Chapter 6: Design for Value

Content

Introduction

Having worked in manufacturing it is natural for me to transfer lean ideas from the factory to the classroom. This chapter explores the aspects of business simulation design that affects leanness to ensure value in terms of efficient and effective learning and design covering:

- **The Scenario – The Business Modelled**
- **Learning and Use Waste**
- **Lean Simulation Design**
- **Design Novelty and Complexity**
- **Project Management**
- **Customisation**

The parallel between lean production and leanness in learning is deliberate for just as *"The goal of the [production] system is customer focus: to deliver the highest quality to the customer, at the lowest cost, in the shortest lead time."* (Dennis, 2002) I believe that the goal of a business simulation is learning focus: to deliver effective learning, efficiently, with the shortest duration".

As each section in this chapter approach the topic of value from a different viewpoint some aspects are revisited.

The Scenario - The Business Modelled

When the business simulation is used in an academic setting where the students have no company affiliation or where it is used on an "open" course where learners are from several different companies, the business modelled is not important. But, when the business simulation is run for in-company training the choice of scenario is vital and you must consider how closely it replicates the business - there is a range of possibilities:

- **A generic business**
- **A reflection of the industry**
- **A replica of the business**

The Generic Business

At the one extreme, the business simulation is chosen where the model is totally different from the learners' business and may even represent a different industry. Such a business simulation may be useful, if the business simulation is:

- **To challenge management**
- **To provide role reversal**
- **Used by a mix of businesses**
- **The only one to meet objectives**

In these turbulent, ambiguous, and rapidly changing times, the reason for using a business simulation can include intellectually challenging management - forcing them to step outside their own business. Using a business simulation different from the "cosy" business they know does this. But, it is important to explain this objective explicitly to the learners. Further, experienced managers may better able to handle this than younger, more junior business people.

Often it is useful to place managers in the roles of their customers or their suppliers. For instance, it may be useful to have a group of bankers being placed in the roles of the management of a small business (especially, if managers of small businesses are invited to role-play bankers!) This was the situation for the DISTRAIN simulation where it was designed to make the client's sales staff better understand their customers.

Where the business simulation is used on a course where learners come from a variety of businesses the choice of business modelled is less important. Although it should not be too specialised and so not relate to the learners' experience and perceived needs. In this situation, a generic manufacturing business simulation is often used. However, there are other options. For example, a major management school required a business simulation to explore marketing concepts and strategy. And, they wanted a service based business rather than a manufacturing one. To provide sufficient detail and realism, a specific business was modelled - a hotel. This was chosen because it was felt that the course learners would have (as customers) experience of and views about hotel marketing.

When choosing a business simulation, development objectives should be matched with the simulation. Ideally, this match should be with the learners' industry. However, you may be faced between choosing a long, complex business simulation related to the learner's industry or one divorced from the industry but precisely meeting the learning objectives. This is a difficult choice. The complexity and lack of focus of the industry specific business simulation may limit learning or require an overly long session. On the other hand, learners may not see the relevance of the other business simulation. Experience suggests that industry focus is more important for junior management than it is for senior management. In both cases, if the non-industry business simulation is used, the reasons for its choice and scenario should be explained to the group.

Industry Reflection

Here the model is not an exact replica of the learners' business rather it reflects the general characteristics, financial and operating structures and issues of their industry. The marketing environment, financial and operational structures and terminology should be similar to but do not fully replicate the learners' business. Over the years I have developed business simulations for most industry sectors – manufacturing (high-tech, heavy, FMCG), retail, services, distribution, banking, not-for-profit and even casinos! All of these differ in terms of markets, operational and financial structures and behaviours. (Note: the banking simulation is not an amalgam of the not-for-profit and casino simulations!)

The market model should replicate the markets' structures (selling direct, through distributors, etc.), buying patterns (repeat purchase, durables, consumables, etc.), etc.

The financial model should be similar in terms of fixed & variable cost relationships, working capital & fixed assets relationships, price margins etc.

The operational structure should reflect that of the learners' industry in terms of product/service delivery, resources, flows etc.

Terminology is, perhaps, the most important aspect. Ensuring that the financial and business terms used in the business simulation are similar to those used by the learners is a major step. With many business simulations it is relatively easy to change the simulator's database to do this. If the learners' brief is available electronically it is easy to change this. Provided the marketing, financial and operational structures are similar to the learners' business, just changing the terminology can be a quick, cheap and effective method of tailoring. For instance, a generic retail simulation was changed into one for a distribution company. This took less than half a day and only involved changing terminology (neither the model nor the master data were changed). Yet, on running, one learner, a Managing Director, congratulated us on realism. He stated that we must have spent a long time with their finance department tailoring the simulation!

The Replica Business

Here the business simulation attempts to model, exactly, the learner's company. When I was involved in building and advising on the use of corporate some forty year's ago, the holy grail of corporate modelling was to build a model of the complete business. Every one of these failed because of their complexity, development costs and changing needs and business.

Even a complex business simulation can never, completely, replicate the business. This is especially true for the behaviour of markets and customers. As some learners may come to the simulation with preconceived views of their business. If the simulation does not reinforce these views, even if the views are wrong (as they often are), the model will be perceived as inaccurate and the business simulation useless.

Recently I developed a simulation that replicated an industrial chemicals business that produced two chemicals in tandem. Because of this and other constraints, the dynamics were exceptionally difficult and as a result it was impossible to make a profit – a situation that was real but caused disaffection.

Another problem with using a replica business is their necessary long duration and the pressure to reduce training time. As a consequence, I see the replica business simulation to be appropriate for full-time academic programmes but not for company training.

The Case Study Simulations

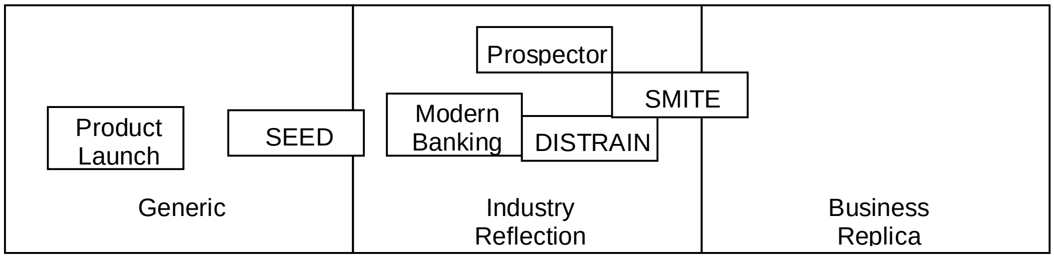


Figure 6.01: The case study simulations positioned across the spectrum

Both Product Launch and SEED have imaginary products and so are generic simulations. For Product Launch this is reasonable because the issues addressed are basic and duration had to be very short. For SEED the scenario was chosen to resonate with the students using it and as none had any company affiliation a generic simulation was appropriate. (A second market for SEED was to train groups of prospective business people and as these would be running different companies a generic simulation is appropriate.) Modern Banking is an Industry Reflection simulation but because it had to be reasonably simple it is positioned towards the bottom of the Industry Reflection domain. In contrast, Prospector is in the middle of the Industry Reflection domain. Because they are more closely related to *actual* businesses, DISTRAIN and SMITE are towards the top of the Industry Reflection domain (with SMITE overlapping very slightly with the Business Replica domain).

Positioning the Simulation on the Spectrum

Where a business simulation is positioned on the spectrum ranging from the generic to the replica has an impact on:

- **Relevance**
- **Complexity**
- **Duration**
- **Cost**
- **Learning**

Relevance

Where the business simulation is used in-company, the business situation must be structurally similar to the company and the issues that the learners are facing or will face in the future must be the same. Thus, In terms of *relevance* the Company Replica simulation will be most relevant and a Generic simulation the least relevant. But, because the Industry Specific simulation has the correct financial, market and operational structure it is likely to be acceptably relevant.

This is illustrated in Figure 6.02. Here relevance increases as one moves from generic to industry reflection. An improvement that is caused by the fact that the industry reflection simulation will address the relevant issues and have appropriate operational, market and financial structures. The relevance curve saturates as it moves from industry reflection to replica and, as it moves further into the replica domain, relevance lessens as the simulation begins to include decisions and issues that are not appropriate to the learner's job.

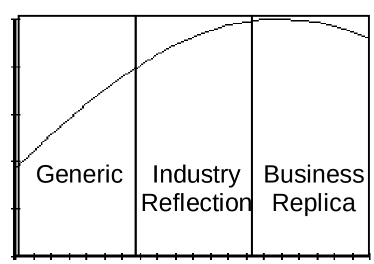


Figure 6.02: Relevance

Complexity

In terms of *complexity* a Business Replica simulation will be far, far more complex than a Generic simulation because a really accurate model is very complex. But an Industry Specific simulation will be only slightly more complex than a Generic Simulation. This is illustrated in Figure 6.03 where the curve is relatively flat for Generic and Industry Reflection simulations but increases significantly for Business Replica simulations.

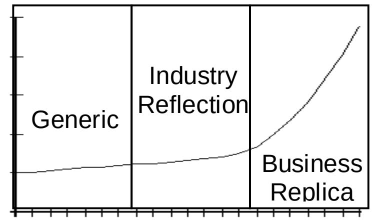


Figure 6.03: Complexity

Duration

As *duration* is strongly correlated with *complexity* (Hall & Cox, 1994) the shape of the duration curve is similar to the complexity curve with Business Replica simulations taking much more time to run than Generic simulations and Industry Specific simulations taking only a little longer than Generic simulations.

Cost

This consists of two parts – the cost of development and the cost of use. The cost of development is highly correlated with complexity (Hall, 2006). The cost of use depends on duration (that this is highly correlated with complexity (Hall and Cox, 1994)). The cost relationships are similar to complexity with Business Replica simulations much more costly than Generic simulations but Industry Specific simulations are only slightly more costly than Generic simulations.

Learning

Although *relevance* improves learning, *complexity* (as in the real world) can become confusing and detract from learning. Perceptions influence learning and I believe that *learning* peaks for Industry Specific simulation with Generic simulations slightly less effective and Business Replica simulations significantly less effective. This perception that Business Replica simulations do *not* provide good company training is based on experience with actual use on company training courses. However the position of the peak depends on maturity with immature students requiring a business simulation closer to being a replica of their business.

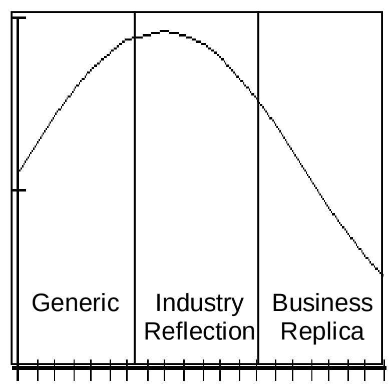


Figure 6.04: Learning

Learning and Use Waste

A core aspect of leanness is identifying and eliminating wastes. For business simulations these divide into two parts – those associated with the learning provided by the simulation and those associated with its use.

Learning Waste

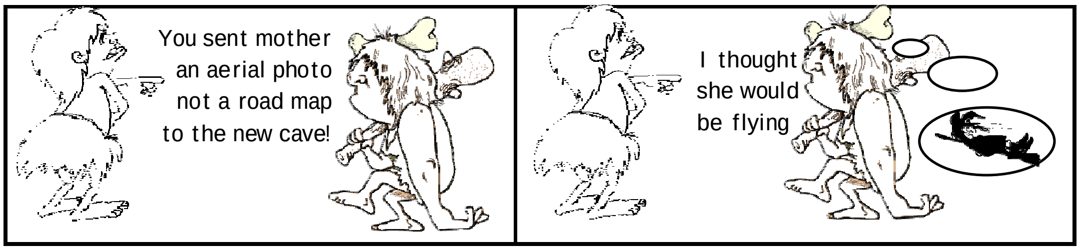
This waste is associated with the way it effectively and efficiently delivers learning and consists of:

- **Simulation Focus Waste**
- **Complexity Waste**
- **Confusion Waste**
- **Prior Knowledge Waste**
- **Experience Waste**

Simulation Focus Waste

Simplification and stylisation are commonly used when designing models to focus on purpose. Consider journey planning. Which is better for this, an aerial photograph or a road map? The aerial photograph is the most *real* iconic model but it has a confusing amount of irrelevant detail and so is virtually useless when journey planning. In contrast, the road map eliminates irrelevant, reducing confusion and focussing on purpose.

(However, there may be situations where an aerial photograph would be best – Cartoon 6.01.)



Cartoon 6.01: Models must be relevant

If a business simulation is not aligned (focused) wholly on the desired learning needs and objectives it will deliver irrelevant learning and waste learners' time. We can look at the business simulation and learning needs as two sets (Figure 6.05) – a set that covers the learning provided by the simulation and a set that defines the learning needs.

A + B = Business simulation

B + C = Learning Needs

B = Relevant Learning

A = Irrelevant Learning

B / (A + B) = Simulation Efficiency

B / (B + C) = Simulation Effectiveness

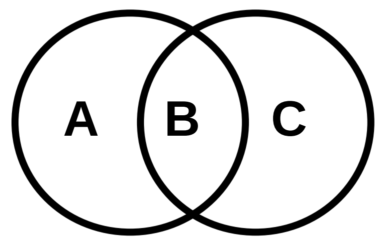


Figure 6.05: Simulation/Needs Sets

Here the business simulation is represented by the **A + B** set and learning needs by the **B + C** set. Only for the overlap (**B**) is the learning *relevant* and **A** is *irrelevant* learning. If **B** is large compared with A then the simulation is efficiently focussed on the needs of the learners.

Area **A** shows where learners' time is *wasted* as they spend time dealing with irrelevant issues and learning things that they do not need (or want) to learn. If **B** is small compared with **A** then the business simulation is unfocused and makes poor use of learners' time and as adult learners realise this they will become disaffected. The larger **A + B** is the longer the time required to develop the simulation, the greater the development cost and the longer duration. Area **A** is *waste* that reduces both learning and design efficiency. This area of waste parallels *overproduction* in a factory – something that Taiichi Ohno saw as *"the root of all manufacturing evil"* (Ohno, 1988). (Arguably, including irrelevant learning is *"the root of learning evil"*!

