTRRAIN does not. As for Learner's Manuals, I believe that Background Notes should be short.

Simulation	Product Launch	Modern Banking	Distrain	SMITE	SEED	Prospector
Number of Pages	9	18	30	48	N/A	30

Figure 7.51: Number of A4/Letter size pages in Background Notes

Description of the simulation

This provides a short introduction, information about key learning, target audiences, durations and manner of use. Figure 7.52 shows the introduction to the Modern Banking trainers' manual

DESCRIPTION

This simulation was developed for the Academy of Banking and Finance of the National Bank of Serbia as part of a project supported by the European Agency for Reconstruction. It provides learners with the opportunity to manage a complete commercial bank making decisions covering funding, marketing, business development, staffing, quality and productivity improvement.

KEY LEARNING

As described later, the simulation covers the following key areas:

**BANKING APPRECIATION
BANKING OBJECTIVES & MEASURES
FINANCIAL APPRECIATION
CLIENT MANAGEMENT & POLICY
PRODUCT MIX & CONTRIBUTION
MARKETING - PRICING & PROMOTION
FORECASTING & CONTROL
BUSINESS IMPROVEMENT POLICY & IMPACT
TEAM WORKING
(BUSINESS PRESENTATION)**

DURATION

Approximately one day. (Although it can be run in an accelerated manner in just over half a day.) Its duration makes it suitable for use at the end of a training course, in stages over the course, as a one-day seminar or as a feature of a conference.

TARGET AUDIENCE

The simulation is designed for use by trainees, junior management, functional specialists and supervisors on banking appreciation courses. (If used in an accelerated manner, it is suitable for more senior management.)

METHOD

After a short briefing, the training group is divided into between two and eight teams of between three and five learners. These teams consider the problem facing them and then make decisions for a trading year. These decisions are fed into a microcomputer that simulates their effect. The results are returned to the teams who analyse them before making the next decision. This decision-making cycle is repeated for six to eight periods before the group recombines to discuss and compare results and justify their performance.

HARDWARE REQUIREMENTS

A single PC and printer for each group of two to eight teams.

Figure 7.52: Description of Modern Banking

Specimen Timetables

These provide information about timetables where the simulation is run in a single block of time, in an abbreviated manner or spread over a course. Figure 7.53 shows the timetable where Modern Banking is used in a single session.

STANDARD TIMETABLE	
00:00 - 00:15	Briefing
00:15 - 02:00	Preparation
02:00	Submit decisions for Year 1
03:00	Submit decisions for Year 2
03:45	Submit decisions for Year 3
04:30	Submit decisions for Year 4
05:15	Submit decisions for Year 5
06:00	Submit decisions for Year 6
06:45	Submit decisions for Year 7
07:30	Submit decisions for Year 8 and prepare for review
08:00 - 09:00	Review

Figure 7.53: Modern Banking Timetable

Simulation Models

This section describe the simulation models and the processing cycle. For instance for DISTRAIN, the background notes describes Customer Response Models, Inventory Models, Staffing Models, Financial Models and the Buying Process. Figure 7.54 shows an abbreviated processing cycle for Modern Banking.

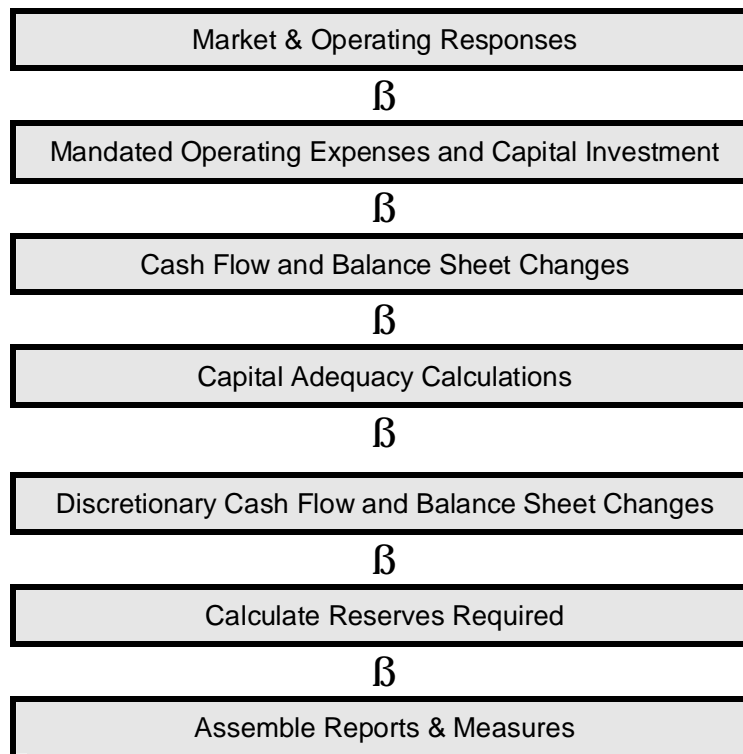


Figure 7.54: Processing Sequence for Modern Banking

Decisions and Issues

This section lists and discusses the decisions and what learners should consider when making decisions. Figure 7.55 shows this for one of the decisions in the DISTRAIN simulation.