For the learning needs set, comparing **B** with **C** provides a measure of learning effectiveness. If **B** is large compared to **C** then the simulation is highly directed towards meeting learning needs. But the relationship between **B** and **C** is problematic as it is very unlikely that any simulation will cover all learning needs. The business simulation relative to the time spent on other learning initiatives provides a way of critiquing the simulation. Comparing the proportion of the learning provided by the simulation relative to the total learning need (**B/(B + C**) with the time spent on the simulation compared with the total time spent learning provides a measure of training efficiency. Though measuring learning provided by the simulation relative to the time spent on it is complicated by the fact that the simulation is likely to be addressing *comprehensional needs* and some of the other learning initiatives may be addressing *apprehensional needs.* In other words the other learning initiatives may be providing the basic knowledge that is then explored by the simulation.

How then do we ensure that the simulation focuses on learning needs and does not include spurious elements? A basic philosophy is that elements are only built into the simulation to elicit appropriate learning (cognitive dynamic) and to ensure an appropriate level of engagement (affective dynamic) - avoid building into the simulation anything that is *cool* or is *real* but does not provide learning or engagement. Arguably, a general problem with software design is that extra features are built in (software bloat) because it is easy to do and, for the designer, it is fulfilling (http://en.wikipedia.org/wiki/bloatware - retrieved 29/04/2009).

Cartoon 6.02: Irrelevant Features

For example, Product Launch's purpose was to explore basic marketing and financial concepts – pricing, promotion, the product lifecycle, profit generation and cash flow. It was designed to last only two hours and as duration and the number of decisions are correlated (Hall & Cox, 1994) this limited decisions to three – price, promotion and production. With price having both marketing and financial implications driving demand and impacting margin and the combination of demand and margin impact profit. Promotion has a marketing impact (making people aware of the product, getting them to trial it and then reminding them to repurchase) and through this demand. From the financial viewpoint, promotion impacts fixed costs. Finally, the production decision forced learners to forecast the impact of their price and promotion decisions and had impact on costs, profit and cash flow.

To constrain complexity several marketing and finance aspects were excluded. What was left out? In marketing terms several things – first there was only a single market and there was no opportunity for product development. Although in reality, there would be multiple market segments for the product each with different potential, pricing and promotion responses. Likewise, as the product reached the end of it's life cycle, one would normally modify/replace the product and extend the product range or markets served. But an exploration of this could be part of the ending review discussion and so there was no need to include these issues. Likewise the financial model left out depreciation, assumed that payment was made immediately (no debtors or accounts receivable) and there was no tax. And, instead of having a mix of debt and equity, the only source of cash was the initial 400,000 plus any accumulated profits and less investment in capacity and inventory. Additionally, there was no dividend or interest paid on the initial 400,000 or interest earned on accumulated profits. In hundreds of runs with thousands of business people over more than thirty years, no one questioned these simplifications and they never compromised learning.

Complexity Waste

Complexity drives the need for cognitive processing and cognitive processing coupled with the time available (duration) leads to cognitive pressure (workload). As discussed in Chapter 4 (Design for Process), if cognitive pressure (workload) is too high or too low this leads to role overload (French and Caplan, 1972) and disaffection. As a result, for effective learning, *duration* must match *complexity* (Hall & Cox. 1994). Additionally, complexity (in terms of simulation model size) is also highly correlated with design time (Hall, 2006) and this means that inappropriate, irrelevant complexity both lengthens the simulation's duration and it's design time – inappropriate complexity is *waste.* In a lean manufacturing sense, increased design time equates to *material waste* and a longer duration equates to *process waste*.

Confusion Waste

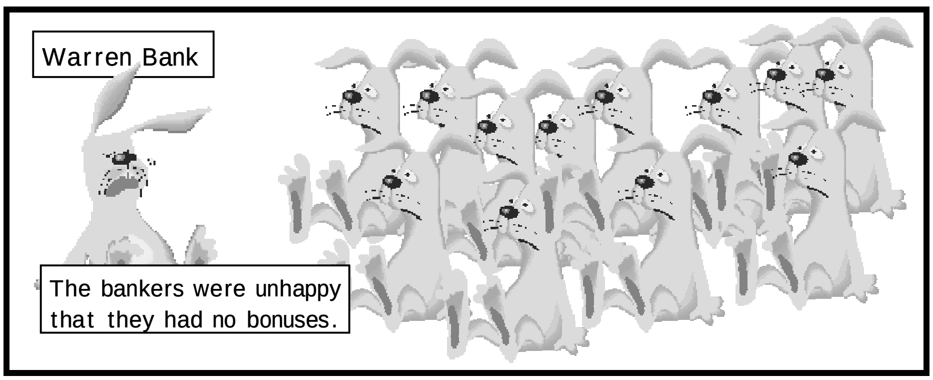
A second problem is that complexity and the links between interactions and the model may be confusing and cause waste. Although the issues covered by the simulation address learning needs, the links between decisions and results may be too ambiguous and learning does not happen. This can be ameliorated by suitable consideration of these links, tutor support information and reducing the ambiguity of the decisions and results.

There are several ways to reduce confusion waste.

1. The relationship between the model and interactions can be simplified and stylised.
2. Relationships can be moved from complex intersecting ones to simpler, more direct relationships.
3. The simulation's Tutor Support System is designed to make explicit the links between decisions and results.
4. Decision and result ambiguity can be reduced.
5. Textual *comments* can be used to help *reveal* the relationships and stimulate discussion.

Prior Knowledge Waste

Simulations address higher order learning needs (Whiteley, 2006) and so there is a need to take into account prior learning – the knowledge that learners must bring to the simulation. Where prior learning is an amalgam of previous learning, learning provided earlier by the course and the mix of knowledge and experience of the learners.



Cartoon 6.03: Unhappy bunnies

What is the relevance of the cartoon? Obviously bankers will be unhappy if they lose their bonuses. But why represent them as bunnies? British readers are likely to have heard of the phrase *"happy bunnies"* but my American readers may not have. Rather, they will have heard of *"happy campers".* Just as prior knowledge is vital to understanding this cartoon, it is as important to using a simulation. (I could have drawn the cartoon with campers except that bunnies are easier to draw!)

This has several possible wastes Figure 6.06:

1. prior learning that must be refreshed and revisited by the simulation,
2. irrelevant prior learning that is duplicated by the simulation
3. missing prior learning that must be provided.

The dashed line circle (figure 6.06) shows the prior learning, the solid line circle shows learning needs and the double line circle shows the learning provided by the simulation.

P1 is the prior the learning that will be refreshed and used by the simulation and that is relevant to learning needs.

P2 is prior learning that is relevant to leaning needs but is not addressed by the simulation and may need to be refreshed or used by other learning initiatives.

P3 is prior learning that is both relevant to learning needs and not used by the simulation and as such does not cause waste.

P4 is the prior learning refreshed and used by the simulation that is not relevant to the current learning need and so is *waste*

P5 is learning addressed by the simulation that is relevant but where there has been no prior learning. This may mean that the simulation (or the trainer running the simulation) will need to provide this.

P6 is worst as not only is the simulation-derived learning is irrelevant but also prior learning is missing and will have to be provided during the simulation.

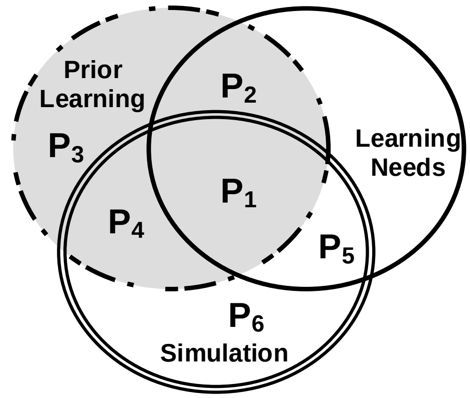


Figure 6.06: Prior Learning and Simulation

Ideally, **P4, P5** and **P6** should be small compared with **P1**.

Figure 6.7 shows the learners' prior learning as several overlapping but not identical dashed line circles. The extent of this spread depends on the mix of experience and knowledge that is available to the team running the simulated business.

Ideally the simulation's prior knowledge needs (double line circle) is completely covered by the prior learning of the learners (multiple dashed circles). However, no individual learner is likely to have adequate prior knowledge. Also, the simulation's prior knowledge needs is strongly related to the learning needs.

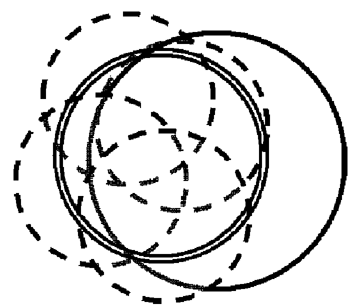
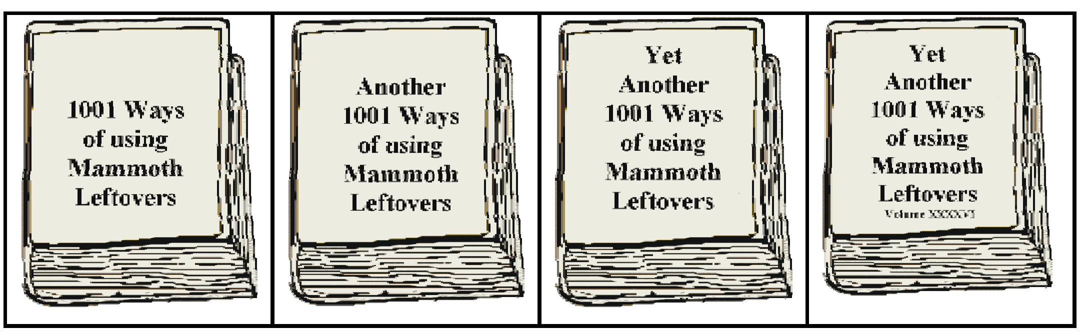


Figure 6.07: Ideal Prior Learning

A second facet of prior learning is duplication of *comprehensional* learning. Unless the purpose is to refresh this learning, any duplication is wasteful and should be minimised. To minimise comprehensional waste, it is necessary to be very clear about prior learning needs (**P3**) and the learners' prior learning (**P1** to **P4**).



Cartoon 6.04 Simulations may duplicate prior knowledge.

Experience Waste

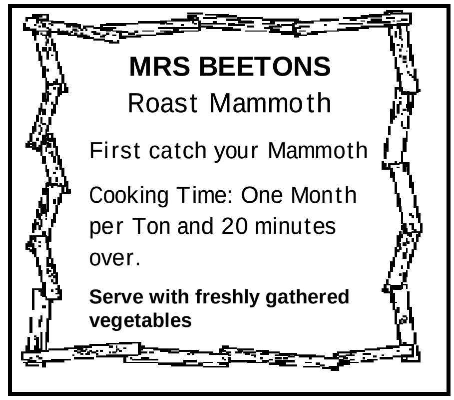
This occurs when the simulation does not provide an adequate level of experience and this may be caused by too little ambiguity, irrelevant learning or a short-circuited experiential learning cycle. When designing the SEED simulation it was important that there were several entrepreneurial opportunities and none of these were either obviously right or obviously wrong as this would have limited discussion. Consequentially, having identified three opportunities (Early Learning, Companions for the Elderly and Autistic

Socialisation) I calibrated these so that all had similar potentials but different characteristics and risks. A second situation where the simulation is not perceived as relevant for the learners. I emphasise the phrase *perceived as relevant.* For example, I feel that a basic understanding of finance is essential to any businessperson. Yet, I have found that some junior managers (and occasionally senior managers) think that this is not so and see learning about basic finance as irrelevant.

Simulation Use Waste

Besides simulation choice waste there are other wastes that add to simulation duration and mean that learners waste time as thus:

- **Decision Entry Waste**
- **Result Waste**
- **Facilitation Waste**
- **Recovery Waste**
- **Architectural Waste**
- **Interface Waste**
- **Process Waste**
- **Workload Waste**
- **Documentation Waste**



Cartoon 6.05: Usage Waste

(Note: one month per ton (Cartoon 6.05) equates to twenty minutes per pound!)

Decision Entry Waste

This involves the learner or the trainer wasting time while entering decisions. This may because it takes time to find how to enter decisions, convoluted decision entry, having to enter decisions unnecessarily, misunderstanding decision entry, etc.

Finding how to enter a decision

This is a matter of interface design and, in particular, if the screen display is *cluttered* or *confusing.* (As such this parallels a factory where a *cluttered* working environment causes waste and errors (Goldsby and Martichenko, 2005).) This is a particular problem where the simulation is used directly by the learners and, in extremis, used by a *team* of learners. Where the simulation is used directly, the learner(s) interface design is crucial (see Interface Waste). Where the simulation is used by a team, learners will sit further from the screen than if used by an individual. This means that the size of information on the screen (such as font sizes) must take this into account. A good default font size is 12 point but this is a compromise (as a large font will limit the number of decisions on a decision entry template). For Tutor Mediated simulations, the trainer can become familiar with decision entry before using on a course and this eliminates this waste.

Convoluted Decision Entry

This is where entering a decision involves repeated mouse clicks, precise position of the mouse or complex keyboard use. Although many users are not expert at using the keyboard, often it may be quicker to use the keyboard rather than the mouse. (I feel that it is better to use a mouse rather than relying on the touch pad built into the computer. This is because (for me) the touch pad may be too sensitive and may initiate unwanted actions.) This waste is a particular problem for Tutor Mediated simulations as the trainer will be under considerable pressure to enter decisions quickly and must enter several sets of decisions.

Unnecessary entries

This occurs because a decision has not changed but it is still necessary to enter the decision. For example, it might not be appropriate to change price every period, yet the simulator might require the price to be re-entered each period. Rolling forward previous decisions and initialising others appropriately can overcome this. For example, as it would be inappropriate to purchase new capacity repeatedly and this decision might be initialised to zero (rather than using the previous period's value). But price, promotion, staffing levels and such would be *rolled* forward.

Decision entry misunderstanding

This means that the decision entered does not reflect what the learners wanted and is likely to require rerunning (rework) and cause disaffection. For example, a decision might be to *increase* productive capacity but instead learners mistake this and enter the *total capacity* that they want. Although misunderstandings can never be fully eliminated, they can be minimised by decision checking and suitable online help. For example, where learners enter total capacity rather than no increase, the decision check could warn that this decision would double capacity.

Result Waste

This occurs when it is difficult to interpret the results because they are incomplete, presented in a confusing, convoluted, inappropriately ambiguous or too granular way.

Incomplete results

Here there is insufficient information for the learner to link cause to effect. Here I mean incomplete rather than ambiguous because some degree of ambiguity is necessary to ensure that learners reflect appropriately and deeply. For example for the SMITE simulation, learners needed to be able to link results with causes and this meant that I had to provide reports that looked at the business from several viewpoints (area-by-area, sales person-by-sales person etc.) and backed these with comments from the sales people and *"head office".* Combined these ensure that there was sufficient information to enable learners to *tease out* cause and effect.

Confusing results

These occur when there are too many results to analyse properly in the available time (*analysis paralysis* (Teach, 1990)) or where the arrangement of the results or their display is confusing. Teach (1990) goes on to suggest that the amount of data correlates with the knowledge and experience of the learners – a view that I agree with wholeheartedly. For example, when producing a version of Executive Challenge for school pupils, the existing reports (designed for senior specialists and junior managers) would be too complex and so I reduced the number and range of results.

Unfortunately, incomplete results and confusing results are two ends of the same spectrum. Reducing confusion may be at the expense of incompleteness and visa versa. Consequentially, the designer is always attempting to balance completeness with clarity.

Convoluted results

These occur because results are too elaborate. For example, for both Modern Banking and DISTRAN, the Profit and Loss Account (Income Statement) did not detail all cost areas rather it showed these accumulated into a few key groups. But occasionally, learners do not understand how a particular cost was calculated and, part of the *reconciliation* reports detail each cost area.

Too ambiguous results

These mean that it is impossible interpret for the learner to link cause to effect. This is a particular problem when the results are impacted by several decisions. For example, in DISTRAIN; Mark-Up, Advertising, Staffing Levels, Product Range etc all impact Sales. This problem was addressed in several ways – business research allowed learners to explore how Mark-Up impacted Market Share, comments helped indicate problems with Staffing Levels and Product Range and the Tutor's Audit Market Response Report (Figure 6.08) revealed the impact of the factors that affected sales (for the tutor so he or she could identify problems and discuss the impact of decisions with the learners).

Marketing Responses - Year 0 Quarter 4			
	Counter	Industrial	Commercial
Price Response	1.06	1.19	1.08
Promotion Response	0.69	0.69	0.69
Product Range Response	1.00	1.00	1.00
Improvement Response	1.00	1.00	1.00
Receivables Response	1.00	1.00	1.00
Market Response	0.74	0.82	0.61

Figure 6.08: Providing unambiguous tutoring information.

Although for Tutor-Mediated simulations, inappropriately high ambiguity can be dealt with by providing information to the trainer, for Direct Use simulations this is not usually possible. Here a useful approach is to use subjective, qualitative comments. For SMITE, where the factors that drive sales are complex, interacting and ambiguous, and as described earlier the problem was dealt with by providing comments from two sources – individual sales people and "head office".

Too granular results

This means that the results are too detailed and take too long time to analyse or interpret. It may be wrong to show financial results to two decimal places or even to the nearest 1000 $s, £s or €s (Figure 6.09).

Revenue & Cost Summary	
Total Revenue	1234567.89
Total Cost	1000000.09
Total Profit	234567.80

Figure 6.09: Excessive Granularity

A second aspect of *granularity* is the amount of information provided in a report. Recently, while running a train-the-trainer course for Product Launch, one trainer felt that it was wrong to include Sales, Profit and Loss and Assets in one table and that these should be shown in separate reports. However, doing this would increase granularity and hence cognitive load and simulation duration. Another thing I do to reduce granularity is to use a mythical currency (the Account Unit (or AU)) as this allows me to express costs with fewer digits (e.g. Annual Labour cost 25 AUs rather than £25,000 or $40,000). This has two additional benefits – I do not need different versions for different countries and I do not to be updating the simulation continuously for the effects of inflation (the chemists among you will recognise that AU is the chemical symbol for Gold!)

Too little ambiguity

This is a problem as it means that the link between cause (decisions) and effect (results) is too obvious. Where this happens, learners will not think deeply enough about their actions. Unfortunately there is a conflict between this and *too much ambiguity* (above) and designers must balance the two.

No reflection waste

This is where the learners oscillate between active experimentation and concrete experience and as a consequence do not reflect or conceptualise (figure 6.10). Without reflection or conceptualisation learning does not occur and the experience is wasted. Typically this problem occurs when learners cannot link cause and effect or where they are able to make decisions without consequence (Hall, 1995b).

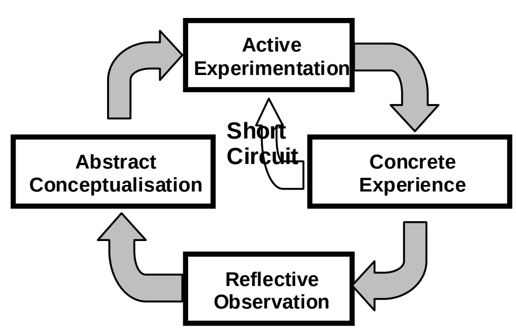


Figure 6.10: Short-Circuited Experiential

Facilitation Waste

This occurs because learning is delayed because explanations and coaching information is not available expeditiously. For example, a client ran a simulation both on Stand-Alone Workshops and as a remote activity where the decisions were submitted to the facilitator for processing and the results returned. At the workshops, the facilitator could coach and answer questions. To provide similar support for the remote activity, each team of learners had a mentor. Except that one mentor was not available – as a consequence this team spent about *three-times* as much time on the simulation because facilitation support was unavailable.

Architectural Waste

This occurs because the simulation's software architecture does not ease design and customisation because of the lack of existing software components and because the simulation is not driven from data files and databases. The time saved by using standard components can be huge. Excluding travel, Modern Banking took about 36 man-days to create compared with a banking simulation created by a university that took 360 man-days to create (from scratch using EXCEL™. Where the simulation is driven from data files and where all that is required is to change language, a new version can be developed in hours. (See later sections on data structures and customisation.) The architecture that I use is described later.

Recovery Waste

This occurs when it is necessary to recover to an earlier period or stage or to recover from a printer problem. This may be necessary because of a software *bug,* incorrect decision entry, improper simulation, to furnish learning opportunities or to handle a printer problem or other hardware problems.

Software Bugs

These should not occur during use but they do because an unexpected decision or combination of decisions can occur during use. For example, recently, a simulation that had been in use for nearly 25 years crashed because the person made a ridiculously low production decision!

Incorrect Decision Entry

This occurs because of misunderstandings and mistakes. (As illustrated by the increased production capacity decision described earlier.)

Improper Simulation

This occurs when the learners or the trainer *accidentally* simulates the period before all the decisions are entered or before the decisions entered have been properly checked. (This may be due to inadvertently clicking on the simulate button or because the touch pad is too sensitive and this invokes simulation.)

Learning Opportunities

It may be appropriate to repeat a period or stage with different decisions to compare outcomes. This situation can occur when there are differences within a team and they would like to see the outcomes of an alternate strategy.

Printer Problems

Unfortunately, these are commonplace – the printer may jam, run out of paper or ink or toner. The impact of these problems can be minimised by printing after simulation and saving the current data before printing. Then results can be printed after correcting the problem. (To ameliorate running out of toner or ink it is *vital* to have a spare cartridge!)

Other Hardware Problems

Today's computers are reliable but problems can occur. For example, one client came to us because a *feature* of the simulation that they were using was that the computer could not be switched of during its use. If this happened, it was necessary to re-enter *all* the decisions from the beginning. Further, because the simulation was complex and stochastic there was no certainty that the results would be the same. Naturally, being helpful and to save electricity, the overnight cleaners switched off the computers!

Reducing Recovery Waste

Recovery waste can be reduced by speeding and easing the ability to rerun a period or stage. This can be done by *journalising* period or stage data and, if appropriate, saving decisions. Data journalisation means that at the end of each period or stage I save all the team data but I only save decision data for Tutor Mediated simulations and complex Direct Use simulations. Whether saving decisions is germane depends on the number of decisions made. If less than about six, saving decisions is not necessary. But above six decisions, saving decisions is helpful and for more than a dozen decisions necessary. As a consequence I save data for all Tutor-Mediated simulations but only for the more complex Direct Use simulations. So data was saved for Modern Banking, DISTRAIN and SMITE (Tutor-Mediated) and Prospector (Direct Use) but not for Product Launch and SEED (Direct Use).

Although it is generally only necessary to recover to the previous period, today's large discs means that it is possible to keep decision and data from *all* past periods and, this allows one to recover to *any* previous period or stage. (Journalising *all decisions* is very helpful during the design stage when one is attempting to *calibrate* the economic response of the simulation as this means that all one has to do is change parameters and simulate without having to re-enter decisions.)

Besides saving at the end of each period or stage, for some simulations it may be appropriate to save during decision entry or simulation. Saving during decision-making is appropriate when this is complex involving many decisions. Saving during simulation is appropriate for complex planning, process or operations management simulations. For example, Prospector involves several sessions where the learners move the simulation forward in *micro-steps.* This means that they could have used the simulation for a considerable time when a problem occurred. If data were only saved at the end of the session, recovery would require *all* the work from the start of the session to be repeated. Saving the current position continuously to a temporary file prevented this. If a problem occurred, the simulation would reload the current position from the temporary file.

For Direct Use simulations it is not advisable to allow learners to rerun a period, but is necessary for the trainer to be able to do this. (Learners should not be allowed to rerun a period or stage as this may result in insufficient thought and a short-circuited experiential (Figure 6.10) (Hall, 1995b)).

Interface Waste

Is where the interface design incorporates features such as graphics and animations that take considerable time to design but do not contribute to learning (and may lead to Decision Entry Waste (see earlier)). A cluttered or illegible screen can cause this waste. I try to have the screen showing just the current task rather than showing every possible thing. Al described earlier, to improve legibility I use a reasonably large font whenever possible (12 point). Only where this would result in a report wider or longer than the screen do I use a smaller font size.

Process Waste

Process waste is due to bottlenecks in the simulation process and this causes the learners wasting time awaiting the results. This may because it takes a long time to process the decisions or too long to provide them in a suitable way. This waste may be caused by network access, the use of spreadsheets, the time required to print and, where a trainer is entering decisions, the time he or she takes to do this. A typical processing cycle is shown in Figure 6.11.

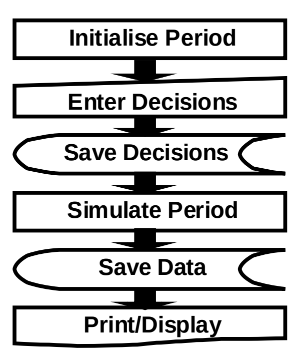
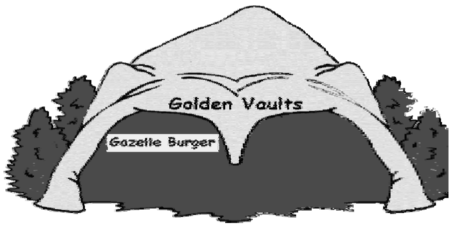


Figure 6.11 Period Process

Some of these can be automated but if this is not possible or not done the step may be a bottleneck.

Initialisation Delays

These have been discussed earlier as part of Decision Entry Waste and Recovery Waste. Normally automating initialisation and recovering from saved decisions will eliminate them and so this does not cause waste.



Cartoon 6.06: Process speed is vital

Decision entry process delays

Where the trainer enters decisions (Tutor Mediated Simulations) there can be process delays but this problem can be overcome if decisions are submitted early and decision entry starts before the decision entry deadline. For this to be possible it must be possible to enter the decisions in any order.

Decisions Recovery

When decisions are saved, decision recovery will be automatic and will not add to Process Waste. I learnt this lesson many years ago when I entered a wrong decision and simulated. This led to plaintive cries of anguish and my having to re-enter the decisions for several teams while senior managers milled around waiting for the re-run.

Simulation Delays

These are less of a problem as today's computers are fast. However, if the simulation is based on a spreadsheet, overheads associated with their use may mean that for a very complex simulation, calculations may take a considerable time. (I remember on one occasion where we used a spreadsheet model to help the learners with decision-making, we suggested that they go have a cup of coffee during recalculation.)

Printing Delays

Printers are a bottleneck, yet, printed results are desirable because the amount of data that can be displayed on paper is much larger than the amount that can be displayed on the computer's screen. (Where the results are printed, learners can make notes on the paper and the paper-based results act as a permanent record. However, printers are bottlenecks and where results consist of several pages, it may take a considerable time to print the results.)

This depends on the speed of the printer and the sequence that the results are produced. For example, for my Tutor-Mediated simulations (such as Modern Banking, DISTRAIN and SMITE) where several teams are in direct competition, I produce results in stages (Figure 6.12).

1. Preliminary Results
2. Full Team Results
3. Business Research
4. Tutor's Audit
5. Team Commentaries

Figure 6.12: Result Sequence

Workload Waste

This waste occurs because the learners are not fully occupied as illustrated in Figure 6.13a. This happens if the complexity of the simulation is not *ramped* (Figure 6.13b) where new issues (decisions and reports) are introduced or the economic environment becomes more challenging.

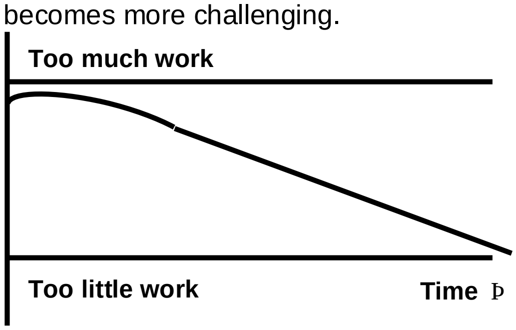


Figure 6.13a: No workload ramping

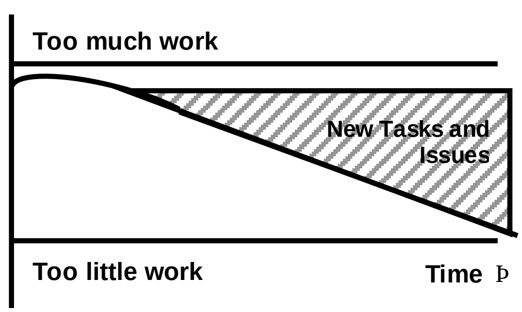


Figure 6.13b: With workload ramping

The shaded section in Figure 6.13b shows the area of waste as workload falls off and by introducing new tasks as the simulation progresses workload is kept high and waste minimised. An example of how this was done is described later in the customisation example.

Documentation Waste

I am a great believer in *leanness* in documentation and keep learner and tutoring documentation short (believing that if long it will not be read). This is illustrated in figure 6.14 that shows typical documentation lengths (in A4 or Letter Size pages).

Simulation Duration (hours)	< =4	8	12	> 12
Learner's Brief (pages)	2 - 3	8 - 12	10 - 16	16 +
Simulation Brief (slides)	7 - 12	9 - 17	12 - 20	18 +
Background Notes (pages)	8 - 12	15 - 18	18 - 24	24 +

Figure 6.14: Documentation Sizes

Note: If worksheets are included in the Learner's Brief or Tutor's Background Notes this adds pages. As I usually develop simulations up to a day's duration, the figures for simulations with durations longer than a day are based on a small sample.