Percent Markup

Each Product Category has different responses to price (mark-up) and different optimum prices.

Issues: Besides attempting to maximize Gross Margin, teams must consider the effect of changes in sales on inventory needs and how this effects cash flow and Return on Assets.

Figure 7.55: Example of discussion of decision and issues

Descriptions of the Reports

This section provides information about the reports produced by the simulation.

Run Statistics

This section provides information about how teams have performed in the past. For Product Launch this information was provided as series of statistics (Figure 7.56a) and for Prospector this information was provided as examples of typical teams' results (Figure 7.56b). The statistics for Product Launch (Figure 7.56a) were based on more than 100 runs with a mix of sales, marketing and general management learners. But as usage history was and would be restricted for Prospector, I just used the first eight runs of the simulation to provide examples of the range of possible results.

Cumulative Profit
 < previous worst
 < lower decile
 < lower quartile
 < mode
 > mode
 > upper quartile
 > upper decile
 > previous best

Figure 7.56a

Project Summary								
Team Name	IntoEU	Antiks	SOTP	Grands	Zlatco	Topdog	JJB	Profit
Total Number of Projects	28	38	29	33	48	31	33	23
Number of Opportunities	12	8	18	3	7	11	11	7
Number at Pre-qualify	8	13	4	14	24	6	18	11
Number at Tender	0	3	0	6	7	0	0	0
Number at Negotiation	2	2	4	0	1	1	0	0
Contracts Won	4	5	2	6	5	8	3	2
Contracts Lost	2	7	1	4	4	5	1	3
Opportunity Conversion	7.0	7.6	14.5	5.5	9.6	3.9	11.0	11.5
Pre-evaluation Conversion	4.0	6.0	5.5	5.0	8.2	2.5	7.3	8.0
Tender Conversion	2.0	3.4	3.5	2.7	3.4	1.8	1.3	2.5
Negotiation Conversion	1.8	2.2	3.0	1.7	2.0	1.5	1.3	1.0
Win Conversion %	67	42	67	60	56	62	75	40
Budgeted Work	2423	2423	2409	4381	2150	3286	2228	1994
Average Size Won	572	485	1204	730	430	411	743	997
Value Won	11671	12335	11816	25069	10834	16719	11289	9980
Value Lost	6096	14909	1403	15486	3089	19724	606	2860
Ave. Margin/Revenue Won	11.6	11.3	11.0	10.5	10.6	11.3	11.0	10.0
Ave. Margin/Revenue Lost	11.4	11.4	13.0	10.6	11.8	11.0	11.9	11.9
Remaining Effort	58	37	58	10	23	34	31	50

Figure 7.56b: Sample results (Prospector)

As I build the simulation models I draft the appropriate sections of the Background Notes.

Using Simulations for Training

This tells the trainer on how to use the simulation for training and might include:

- Pre-course Preparation**
- Time Table Planning**
- Briefing the Simulation**
- Running the Simulation**
- Review and Debriefing**

Because the issues associated with using simulations for training are common to most simulations I used a section of my book SIMULATION: Virtual Business Experience to cover this with separate sections on general tutoring issues and separate sections on Tutor Mediated and Direct Use simulations.

Using the Software (Simulator)

This provides information about the simulator (software) and might include:

Installing the Software
Running the Simulator
Problems
Software Information

Installing the Software explains the installation process.

Running the Simulator explains the normal use of the simulator.

Problems lists and explains common problems and suggests solutions.

Software Information is provided for computer services staff and need not be read by the tutor.

As for Using Simulations for Training, because I use a platform I only need two documents explaining software use – one for Tutor Mediated and the other for Direct Use simulations.

Documentation during design

A key element of any software design is the documentation of software and this section provides some ideas about how you might go about this. I feel that you *must* document your model using several overlapping forms of documentation:

In Software Documentation
Model Documentation
Written Documentation
Online Help

Documentation is vitally important if you ever need to revisit the simulation model to extend it or update it. In the early 1990's I was asked by a client to transfer a simulation from a *first* generation microcomputer to MSDOS. Unfortunately, there was no documentation – all I had to work with was the program code. This created numerous problems as the original designer relied on his memory to explain convoluted models and, often, his memory was wrong.

In Software Documentation

Primarily, this involves using *comments*, appropriate *variable names*, *suitable formatting* and *statements per line* when building the model. (See Chapter 7 – Design Craft).

Comments in the model add explanations (Figure 7.57) shows the comments in italics for clarity). It is my normal practice to place a single statement on a line with a comment for each. Not only does it help during design and debugging but it also simplifies and shortens the time taken to modify or customise the simulation.

```
ProfitPCT=100 * Profit/Revenue 'calculate profit percentage'
```

Figure 7.57: In model comments

(McConnell (2002) argues against comments on each line and for comments for blocks of code. However, as a simulation model consists of a series of algorithms, I find it better to comment on a line-by-line basis.)