Simulation Architecture

This section describes my award winning simulation architecture and platform that has cut simulation design and customisation time very significantly and discusses the reasons underlying the architecture. This architectural level model development raises several issues - structure, architecture and the modelling language.

Structural Issues

A simulation has two types of code - the model itself and the code that manages the user interface, report generation, decision input etc. (Hall, 2003).

This separation has implications in terms of the economics and agility of design. Although the model will be different for different simulation, the code that manages the user interface, report generation, decision-input etc. can be common to several simulations. This is a similar situation to those facing automobile designers where it is common to have a standard *platform* that is used with different body styles and engines. Experience developing new simulations and transferring existing simulations into a shell suggest that the model generally only accounts for between 5% and 15% of the code with the shell being the remaining 85% to 95%.

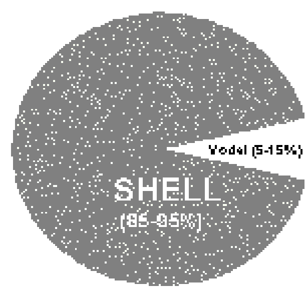


Figure 6.15: Code Proportions

By using a shell, development time can be reduced by 80% or more. Even if you do not use shell programs, it is sensible to separate the model from the reporting and decision entry routines. This is because as platforms change, the model can remain the same even though the user interfaces etc. change. Since the early 1970s, I have moved simulations from Computer Timesharing, through first generation microcomputers (Tandy 1, Apple 2), to MSDOS, and to Windows. As *basic* management development and business training needs are not changing this extends product lifecycles significantly (the Product Launch simulation was originally developed in 1977 and it is still in use over thirty years later). As discussed in the methodology section the appropriate design method differs between developing the shell (components) and the simulation model.

An Architecture

Figure 6.16 shows the architecture that I use and the links between the components.

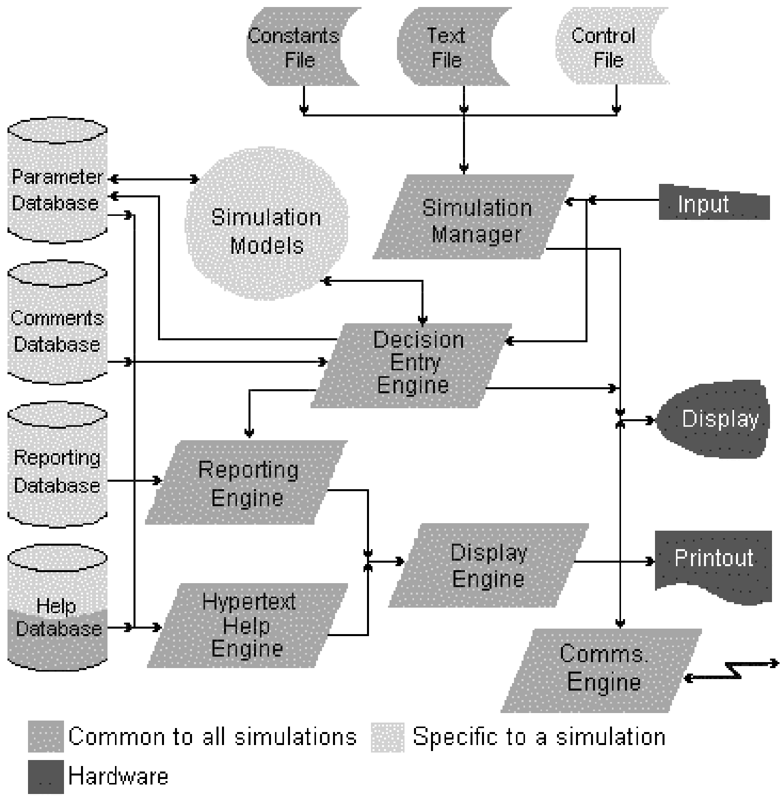


Figure 6.16: Simulator Architecture

With over forty simulations in my products range together with the need to develop simulations for clients on a regular basis it was decided to implement the architecture as a shell (or platform) that was common to many simulations and where a specific simulation only differed in its simulation model and associated data. (In figure 6.16, the *light grey* software components are those that are specific to an individual simulation and the *medium grey* software components are common to many simulations.

Elements of the Architecture

As discussed earlier, I use a series of standard shells, each of which is common to several simulations and provides 85% to 95% of the code. The only difference between one shell and another is the Simulation Manager as the Decision Entry Engine, the Reporting Engine, the Hypertext Help Engine and the Display Engine are all common.

The Common Components

These are the common components:

- **Simulation Manager**
- **Reporting Engine**
- **Decision Entry Engine**
- **Hypertext Help Engine**
- **Communications Engine**
- **Display Engine**
- **Text File**
- **Constants File**
- **Help File**

The Simulation Manager
This controls the processing sequence and a typical but simplified processing sequence was shown earlier (figure 6.11).

The simulation manager differs depending on the type of simulation and below is a list of the simulation managers that I use:

- **Tutor-Mediated**
- **Direct Use**
- **Decision Support**
- **Planning**
- **Process**
- **Negotiation**

Tutor Mediated Simulations use a single computer and where the trainer enters decisions on behalf of the learners. Typically, these are interactive, multi-team simulations. Examples of these types of simulation are DISTRAIN and Modern Banking.

Direct Use Simulations involve learners entering their decisions into their own microcomputer. Typically these are short concept simulations and ones where it is not necessary for interactions between teams. An example of this type of simulation is Product Launch.

Decision Support Simulations involve learners using their own microcomputer to plan their actions and using a Tutor Mediated simulation to determine the impact of these plans.

Planning Simulations involve learners making a business plan using the simulation model to test and assess their assumptions. An example of this type of simulation is SEED

Process Simulations are used to help learners explore the use and implications of statistical and management science techniques. An example of this type of simulation is the Prospector simulation.

Negotiation Simulations involve a role-playing negotiation with the buying and selling groups each supported by their own simulation model.

The processing sequence for Direct Use and Tutor Mediated Simulation Managers are shown in Appendix 2.

The Simulation Manager is configured by the Control File (see below) and the structure of this file differs to some extent depending on the type of simulation and its contents define the simulation version.

The Reporting Engine
This allows the designer to define and modify the reports produced by the simulation as necessary. The formats for the reports are defined in the Reporting Database rather than in code. When the Simulation Manager calls for a report to be produced, the report is called from the reporting database and populated with data from the Parameter Database. Storing report formats in the Reporting Database not only allows additional reports to be added to create new versions but also handles the large number of reports required for today's simulation. For example, my simplest simulation (Product Launch) requires 57 reports. For simulations of intermediate complexity (DISTRAIN or Modern Banking) the numbers of reports required are 209 and 137 respectively. Besides defining tables of results, reporting structures include graphs and charts. Options include using accounting convention for negative numbers and changing the number and forms of reports that are produced automatically as the simulation progresses.

The Decision Entry Engine

This allows for different decisions to be used for different periods (to allow for an evolving challenge and maintain workload). It provides extensive decision checking (to identify illegal, unusual and sophistic decisions) and ensures mistakes are not made. Like the Reporting Engine, the templates produced by the Decision Entry Engine are defined in the Reporting Database. These templates are pre-populated with the previous decisions (from the Parameter Database) changes are made by the user and the decisions rewritten to the Parameter Database.

The Hypertext Help Engine

This proves context sensitive hypertext help with using the software, the current task and explanations of report lines and decisions. Besides textual help it is useful to provide pictures, diagrams, sound and music.

The Display Engine

This takes output from the Reporting, Decision Entry and Hypertext Help Engines and formats them for display to the screen or for printing.

The Communications Engine

This allows the simulation to communicate over an inter or intranet with other simulations and with an LMS.

The Text File

This contains the standard texts required by the shell. This allows the simulation to be *localised* to different countries with a different text file for each language.

The Constants File

This contains the references to standard text and shell specific help.

The Help File

This consists of hypertext linked help pages. A basic sub-set of these is the help pages associated with a specific shell (describing software use and providing error explanations). The rest are the help pages for the simulation (providing task help, explanations of terms and an on-line manual). This allows the simulation to be *localised* to different countries and versions with a different help file for each language and (if necessary) version.

Simulation Specific Components

These are the components that differ from simulation to simulation and consist of the following:

- **The Simulation Model**
- **Parameter File**
- **Reporting File**
- **Comment File**
- **Control File**
- **(Help File)**

The Simulation Model

This contains the logic that transforms decisions, parameters and data into business results. (The structure of the simulation model is described in Chapter 7 – Design Craft.)

The Parameter File

This defines the parameters and data used by the simulation model. Besides numeric data, this file contains the names of the variables and their format. Where it is necessary to *localise* the simulation to a different language, this file is replicated in the second language.

The Reporting File

This defines the reports and decision-entry templates available. Where it is necessary to *localise* the simulation to a different language, this file is replicated in the second language.

The Comment File
This defines the comments and parameters associated with them. Where it is necessary to *localise* the simulation to a different language, this file is replicated in the second language.

The Control File
This defines which files, decisions and reports are to be used by the simulation and the *flags* that control it. A separate Control File is required for each version of the simulation. The file contains references to the Text, Parameter, Reporting, Comments and Help Files and to flags that control the operation of the Simulation Manager. In particular, these flags control which decisions are made and results produced and their timing.

The Help File
This is the same file as described above but with additional pages that provide simulation specific information.

The Architecture and Modelling

To facilitate ***lean design*** the *architecture* must facilitate the creation of only the models necessary to fulfil market needs and allow this to be done on an incremental and agile basis. To speed, facilitate and support this incremental development process the shell employs:

a) The Parameter Database allows variables to be added to the model as needed and so they do not need to be predefined.
b) The Parameter Database documents the variables used by the simulation.
c) The Reporting Database defines reports and decision entry templates and allows these to be modified, augmented and restructured.
d) The Parameter Database that in association with the Reporting Database allows reports to be produced revealing how the models are *behaving* to help with the model's quality assurance and validation.
e) Built in design aids and utility programs.

The Architecture and Systems Dynamics

To improve the ***delivery process*** the architecture must facilitate economic calibration, ramped complexity, tutor interventions and provide different feedback styles. To facilitate and support the delivery process the shell employs:

a) The Control File together with the Parameter and Reporting Databases allow changes to the Economic Parameters as the simulation progresses.
b) The Control File together with the Reporting Database allows new reports and decisions to be introduced as the simulation progresses to allow complexity to be ramped.
c) The Simulation Manager together with special reports for the trainer provides tutor support information coupled with the ability to intervene using ad-hoc reports that can be provided to the learners to stimulate discussion and cognitive processing.
d) The Reporting and Parameter Databases to provide quantitative reports and the Comments Database and the Simulation Manager provide proactive and pre-planned qualitative feedback.

The Architecture and Tutor Support

To improve learning the architecture provides a system ***support for the trainer*** *and* ***the learners*** and this is done by the following architectural elements

a) Help is provided by the Help Database and Help Engine and the context for this help is defined in the Parameter, Comments and Reporting Databases and for the Simulation Manager and Display, Decision Entry and Reporting Engines by the Constants File.
b) Decision Screening is provided as part of the decision entry engine utilising logic in the model and data from the Comments Database.

c) Explanations are provided both as a separate group of reports and provided by the Display Engine using data from the Help, Parameter, Comments and Reporting Databases.
d) Comments are obtained from the Comments Database and based on outcomes of the simulation model are produced by the Simulation Manager and Reporting Engine.
e) The Tutor's Audit is provided as a separate group of reports accessed from the Simulation Manager.
f) The Team Commentaries are provided as a separate group of reports accessed from the Simulation Manager.

The Architecture and Versions

The Control File defines which decisions and reports are produced it is used to define a specific version. Although usually the other files are common to all versions of the simulation, it is possible to use different Text Files, Parameter, Comment, Reporting and Help Databases to facilitate different terminology and languages.

The Architecture and Standardisation

All my simulations use a common display engine and thus have a common look and feel. This helps branding but, more importantly, it means that once a trainer has experience using one simulation, he or she, does not have to spend time learning how to use another.

Figure 6.17 shows a typical desktop and it's components. At the top of the screen is the simulation name and version – in this case the simulation is Management Challenge and the version being used is the "Classic" version.

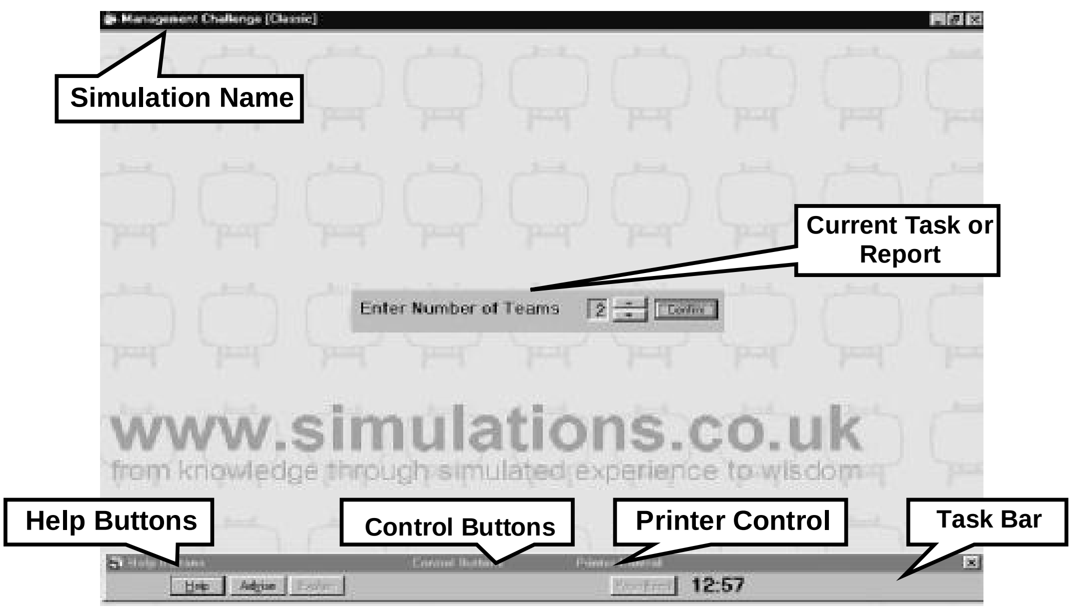


Figure 6.17: Typical Desktop

At the top left of the screen is the simulation name and version.

Displayed in the middle of the screen is the current task or report. Here, the trainer is about to enter the number of teams that will be competing against each other (Management Challenge like DISTRAIN is a total enterprise simulation where learners compete in teams against each other).

At the bottom of the screen is the Task Bar. To the left of the Task Bar is a cluster of help buttons. In the example the Help Button is active and if clicked a window will open providing help with software use. The Advice Button is also active and if clicked will provide help with the current task. The Explain Button is not active, but if it were, it would provide an explanation of the current menu item or decision entry field. Not shown, is a Manual Button that would be positioned to the left of the Help Button. If available, the Manual Button allows learners to access an on-line version of the learners' manual.

Towards the middle of the screen are the Control Buttons. As the current task just has a Confirm Button, this is associated with the object in the middle of the screen. But, normally, and as appropriate, Confirm or Continue and Abort Buttons are displayed. (The Abort Button is provided so that the trainer can *back out* of the current task is this is needed and possible.)

To the right of the Control Buttons are printer controls – a Print Button (not shown) and a Page Feed Button (shown but not active). The Print Button is only shown when it is appropriate to print.

To the right of the print buttons, the current time is displayed and, if appropriate, buttons to scroll the display to the left or right or up and down. In the same area, buttons can be provided to allow the learners to use an on screen calculator or spreadsheet and a word processor.

Behind everything is wallpaper that can look pretty and brand the simulation!

Simulation Novelty and Complexity

This section discusses simulation novelty and complexity and how these impact design difficulty, time and cost to help designers forecasting costs and the development schedule before starting the design and development work. This section uses the *case study* simulations to illustrate how novelty and complexity impact design.

The areas of design novelty and difficulty are described and discussed in the context of the six case study simulations. Following this the impacts on actual design difficulty and complexity are discussed and analysed for the simulations. Finally, a design time model is proposed to link model size with duration and novelty and design time with model size. In conclusion, this section suggests that design time can be forecast from simulation duration provided the forecast is adjusted to take into account simulation novelty.

Simulation Summary

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Duration	2 Hours	8 hours	9.5 hours	6 hours	16 hours	6 hours
Simulation Type	Concept	Total Ent.	Total Ent.	Planning	Functional	Process
Delivery Mode	Direct	Tutor	Tutor	Direct	Tutor	Direct

Figure 6.18: Simulation Summary

Duration

This is the amount of time taken to run the simulation. As discussed by Hall & Cox (1994) it affects the complexity of the simulation and the number of decisions possible. If there is a conflict between the required number and complexity of decisions and duration this will give rise to duration issues that must be addressed by the design of the simulation. As Figure 6.18 shows, the case study simulations range from a very short and simple simulation (Product Launch) to a long and complex simulation (SMITE).

Simulation Type

The simulations were classified based on the taxonomy described in Chapter 1 (Types of Simulation).

Delivery Mode

This is who will use the simulator (software) - the learners (direct) or the trainer (tutor). Who uses the simulation has a bearing on the learner and tutor support needs. Where the learners use the simulation directly, there is the need to incorporate a Learner Support System that provides help with software use, tasks and definitions of terms. Where only the trainer uses the software, the support needs may be substantially less. If the trainer has significant business knowledge, he or she will not need definitions of business terms. If the tutor is an experienced computer and simulation user, he or she will not require help with the software. The amount of business knowledge support depends (for Tutor Mediated simulations) on the tutor's business expertise and (for Direct Use simulations) the learners' business expertise.

Simulation Novelty and Design Issues

Novelty has two dimensions:

1. The novelty of the situation modelled.
2. The novelty of the simulation and learning process.

Novelty is measure from the viewpoint of the simulation designers. In other words, if the simulation designers are inexperienced then all simulations are novel to them.

Situational Novelty

This refers to the extent to which the designers have industry experience. If the designers have experience of the industry modelled then *situational novelty* will be low. But where the designers have no such experience, situational novelty will be high. The simulation exemplars have situational novelties range from low to average - none have a high situational complexity. However, an example of a high situational complexity is my Foundation Challenge simulation (that simulates the operation of a not-for-profit organisation). Here I had no experience of working in or with this sector. Likewise, although I have not worked in banking and so lack industry experience, I have been a bank customer for many years and have a very good idea of how a bank *should* be run.

A subset of situational novelty is that of *issue novelty.* This means that the extent to which the issues raised by the simulation are usual. This can be illustrated by DISTRAIN where promotional decisions and their implications were industry specific addressing specific industry issues.

Another aspect of situational novelty is the extent to which models from an existing simulation can be used. For DISTRAIN, about 70% of the simulation already existed and this reduced situational novelty substantially. For Modern Banking, models from a generic service simulation (Service Challenge) were used but these represented a smaller percentage of the model. For Prospector no models existed and all had to be created.

Situational novelty is also affected by the complexity of the simulation. Even if the designers are unfamiliar with the industry, a short simple simulation has a much lower situational novelty than a complex simulation. This is because a short, simple simulation will focus of a few, basic issues and so will not require extensive industry knowledge. In contrast, a complex simulation will focus on a wide range of issues, many of which are industry specific.

Cartoon 6.07 Situational Complexity

Simulator/Process Novelty

This is the extent to which the designers have had experience with the design of the particular form of simulation and the manner that it will be used. It has several, linked dimensions - the time dimension of the simulated business, the learning process and interactions between the learner(s) (and, if appropriate the trainer).

The most simulations involve running a whole or part business over several time periods (that may be years, quarters or months). Product Launch, Modern Banking, SMITE and DISTRAIN are examples of this form. Other simulations involved producing and evaluating a series of plans (SEED) or taking the learners through a business process (Prospector). Each of these involves different processes - both computer processing and learning.

The learning process also depends on whether the simulation is run in a single session or whether it will be run over several sessions and, when run in several sessions, how this is spread. With the exception of Prospector and SMITE, all the simulations were to be run in a single session. In contrast Prospector and SMITE were run in several sessions as the course theme. Another situation is where the simulation is used over a prolonged period with decisions every week or so. The difference has implications in terms of learners' ability to reflect (consciously and unconsciously) and the opportunity to intersperse and link the simulations to other learning activities.

The interactions between the learner(s) and the trainer involve the following:

1. The simulated business can be run by teams of business people or by individuals. Here all the simulations were designed for teams of business people who would share their knowledge and experience.
2. Who uses the simulation software (simulator) – either the learners entering their decisions directly or the trainer entering decisions on behalf of the learners. DISTRAIN, SMITE and Modern Banking had decisions entered by the trainer. All the remaining simulations involved the learners entering their decisions directly into their own simulator.

Simulator/process novelty is also affected by mismatches between desired duration and simulation complexity.

Key Novelty Areas and Simulation Characteristics

This suggested several key simulation characteristics and areas of novelty (Figure 6.19).

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Scenario Novelty	Low	Low	Low	Average	Average	Average
Issue Novelty	V. Low	Low	Average	Average	High	High
Duration Issues	None	None	Some	High	Some	Some
Delivery Mode Issues	Low	Low	Low	High	Some	Average
Simulator Type Novelty	V. Low	V. Low	V. Low	Low	Low	High
Learner Support Need	Average	None	None	High	Some	High
Tutor Support Need	Minor	High	Average	Minor	High	Minor
Train the Trainer Need	No	Yes	Yes	No	Yes	No
Versions Need	Several	One	One	One	Several	One

Figure 6.19: Novelty and Design Issue Characteristics

Scenario Novelty

This is the extent to which the business modelled might cause problems. Product Launch, DISTRAIN and Modern Banking all had scenarios that were easily described and modelled. The SEED and Prospector had scenarios that were substantially more novel. For SEED the scenario was novel because the product – the Cuddl-Etoy – was an imaginary one and the design had to involve inventing the marketing, operational and financial situations. In contrast, for Prospector, scenario novelty related to the (stage-gate) business process being modelled. The first version of SMITE had low scenario novelty as I had sales management experience for industrial products. But the second version of SMITE involved the sales management of a sales force selling consumer products to retail outlets and involved *three* classes of sales people (key account managers, sales representatives and merchandisers).

Issue Novelty

This is based on the number and type of business issues addressed by the simulation. For Product Launch the issues were those associated with the Product Life Cycle and as these are straightforward and well known, Issue Novelty was very low. Likewise, the issues facing learners in Modern Banking were well known and had been explored in similar simulations (for other industries). For DISTRAIN the client need was for the simulation to explore industry specific issues and this increased the Issue Novelty.

Although SEED involved the exploration of the wide range of issues associated with setting up a new business, these were well known and so its Issue Novelty was only average. For Prospector, Issue Novelty was high and because the issues were not known and would need to be defined during the design process. For SMITE, Issue Novelty was high because of the complexity with the *real-world* management of sales forces.

Duration Issues

These occur where there is a mismatch between the simulation's duration and its complexity (in terms of decision and model complexity). Where this is *none* there is no apparent mismatch between complexity and duration. Where this is *some* (DISTRAIN, SMITE and Prospector) they need to be considered but can be reduced by introducing decisions in stages during the simulation and reducing interaction granularity.

The SEED simulation had a major mismatch between the number of decisions and the required duration. To some extent, it was addressed by introducing decisions in stages and providing direct use to the software. (Direct Use of the simulator by individual teams of learners allows them to work *asynchronously* and this shortens the simulation.) However, for SEED, despite introducing decisions in stages and the direct use of the simulation, there was still a major mismatch between duration and decision complexity. This was overcome by reducing decision granularity significantly.

Delivery Mode Issues

This answers the question as to whether the manner of use and simulation type might cause problems that must be addressed during the design phase. It was felt that the Prospector simulation *might* have delivery mode (software use problems) and it was certain that the SEED simulation would have problems. For Prospector, it was felt that the complexity of the Stage-Gate process might cause Delivery Mode Issues. For SEED, as the learners would not have business knowledge this would be a problem that needed to be addressed with the timing of interactions and learner help.

Another Delivery Mode issue is who uses the software. For Tutor Mediated Simulations, the trainer can spend time beforehand becoming familiar and may use the software regularly. In contrast, for Direct Use simulations the learners who will have no prior knowledge of the software use the software. Delivery Mode Issues are higher for Direct Use (or Decision Support) simulations than for Tutor Mediated Simulations. Where the software is used remotely to the trainer, Delivery Mode Issues are even higher.

Finally, SMITE is Tutor Mediated and involves a large number of decisions. Because of this decision-entry is prolonged and to avoid learners being idle while the simulation was blended with other learning (to allow decisions to be processed while other learning was taking place). An alternative would be to provide each team with Decision Support software so that it is the learners who enter the decisions as part of their decision-making process.

Simulator Type Novelty

This looks at the novelty in terms of the structure of the simulation and the decision-making process. Simulations that involve the learners making a standard set of decisions that are used to simulate a period of time have a very low Simulator Type Novelty (as was the case for Product Launch and DISTRAIN (where decisions were made on a simulated month-by-month basis), Modern Banking (where decisions were made on a simulated year-by-year basis) and SMITE (where decisions were made on a simulated year-by-year basis). In contrast, Simulator Type Novelty was higher for SEED where, although, learners were developing a business plan, in order to emphasis the need to be *fast to market*, plans were made on a month-by-month basis. Thus the simulation was an amalgam of a planning and a normal period-based simulation. Prospector embodied another simulation form. Here time did not pass nor were several plans produced. Rather the learners were led recursively through a business process. This meant that for the Prospector simulation Type Novelty was high.

Learner Support Need

This depends on the type of simulation and the prior knowledge and experience of the learners. It is only relevant when the simulation is used directly by the learners (Delivery Mode - Direct). Were the software is used by the trainer (Delivery Mode – Tutor) online learner support is not needed, but it may be necessary to provide definitions of terminology and explanations of tasks and reports in the printed manual. Where the learners have reasonable business experience and knowledge or the concepts covered by the simulation are simple the need is average. However, where the learners have little business knowledge or experience (SEED) or where the simulation is complex (Prospector) learner support needs are high.

Tutor Support Need

This defines the amount of support required by the trainer and like the learner support need depends on the simulation type and the business and industry experience and knowledge of the trainer. Thus it is very relevant for simulations where the software is used by the trainer (Delivery Mode – Tutor). The two Total Enterprise simulations illustrate this need. DISTRAIN was to be used by experienced business people with substantial business and industry experience and knowledge. As a result the Tutor Support need was low and focused on using the software. In contrast, with Modern Banking, it was not possible to be sure that the trainers would have adequate knowledge and experience. This meant that is was necessary to incorporate comprehensive Tutor Support covering using the software and definitions of terms and explanations of reports. SMITE illustrates a third reason for comprehensive Tutor Support. SMITE's complexity meant that the trainer needed considerable on-line support when running it – support that allowed him or her to answer learners' questions quickly and authoratively.

For complex Direct Use simulations, where there is a need to have reports to assess and explain team performance, Tutor Support may be necessary. For BOSMAN (a very complex Operations Management simulation), after each simulated period, the current situational data was transferred to the trainer's computer for analysis. Here, besides the reports accessed by the learners other reports were provided to help the trainer manage learning and answer questions.

Train the Trainer Need

This depends on who will be running the simulation, their business and simulation use experience and simulation complexity. Where the simulation is simple or average complexity and the trainers are expected to be reasonably experienced, there is no need for a Train the Trainer course. If the simulation is only to be run by very experience trainers who have been involved in the design process a Train the Trainer course is not necessary. Product Launch is sufficiently simple for a Train the Trained course not to be necessary. Although the Modern Banking simulation is not especially complex, because the trainers might not have had the requisite business and simulation use experience a Train the Trainer course was necessary. Although the SEED and Prospector simulations were complex, trainers who were very experienced would use them and who were involved in their development and so a Train the Trainer course was unnecessary. For DISTRAIN, although the simulation would be used by experienced business people, a short Train the Trainer session was amalgamated with the pilot. This session helped adjust the simulation's calibration and extend the reports required to help the trainers manage learning and answer questions. The complexity and scope of SMITE meant that a Train the Trainer session was necessary.

Versions Need

This answers the question of whether the simulation need only exist in one form or whether it needs to have several versions to meet different training needs, audiences, markets etc. For example, Product Launch needed to be used in several ways (on courses, at business conferences, at schools and universities) and with different groups (managers, sales people, school pupils and MBAs) and so several versions had to be developed. For Modern Banking, budgetary restraints meant that only a single version could be developed, but at a later stage it is envisaged that other versions may need to be developed. The specialised focus of DISTRAIN, Prospector and SEED meant that it was not appropriate to consider multiple versions. As described earlier, SMITE was originally designed to for the Sales Management of Industrial Companies, many years later, a second version was created for a Consumer Products Company selling via Retailers.