Suitable variable names

These make the model easier to read. So **Profit** (Figure 7.57) is used to represent operating profit, **Revenue** to represent total sales revenue and **ProfitPCT** is profit expressed as a percentage of revenue. (Personally, I have a conflict here because I use a series of data arrays (Figure 7.8), if I am to use descriptive variable names, I have to copy from an array, perform the calculation and then copy back to the array. I rely on making sure the line is suitably commented.)

McConnell (2002) in his book on programming (Code Complete) dedicates a whole chapter (Chapter 32) to documenting code.

Formatting Code

This impacts the *readability* of the model's code. With the simulation model ranging in size from a few hundred algorithms to several thousand algorithms and total software size between ten and twenty thousand lines of code readability is crucial. McConnell (2002) has a whole chapter addressing the issues associated with the best ways of *formatting code* (Chapter 31). Key to this is plenty of white space, blank lines (to clearly separate groups of algorithms) and indentation (see Figures 14, 15, 16, 24 and 35).

Number of Statements per Line

Even if it is possible to have multiple statements on a line, I believe that it is better to have a single statement as this improves readability.

Model Documentation

I find it useful to make notes as I build the simulation models that record my thoughts as the design proceeds. This helps organise thoughts during design, is a major help if you need to update or extend the simulation (as was the case with DISTRAIN) and eventually will form part of the tutoring manual.

My own architecture allows design information to be provided when producing reports. With the design information option switched on (Figure 7.58), information is provided about the report (its number and help page number) and each result (their numbers and help page).

15 Assets (184) Year 1 Quarter 1		
176	Accounts Receivable (0)	58061
020	Inventories (0)	20375
021	Cash (0)	0
025	Total Assets (0)	78435

Figure 7.58: Report in Design Mode

Figure 7.58 shows a report from DISTRAIN that summarises assets. When the simulation is used normally, the top line would just show – **Assets Year 1 Quarter 1**. However, here the number (15) to the left of Assets is the number of the report in the reporting database. The bracketed number (184) to the right of Assets is the number of the help page describing the report. Below the top line are the results and for each of these have a variable number to the left. The bracketed number to the right of the variable number is the help page number. Here all variable help pages are zero indicating that no help record has been created for the variable. Another aspect is that although Inventories and Cash have sequential variable numbers, Accounts Receivable has a much higher number. This is because the original simulation (Distribution Challenge) did not have an Accounts Receivable variable and when Distribution Challenge was customised to create DISTRAIN was it necessary to add this variable – hence the out of sequence number.

Build online help database

Each team decision, result, report and comment should have a help page. Not only will this enable you to build online help for use by the learners but ensures that you have documented decisions, results, reports and comments. Figure 7.59 is an example of an online help record with a title at the top and help text. Beyond this the help page can have graphics, sounds and links to other help pages.

Relative Working Capital/Unit
This is your working capital/unit relative to that of a typical competitor. Thus it is indicative of how effective you are at handling customer payment and inventory, the efficiency (leanness) of your production processes, the agility of your response to changing demand and design.

Figure 7.59: Variable help record.

Help pages should be added to whenever a decision, result, report or comment is added to the design.

Quality Assurance Documentation

This comprises the paper-based information that documents how the quality of the simulation model and it's dynamic response are assured. Figure 7.60 shows a spreadsheet model that was used to cross check creditor (payables) payment and assets. The difference between Creditors and Calc(ulated) Creditors indicates an error. Likewise, the difference indicates a problem because Total Equity does no equal Net Assets (Fixed Assets plus Working Capital). (As a Balance Sheet should balance, the two should be the same.) (Note: because of the issues associated with using spreadsheets, the error may be in the main simulation model or in the spreadsheet.)

Creditor Payments	17665	2952	6655
Input Tax			
Creditors	1052	4242	38282
Calc. Creditors	1083	4211	38218
Difference	-31	31	64
	June	July	August
Total Equity	3052	4385	1713
Fixed Assets	3353	6809	8693
Working Capital	-301	-2424	-6980

Figure 7.60: Creditor and Assets cross check for SEED

Calibration Documentation

This comprises the paper-based and electronic information that documents how the simulation was calibrated and why it was documented in this manner. Figure 7.61 shows part of a spreadsheet used to help calibrate SEED.

SEED Design Worksheet - innovation diffusion				
Coefficients				
Innovation	0.05	Penet Rate	1	
Innovation Factor		3		
Imitation	0.25			
Init. Potential	100000		Unit Sales	Cumulative
Month	Seasonal	Potential		Sales
May	0.04	100000	200	
June	0.04	100000	200	200
July	0.04	100000	201	400
August	0.04	100000	201	601

Figure 7.61: Innovation Diffusion Calibration for SEED