Impact of Novelty and Issues on Design Complexity

The simulation novelty and issues identified above lead to several design issues and complexity as shown in figure 6.20 and discussed next.

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Decision Complexity	V. Low	Low	Average	High	V. High	High
Reporting Complexity	Low	Low	Low	High	V. High	High
Model Complexity	Low	Average	Average	High	V. High	High
Data Complexity	Low	Average	Average	High	V. High	High
Online Help Complexity	Average	High	Low	V. High	High	Average
Calibration Complexity	Average	Average	Average	High	V. High	High
Platform Availability	Complete	Complete	Complete	Most	Complete	Some

Figure 6.20: Probable Design Complexity Issues

Decision Complexity

This is based on the number, type and structure of decisions made by the learners. It ranged from three (for Product Launch) to several dozen (for SEED) and a fewer number but a complex structure (for Prospector). The number of decisions is highly correlated with Duration (Hall & Cox 1994). But, additionally it is also affected by Duration Issues. For example, as for SEED, there was a conflict between the number of decisions required and the duration, this led to *high* Decision Complexity.

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Decision Complexity	Low	Low	Low	High	High	High
Initial Decisions	3	16	7	See note	See note	See note
Final Decisions	3	16	16	See note	See note	See note

Figure 6.21: Decision Complexity and Numbers of Decisions

Figure 6.21 shows the total number of decisions made initially (at the start of the simulation) and the total number of decisions made at the end of the simulation (final), illustrating the extent to which new decisions are introduced into the simulation. Thus DISTRAIN initially (Quarter 1) involves seven decisions and by Quarter 5 the number of decisions had grown to sixteen. But with both Product Launch and Modern Banking, the number of decisions made during the simulation remained the same each period.

Initially, with SEED, all the decisions were available from the beginning. But the pilot showed that this was too difficult and so they were introduced in stages over the first four simulated months.

As SMITE involves managing up to twenty sales people across up to 100 sales areas, decision numbers change as more or less sales people are employed and depend on the extent to which the territory is penetrated. However, beyond this the number of decisions that can be made per sales person and the types of decisions made increase in number as the simulation progresses. Likewise the decision structure of Prospector is complex with the number of decisions depending on the number of projects being investigated and the number of types of decisions increases during the simulation. Thus for SEED, SMITE and Prospector the number of decisions made are partly under the control of the learners and partially predefined by the way decisions evolved during the simulation

Reporting Complexity
This is the complexity of the reports produced by the simulation and depends on the structure and type of reports and number of versions.

For these simulations the reports produced were categorised as follows:

- Business Reports for the learners
- Reports to help the Trainer explain calculations
- Reports to help the Trainer manage learning
- Reports to help with the design of the simulation models

Reports for learners require no explanation as they represent the normal outcome of the decisions. But the other report categories may need some explanation.

The reports explaining calculations serve two purposes. First, when using the simulation, they allow the trainer to answer questions about how results are calculated. Second, during the design stage, they help validate the models.

Reports to help managing learning include business analysis and information about responses to price etc. They allow the trainer to identify team strengths and weaknesses and, during the design stage, help validate and calibrate the models.

Finally, design-help reports are not used during use but are used during the design stage to check and validate the model and calibrate the simulation.

For Product Launch, Modern Banking and DISTRAIN the reports produced are normal financial reports, graphs and charts and so Reporting Complexity is low. In contrast, for Prospector and SEED the reporting structure and types were complex and this led to high Reporting Complexity. In particular, Prospector is a stochastic simulation and it was necessary to ensure that this randomness was appropriate. To reduce the duration of SEED *intelligence* was built into the simulation model and it was necessary to attune and check these models during the design. Thus for both Prospector and SEED, special reports were provided to help with the design. For SMITE the number of reports reflect the complexity of the simulation and the need to help the learners identify cause and effect and the tutor to identify and explain cause and effect.

Figure 6.22 shows the number of reports provided for each simulation. For DISTRAIN there is an apparent conflict between low complexity and the number of reports. This is because DISTRAIN was developed from an existing simulation and many reports existed that were not used by the new simulation.

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Reporting Complexity	Low	Low	Low	High	High	High
Number of Reports	57	137	209	133	215	112

Figure 6.22: Reporting Complexity and Numbers of Reports

Model Complexity
This is the size and structure of the Simulation Model. Besides producing the Business Reports for the learners, the model must produce outputs to show interim calculations, analyse business performance, to help with calibration and design validation. This correlates with the Duration and Simulator Type and is affected by Scenario Novelty and Issue Novelty and, possible, the number of Versions needed and Duration Issues. A further issue is the extent to which existing models can be used. Except for DISTRAIN, the extent of existing model use was negligible. (For DISTRAIN, it was possible to use an existing simulation (Distribution Challenge) to provide approximately 70% of the simulation model). Thus although DISTRAIN has an average relative model size, the additional complexity was low. For SEED the Model Complexity was high because to simplify and hence shorten the simulation *intelligence* was built into the model. Figure 6.23 provides a comparison of Model Complexity and the relative sizes of the models.

(Note the apparent inconsistency for DISTRAIN is because the model size shown is for the whole simulation. In fact the *additions* to the model were of a similar complexity to that of the whole Product Launch model.) The Relative Model size metric is not available for the SMITE simulation.

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Model Complexity	Low	Average	Low	High	V.High	High
Relative Model Size	27%	78%	87%	137%	N/A	169%

Figure 6.23: Relative Model Size.

Data Complexity

This is the size and structure of the database used to store the parameter and reporting data. It correlates with Model Complexity and is affected by Simulator Type, Issue Novelty and Decision and Reporting Complexity. Figure 6.24 compares the number of variables used by the simulation, for decision entry and reporting. A significant number of variables are vectors and so one variable might be price would exist as a vector containing the prices for each market. This is especially true for SMITE as vectors included parameters and data for some one hundred market sectors and other vectors included data for more than a score sales people. As the number of variables does not include variables used for interim calculations, the actual number of variables used by the simulations is higher than shown.

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Data Complexity	Low	Average	Average	High	High	High
Number of Variables	100	537	476	534	513	588

Figure 6.24: Relative Parameter Size.

Online Help Complexity

This relates to the size of the online help database in terms of the number of help screens. The help screens serve several purposes – help with the use of the software, help with the current business task, report explanations and definition of business terms.

Where several simulations are developed, help with the software can be standardised and the size of the help database just depends on the additional Learner and Tutor Support Needs and this depends on the experience and knowledge of the users and the complexity of the simulation.

The two similarly complex Tutor Mediated simulations (Modern Banking and DISTRAIN) illustrate extreme tutor support needs. DISTRAIN was to be used by experienced business people after training and this meant that the help database was limited to just providing help with software use. In contrast, because of uncertainty about the trainers' knowledge, Modern Banking had a large help database (that explained tasks and reports and defined terms). Finally, the complexity of SMITE necessitated a large help database.

Where the simulation is used directly by the learners (Product Launch, SEED and Prospector) the online help need is greater. For SEED the lack of learner business knowledge meant that besides definition of terms and help with tasks, the simulation had to provide business knowledge. Product Launch's short duration and the business experience of Prospector's learners meant that their help databases were of average size.

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Online Help Complexity	Average	High	Low	V. High	High	Average
Help Screen Numbers	254	554	166	668	497	261

Figure 6.25: Help Database Size.

Calibration Complexity

Gold (2003) states, *"The business environment is a highly interactive system"* and (for simulations) *"Complex feedback loops exist and may give rise to unexpected results"*. Thus the risk associated with the dynamic behaviour of the simulation must be considered during design and affects the Calibration Complexity.

Product Launch, Modern Banking and DISTRAIN had a similar average level of calibration complexity. Both SEED and Prospector had a significantly higher level of calibration complexity. For SEED there were three entrepreneurial opportunities and none of these could obviously be the best. Further, for each cash limitations had to be challenging while there was a reasonable chance of being profitable. For Prospector there was the need to randomly generate a representative range of projects with different sizes, urgency etc. Further, it had to be possible for the learners to narrow these down to half a dozen or so in a reasonable time. Thus both SEED and Prospector had a high level of Calibration Complexity. SMITE has a very high level of calibration complexity as besides having some hundred separate sales areas (markets) there could be several dozen different sales people each of whom differed in terms of skills and whose morale would be influenced by team decisions. Yet, overall, it is necessary for teams to make reasonable profits (neither exceptionally low or exceptionally high).

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Calibration Complexity	Average	Average	Average	High	V.High	High

Figure 6.26: Calibration Complexity

Platform Availability

This relates to the extent to which an existing simulation platform could be used for the new simulation. Product Launch, Modern Banking, SMITE and DISTRAIN all used existing platforms and no additional code was required. For SEED it was necessary to provide a new decision entry format. However, this only added about a day to the development process. In contrast, Prospector required a complete new Simulation Manager to be developed and this increased code by 25% and the design time significantly. Even so, even for Prospector, existing platform objects accounted for nearly 80% of the total code.

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Platform Availability	Complete	Complete	Complete	Most	Complete	Some

Figure 6.27: Platform Availability.

Overall Novelty

It was felt that Product Launch, Modern Banking and DISTRAIN had an average level of novelty and design difficulty. But SEED, SMITE and Prospector had a very high level of novelty.

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	SMITE	Prospector
Relative Novelty	Average	Average	Average	V. High	V. High	V. High

Figure 6.28: Novelty and Design Difficulty

Forecasting Design Time

The analysis of novelty and complexity helped identify the qualitative risk and development needs of business simulations but ultimately there is the need to try to quantify the effect on design time.

When assessing software project durations, the expected program size is frequently used when planning, tracking and estimating software projects. (Park, 1992). Design time is forecasted by dividing the expected number of program lines by the expected number lines that can be programmed a day (Figure 6.29).

Duration ê Model Size ¸ Lines/Day ê Design Time

Figure 6.29: Design Time Forecast Model

As described earlier, it is reasonably to expect that the novelty and design issues described will directly impact the size of the simulation model. Hall & Cox (1994) showed that complexity was strongly correlated with duration and Keyes (1977) suggested that complexity is also linked to model size. So simulation duration and model size would be correlated and the size of the simulation model forecasted from the desired duration coupled with an assessment of the effect of novelty.

Duration and Model Size

The correlation between duration and model size was analysed using a group of eighteen simulations that were reasonably homogenous in terms of novelty and design issues. Besides homogeneous novelty and design issues, to minimise the impact of coding style, the eighteen simulations were all designed by me as were the case study simulations (but did not include these). Also to eliminate differences in coding style between the model and the platform only the lines of simulation model code were used in the analysis. However, the simulations analysed did have a significant range of durations (from 3 to 20 hours).

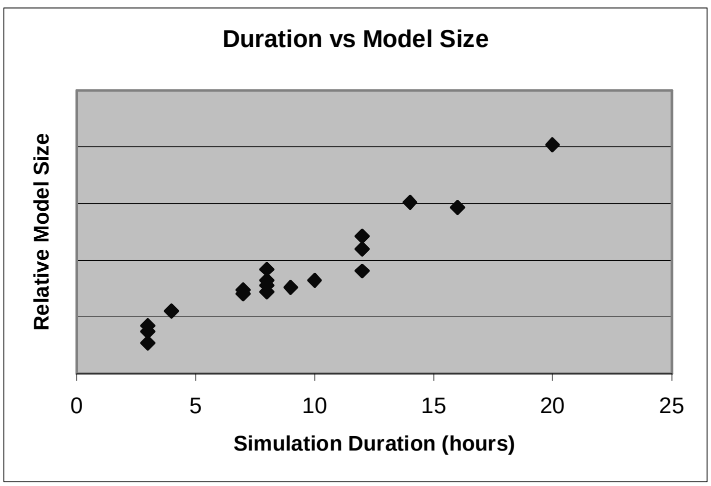


Figure 6.30: Graph linking Duration with Model Size

For the eighteen homogenous simulations (Figure 6.30) it was found that model size correlated strongly with simulation duration (Figures 30) ($p<.01$).

Model Size and Novelty

The regression model was derived from group of simulations with an average level of novelty. The regression model was used to forecast model size for the simulations and compared with the actual size to give an indication of the relative impact of novelty (figures 30 and 31). (In figure 6.31, the eighteen homogeneous simulations are represented by diamonds and the case study simulations (excepting SMITE) by squares.)

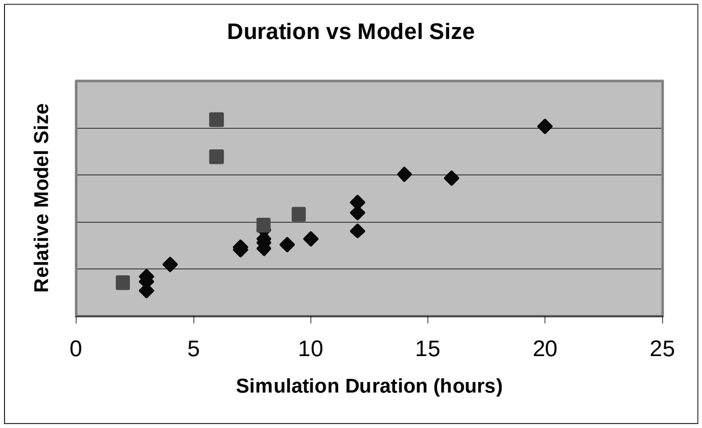


Figure 6.31: Duration/Model Size Graph including new simulations.

Simulation	Product Launch	Modern Banking	DISTRAIN	SEED	Prospector
Actual vs. Forecast %	142%	122%	116%	277%	341%

Figure 6.32: Regression Forecasts Actual vs. Forecast Model Sizes

The qualitative analysis of the novelty and complexity allowed one to determine whether the simulation had a normal (average) level of novelty and complexity or a high level of novelty. If novelty is average, duration can be used to forecast model size. But if the simulation is novel, the model size is likely to be significantly larger than that forecast from duration. This is illustrated in Figure 6.32 where Product Launch, Modern Banking and DISTRAIN all had an average level of novelty and their actual model size was close to forecast model size. But, if the simulation is very novel or complex as was the case for SEED and Prospector this lead to a significantly larger model size and longer development times.

The originally SEED duration based on its decision complexity was to be 12 hours and had this been the case, the model would have been 87% larger. In the event the model was 177% larger than expected and this suggests that the need to handle the mismatch between required duration and actual duration added this additional complexity to the model.

The deviation between actual model size and a forecast based on simulation duration for the Prospector simulation was even greater with an actual model size nearly three and a half times larger than the forecast.

Program Lines/Day

Having forecasted model size, if a consistent number of model lines are programmed each day then design time can be forecast from duration and probable novelty. However, model (program) lines/day depends on several factors:

Language Used
Type of Code
Programming Style
Physical vs. Logical Lines
Design Experience
Model Size

Language Used

The language used has a considerable impact on productivity and the established measured of language productivity is *function points* (Albrecht and Gaffney 1983). For a language, the higher the Function Points number the more lines of code required to program a particular application. For example Visual Basic has a median level of function points of 42 and C a median of 104 (QSM 2005) indicating that Visual Basic is more than twice as efficient as C.

Type of Code

The original function point analysis (Albrecht 1979) separated code into five classes - External Inputs, External Outputs, External Inquiry, Internal Logical Files and External Interface Files and hence was heavily biased to data processing (Somerville, 1992). But Jones (1986) extended these classes to include algorithms. This separation recognises that different classes of code have differing impact on programming time. Here, the separation of the simulation code into two parts – the platform and the model – means that the types of code are also separated and, the model code is practically completely algorithmic.

Programming Style

The number of lines of code will also depend on the skills of the programmers and their programming style. For example figures 6.33a and 6.33b show two price algorithms that are logically the same but involve different numbers of lines.

```
If price>minimum_price Then price_response=(nominal_price/price)^price_sensitivity
```

Figure 6.33a: Single Line Price Response Algorithm

```
If price>minimum_price Then
   price_response=(nominal_price/price)^price_sensitivity
End If
```

Figure 6.33b: Multiple Line Price Response Algorithm

Physical vs. Logical Lines

Figure 6.33 shows another aspect of measuring program size – the difference between physical lines of code and logical lines of code. Although Figure 6.33a is only one line of code it contains two logical elements (the if-then conditional and the calculation). Likewise Figure 6.33b is three lines of code but still has the same two logical elements. For the analysis of model size, physical source lines of code were used and the simulation designer's programming style was consistently that of Figure 6.33b.

Design Experience

A further factor is the design experience of the designers. Here the simulations used to investigate the correlation between model size and design time were all developed after I had designed a significant number of simulations and was experienced. For inexperienced designers, the number of program lines per day will be less and will change as the designers gain experience.

Model Size

Longstreet (2004) suggests that programming time has non-linear relationship with software size and as software increases in size, programming time increases disproportionately. However, Boehm's COCOMO model (1981) suggests low exponents (1.05 through 1.2) depending on the class of projects. Sommerville (1992) suggests that "particularly for smaller systems" the effort curves are almost straight lines. For simulations, the model consists of a large number of sub-models (algorithms) each of which have a similar level of structural complexity. Thus a simulation model consists of an aggregation of the sub-models and this means that design time is built up by summing the time to develop each sub-model and so is inherently linear.

Model Size and Design Time

The final stage in the analysis was to correlate design time with model size. Except for Product Launch and SMITE, design time data was available for the simulations together with the design time for another simulation (Foundation Challenge). But, the design time data for DISTRAIN was the time required to modify the Distribution Challenge simulation rather than the complete design time. So for DISTRAIN only the additional model lines were used in the analysis.

The design time was the total time to take the simulation through the last four stages of my design methodology (Simulation Design, Simulator Development, Simulation Validation and Design Finalization). Figure 6.34 shows the relationship between model size (program lines) and design time and, despite the small number of samples, model size was highly correlated with design time (p<.01).

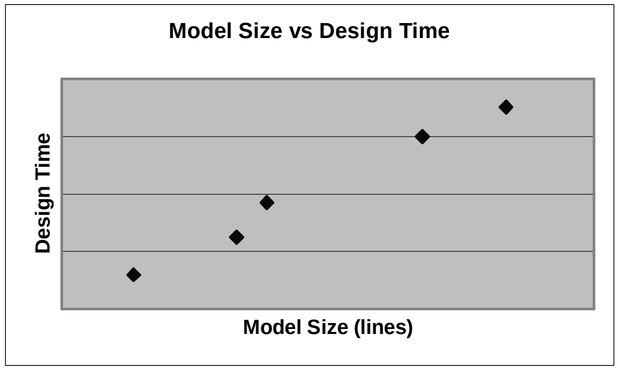


Figure 6.34: Graph linking Model Size and Design Time

Platform Availability and Design Time

If a simulation platform does not exist then there is a need to create the software that manages the process of entering decisions, presenting results, reading and writing data from files etc. However, when creating a complete simulation from scratch, platform functionality can be substantially less than a general-purpose platform and this should reduce software size. Also software size can be reduced further if it is possible to use existing objects. Further, as the design complexity is likely to be less for the platform components, productivity in terms of lines/day is likely to be higher than for the model. Finally, the time taken to design a custom platform has a sizeable fixed element and the only variable elements are likely to be associated with decision entry and report presentation. This means that the time to create the platform does is not proportional to duration (as is the case for models with an average level of novelty).

Impact of Novelty and Complexity on Design Time

The impact of novelty on model size leads to the suggestion that for a simulation with average novelty the designers will draw on their experience and development time is directly correlated to the duration of the simulated experience. However, where the simulation is novel the designers must learn to overcome the novelty and this will lead to a significantly larger model and longer development times.

However, besides model size and novelty, the forecast must also take into account whether a simulation platform must be developed or modified, the computer language used and the programming style and experience of the designer. As all simulations are likely to be novel for inexperienced designers, the analysis suggests that they may take much, perhaps many times longer to design a simulation than where the designers have had experience designing similar simulations.

In summary the design time forecasting model is thus

1. **Forecast model size based on duration**
2. **Adjust model size for novelty**
3. **Divide model size by lines/day to forecast model design time**
4. **(Plus custom platform design or platform customisation)**

Project Management

I do not intend to discuss project management in great detail. Rather I will raise and discuss several issues that I believe impact project success. These are:

- **Scoping**
- **The Resources**
- **Communication**

Scoping

Probably the most important stages in simulation design are the first two stages in my Rock Pool Method (as described in Chapter 8) – *Needs Definition* and *Simulation Specification*.

Needs Definition

This defines learning objectives, the target audience, duration and manner of use. Defining the **learning objectives** goes beyond defining the hard and soft knowledge to be explored to include motivational/engagement needs, assessment and evaluation and how the simulation will enhance learning. Beyond defining learning objectives defining *all* of these are important. In the context of the **target audience** of learners, understanding their starting level (prior learning, group diversity and maturity) so that the simulation does not under or overwhelm them (cognitively and emotionally). The business knowledge and experience of the trainer *managing learning* is equally vital. Running a simulation is not the same as lecturing using Power Point – it is student centred and involved the trainer coaching and challenging the learners – a role that requires the trainer to have business experience and the ability to lead and manage people. Just as the relationship of the simulation to the audience can under or overwhelm, the **duration** can also do so. In my experience, clients commonly under estimate the amount of time that a simulation must take and ask for too much to be included. The problem here is that learners do not have time to reflect and formulate concepts – thus this leads to a *Short-Circuited* Experiential (Figure 6.10 earlier). Finally, **manner of use** has a bearing. One aspect is whether the simulation is run in a single session or multiple sessions. When run in multiple sessions the learners can reflect and discuss in between the sessions and this *free* time allows a more complex simulation to be delivered in the budgeted session time. A second aspect is where the simulation is spread over a course and linked with it. Here a clear understanding of the course schedule is needed so that both coverage and timing can be planned into the design.

Simulation Specification

This defines the business issues to be explored, the business scenario, delivery mode and whether several versions are required. Probably the most important specification element are the **business issues** that should be analysed and explored by the simulation as these are the elements that ensure relevant in-depth discussion about things that are important to the learners and their organisation. Next is the **scenario.** Obviously this must encapsulate the financial operational, marketing and technical structure of the learners' organisation. But the extent to which this is stylised and simplified is crucial relative to the Needs Definition (particularly duration, learners and learning objectives). On several occasions I have merely changed the names of resources and markets to represent a totally different business and have found that it is sufficient to ensure engagement and still provide excellent learning! Both the **delivery mode** (Tutor Mediated or Direct Use) and **number of versions** are usually obvious.

The Resources

Developing a new simulation will involve several resources (Figure 6.35)

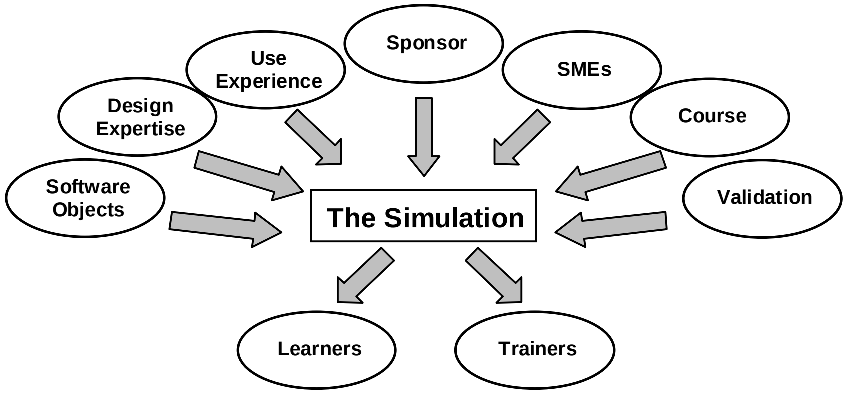


Figure 6.35: Parties and Tasks

Software Objects and Platform

Here we are looking at existing software objects and platform availability as these can reduce design time very significantly (Hall, 2003) and ensure quality (Hall, 1996b). The time saving and quality assurance occurs because the software was developed and tested for a previous project. Beyond, the computer software, time can be saved by using existing documentation – specifically, how to use the software, how to run simulations and elements of the learners' manual, briefing slides and tutor's manual. In terms of the learners' manual, briefing slides and tutor's manual it is useful to use a standard format (discussed in Chapter 7). Based on a recent analysis of average training development times (Chapman 2010) compared to my firm's metrics using a software platform and existing software objects reduces development between five and tenfold.

Design Expertise

Here we a looking at the experience the designer has in developing business simulations of this form, for this industry and used in this manner. The impact on development times was discussed earlier in the section on simulation novelty and complexity. Additionally, there is the impact on quality and this is impacted further by the use of an appropriate development methodology (see the Rock Pool Method in Chapter 8).

Use Experience

Business Simulations are rather like automobiles in that neither can be designed without testing *on-the-road.* Here we are looking at the *process* knowledge aspects of the design (discussed in Chapter 4) and discussed in Chapter 8 in the context of soundness testing.

Sponsor (Client)

This is the person/organisation that is commissioning and paying for the design. With the client there are several concerns - too much content is included relative to the time available - including things that are *real* but are not part of the learning needs set and adding things as the design proceeds (design creep). Ideally, you need to agree the Simulation Needs and parts of the Simulation Specification (scoping earlier) and then the client should bow out and leave you to work with the SMEs and Trainers.

In my experience there are two types of customer – those that let you get on with the design and those who interfere and continuously change their needs. For several years I worked with one client who asked for regular builds that then stimulated new ideas that he wished to incorporate. This was Ok when I charged on a per diem basis. But then I made the mistake of accepting his request for a fixed price project. I tried to limit my risk by stating clearly (as I thought) contract scope. Unfortunately, he ignored this and eventually I terminated the contract as I was spending an unreasonable amount of time responding to his changes and as a consequence was working at a loss with no likelihood of ever finishing the design. Having learnt from that, recently I walked away from bidding for a project because during the initial discussion it was obvious that the prospect thought that he knew about simulation design when he did not. Worse, he had a confrontational arrogant attitude – a situation that would mean that I would not make money and he would not get a simulation that he was happy with.

Subject Matter Experts (SMEs)
These are the people who provide the *content* knowledge and this is different from the *process* knowledge (based on Use Experience). To an extent the problems with SMEs are the same as those with the client (too much content, irrelevance and design creep). Besides subject matter (knowledge content) you need to explore the financial, market and operational structures of the business to be replicated and the *issues* to be explored (Simulation Specification – scooping earlier). Having obtained this information, it is attractive if the SMEs are not too involved with the design process, for if they are this leads to design creep and an over complex simulation. (When designing SEED I had a single meeting with the academics in Imperial's Entrepreneurship Centre and this was sufficient for me to create the simulation.

Course
Here we are looking at how the business simulation integrates with and supports other learning. (This is discussed in Chapter 4 – Design for Process). Having designed courses while working for Honeywell and Ashridge, I feel that not only should the simulation be integrated with the course but where different parts of the course are presented by different people these need to understand each other's contribution and how these link together. Chapter 9 (Ways to Use Business Simulations) explores the ways business simulations can be used, why they are used this way and the issues.

Validation
Here we are looking at how the business simulation will be validated in terms pf learning and engagement. As such it will involve feedback from the sponsor, the subject matter experts, the trainers using the simulation and, especially, the (adult) learners. Chapter 8 (Design for Quality) has sections exploring soundness testing and learning soundness.

Learners
Here we are looking at prior learning (discussed earlier), the mix of experience and maturity. The needs of adult learners are discussed in Chapter 2 (Learning and Simulation) and simulation focus and prior learning needs are discussed earlier in this chapter.

Trainers
There are the people who will be using the business simulation. As described in Chapter 10 (Tutoring Business Simulations) the trainers need two core skills – the ability to coach and challenge in an active, learner centred environment and deep business experience and knowledge. As all simulations are likely to have financial metrics a familiarity with basic financial concepts is an advantage. As described earlier in the sections on *Tutor Support Need* and *Train the Trainer Need* the trainer's business experience and knowledge will impact the need for help and a train-the-trainer course.

Communication

As the project moves into the Simulation Design stage of my Rock Pool Method, I find it important to communicate with the customer about the simulation so as to *lock down* the design before making changes become expensive. I do this using

- **Briefing Slides**
- **The Decisions**
- **The Learners' Brief**

Briefing Slides
These provide an easy way of summarising how the simulation will work, key information about the situation (scenario) modelled, the decisions and results. In effect it bullet points the simulation and provides a *road map.*

The Decisions
These need to be clearly linked back to the learning objectives and the business issues explored and are constrained by the duration, learners' prior knowledge and the trainer's knowledge.

The Learners' Brief
At first glance this is the best medium to communicate the design but in practice it is not. First at this stage it is an incomplete, preliminary draft and second, in my experience, the client will not read it carefully enough. (When I customised my Executive Challenge simulation, the client worked with me designing the Learners' Brief and this worked well).

During the **Simulation Development** I attempt to avoid communication with the customer unless absolutely necessary as this disrupts the design process and encourages the client to make changes. Changes that are expensive and dangerous - expensive because they involve going back and modifying completed models and dangerous because these changes can interact with existing and tested parts of the simulation.

Obviously when **Validating the Simulation** the client must be fully involved. But here it is important that he or she do not attempt to make major changes at this point.

Customising and Modifying Simulations

This section explores the issues associated with customising, modifying or updating a simulation. Initially, the customisation history of a simulation is explored over two decades. Following this a customisation *hierarchy* is described and, finally, an actual customisation is used to explore the issues associated with customisation.

An Example - Management Challenge

This shows how one simulation (Management Challenge) was used as the basis for several totally new simulations and several different versions. The use of an existing simulation had several advantages. Basing a new simulation on an existing simulation has the advantage that the client is better able to articulate his or her needs. This is exemplified by on customisation (Gambro Challenge) where the client actively defined his needs by rewriting the Learner's Brief after experiencing a run of Executive Challenge. Even if the client is not as proactive, it is quick and easy to take the existing simulation's Learner's Manual and modify it to suggest changes and use this to communicate with the client. A second advantage, the amount of simulation development work is lessened as a significant number of models, decision and results already exist and are fully checked.

The Management Challenge simulation was developed in 1986 for use as part of a promotional contest. However, as I had considerable design freedom, besides the original (promotional contest purpose), I designed the simulation to be used on Business Acumen, Business Appreciation and Financial Appreciation courses. Since then there have been three major customisations and several minor ones (figure 6.36).

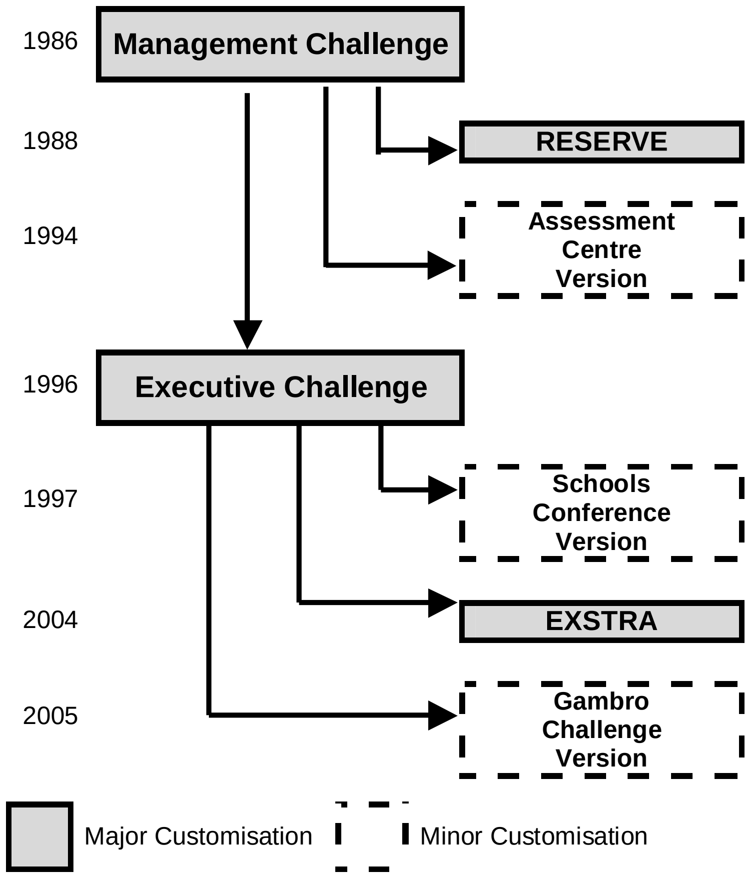


Figure 6.36: Evolution of Management Challenge

In 1988 a completely new simulation (RESERVE) was developed by adding a series of Research and Development decisions and results to Management Challenge. RESERVE was designed to explore how research and development influences the commercial success of a business. This customisation involved creating new models that linked R & D effort to improved product performance (competitive advantage), product quality (reduced scrap and rework cost), material use (reduced production cost) and better processes (more efficient production). The changes practically doubled the number of decisions and introduced a significant number of new reports. However, despite this the customisation only took a few weeks.

In 1994 a version of Management Challenge was created for use for assessment - Assessment/Development Centres. This took a matter of hours, as all that was required was to add a set of reports for the assessors. These reports highlighted and analysed team performance and identified strengths and weaknesses. In doing so they enabled the assessors to concentrate on assessing the managers rather than analysing the results. The changes took less than half a day as all that was required was to provide a set of reports that summarised on a team by team basis their results, key performance measures, business analysis reports and listed strengths and weaknesses. The short customisation time was possible because all the data for the reports existed in the database.

In 1996 I was asked to create a simulation for a pharmaceutical company (Ciba Geigy). This involved extending Management Challenge from two products to a three products and adding a new decision (product strength) that allowed learners to *position* their products and ensure the best value proposition for each market sector. This customisation resulted in a new simulation – Executive Challenge. A simulation that then provided the basis for several further customisations. The addition of the new decision (product strength) was eased because I already had a model that I had used in another simulation (Technique). Adding a third product involved modifying the database. The longest part of the customisation was *calibration* where the changes to the simulation's parameters were adjusted to ensure that it would be just possible for learners to run a profitable company that survived.

In 1997 I was asked to provide a simulation to be used at a school conference for sixteen and seventeen year olds. The purpose of the conference was to enthuse the children about a career in industry and consisted of a series of presentations by business people and practical activities. The simulation was to run throughout the conference acting as a theme. Because the children had no business or financial knowledge and experience, it was necessary to simplify Executive Challenge. In a couple of hours the reports produced and the decisions made were simplified. The modified simulation was still in use for ten years later. Changing the reports involved changing the reporting database and this took less than an hour.

In 2004 the reverse occurred with a client (Rockwell) requiring a simulation to challenge high-flying junior managers in a reasonable time period. This involved modifying Executive Challenge to focus on strategic issues - adding additional decisions covering funding, promotion, process and product improvement and business research. These changes involved extending the possible range of products from three to eight and required additional models and reports. It resulted in a new simulation - EXSTRA (EXecutive STRAtegy simulation).

Finally, in 2005 a major renal equipment provider (Gambro AB) required a simulation that would reflect the issues faced by their senior management – as described in the learner's manual to allow *"an exploration of how our business might function under YOUR leadership".* The customisation involved changing terminology, a few minor changes to the simulation model and recalibrating the simulation and this took three days.

The customisations described above took from a couple of hours to a few of weeks. The most rapid customisations were possible because the simulations were driven from databases and hence all that was required was changing and editing these. The more extensive customisations involved adding models and decisions and recalibrating the simulation. This was aided by driving the simulation from databases but more important was the extent to which the model was documented. Documentation of the model was and is of particular importance because of the long intervals between the customisations.

Customisation Hierarchy

There are several levels of customisation (Figure 6.37) ranging in complexity and cost from just changing terminology to the addition of new models.

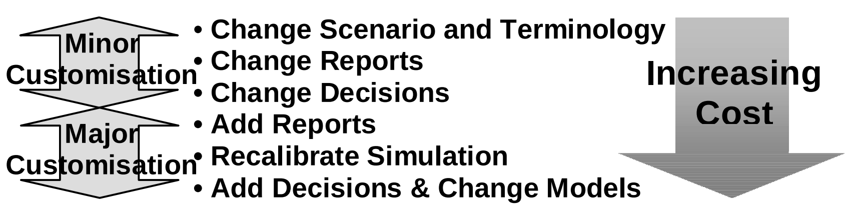


Figure 6.37: Customisation Hierarchy

Change Scenario and Terminology

The simplest and quickest way of customising a simulation is to change the scenario and/or terminology. Besides allowing you to provide the simulation in different languages it may be necessary to change financial terminology. For instance, in Britain money owed by customers is called *Debtors* but in the US it is called *Accounts Receivable* (Figure 6.38).

UK Balance Sheet	US Balance Sheet
Assets	
Fixed Assets	Fixed Assets
Stocks	Inventory
Debtors	Accounts Receivable
Cash	Cash
Equity & Liabilities	
Issued Shares	Issued Stocks
Long Term Loans	Long Term Debt
Creditors	Accounts Payable
Overdrafts	Short Term Debt

Figure 6.38: Examples of Terminology Differences

It is helpful if the business scenario relates to the learner's industry. Although this may require significant changes to the simulation model and it's calibration, this is not always necessary. By changing product, market and decision names, it can be possible to align the simulation to the learners' industry. Some years ago, a training provider asked me if I had a simulation suitable for the senior management of a Carpet Distributor. On talking with the provider, I felt by changing the names of the product ranges and altering the staff numbers decision to transportation, I could transform my Retail Challenge simulation into a simulation that met his and his client's needs. This was done in a few minutes and the simulation was used very successfully – to the extent that the Managing Director (CEO) of a subsidiary said to me *"it must have taken you a very long time to replicate our business so accurately"!*

Change Reports

This involves changing the timing, range and format of reports produced. If the simulation is driven by databases and the data for the reports exist in the parameter database this can be done quickly. An example is where all the middle management of a financial institution used a simulation on groups in a team-building course. Because of the size of the company this would require several dozen courses spread over two years. About halfway through the run of courses, the focus of the business changed and this made the simulation less relevant. However, by re-sequencing the decisions and the reports, it was possible to re-align the simulation – a change that took about half a day and allowed the simulation to continue to be relevant and deliver learning.

Change Decisions

This involves changing, removing or delaying existing decisions. For example, when developing Executive Challenge from Management Challenge, the additional product range and product strength decision added complexity of the simulation and hence would lengthen its duration. To correct this the Accounts Receivable (Debtor Days) decision was removed. When simplifying the Executive Challenge for the schools conference, instead of allowing learners to redesign the products every period, learners could only do this once. If decision timing and structure is driven from the database, eliminating decisions or changing their timing can be done quickly in an hour or so.

Add Reports

This involves adding reports where the data already exists in the database and where it is not necessary to add to the simulation model. For example, when creating the version of the Management Challenge simulation to build the financial understanding of non-financial managers additional reports were created that allowed learners to check underlying financial calculations (inventory valuation, cash flow, manufacturing costs etc.). Another example of this was the Assessment Centre version of Management Challenge, where reports were created to be given to the assessors to provide pre-digested information about the problems and issues facing the assesses.

Recalibrate Simulation

This involves taking the basic simulation and changing parameters to replicate the learners' industrial situation. This was done when creating the Gambro Challenge. It is a process that involves changing the parameters and then testing the impact by running the simulation several times. A general need for a Total Enterprise simulation is to balance the ability to be profitable against funding (cash flow). Calibration is a heuristic process and so can take a considerable time – possibly days. (This process can be aided by adding reports that help identify sources of profitability and uses of cash and thus track the financial impact of the changes.)

Add Decisions and Change Models

This is the most time consuming and costly. The time and cost depends on the extent to which the simulation models must be changed. For example, the Executive Challenge model is only 1% larger than the Management Challenge simulation model and the Gambro Challenge simulation 7% larger than the Executive Challenge simulation. But, the EXSTRA simulation model is 70% greater than the Executive Challenge simulation. Both Executive Challenge and Gambro Challenge took a couple of days to customise but EXSTRA took much longer.

DISTRAIN – a customisation case study

My Distribution Challenge simulation was customised in 2004 for Schneider Electric/Square D to:

- **Improve sales engineers "Business to Business" knowledge**
- **Replicate a Schneider Electric Distributor (Customer)**
- **Be run by Schneider Electric staff**
- **Provide Active Learning**
- **Last no more than a day**

The customisation involved adding new decisions, associated models and reports and recalibrating the simulation to replicate the business situation facing a Schneider Electric distributor.

In contrast to the earlier section on Management Challenge that illustrates the strategic benefits of customisation, this explore the practical details of customisation.

Changing the Decisions

The original simulation (Distribution Challenge) involves making five different types of decisions, with the new simulation adding seven new decisions (Figure 6.39).

Original Decisions	New Decisions
Percent Mark-up	Training Days
Inventory Purchases	Number of Products
Advertising	Receivable Days
Transportation	Electronic Linkage
Payables Days	Demo Equipment
	Demo Room
	Small Project Initiative

Figure 6.39: Changes to Distribution Challenge Decisions

Advertising was renamed Marketing and Transportation renamed Staff Numbers but for neither were the models changed. Payable Days was removed as a decision. New decisions were added to explore the issues associated with staff quality (training days), impact of product offering (number of products), customers on cash flow (receivable days), process improvement (electronic linkage) and market focus (demo equipment, demo room and small project initiative).

Adding Reports

The introduction of new decisions necessitated adding 74 new reports to be provided to the teams and the trainer. For the teams the reports showed the outcomes of the new decisions and provided *clues* to the way decisions affected them. For the trainer the new reports explicitly revealed causal links and measured the effectiveness and efficiency of the decisions made.

Decision Introduction

As increasing the number of decisions would lengthen the simulation, it was necessary to introduce the decisions progressively (Figure 6.40). The original simulation's decisions have the 1 superscript and the additional decisions have the 2 superscript.

Introducing decisions in this way had two consequences. It reduced the duration from a day and a half to a day. And, the introductions changed the issue focus and maintained interest and engagement.

Decisions	Period
Percent Markup[1]	1
Inventory Purchases[1]	1
Marketing (Sales Support) [1]	1
Staff Numbers[1]	2
Training Days[2]	3
Number of Products[2]	4
Receivable Days[2]	4
Electronic Linkage[2]	5
Demo Equipment[2]	5
Demo Room[2]	5
Small Project Initiative[2]	5

Figure 6.40: Decision Introduction

Changing and Adding to the Documentation

Changing the learners' manual involved changes to the scenario, business data and history and this did not lengthen the manual. However, adding decisions in stages necessitated creating handouts for each period where new decisions were introduced. These handouts described the new decisions, the business data associated with these and the reports relating to these. Although the Learners' Manual did not increase in size the Background Notes (Tutor's Manual) was significantly larger than the original manual.

Recalibrating the Simulation

The final task was to reduce the level of profitability to that of a typical electrical equipment distributor. This was much more of a problem as there was a conflict between reality and engagement. Real world profitability in the industry was very low and difficult to improve. If this situation had been transferred to the simulation there would be problems with engagement and affection. Although profit levels were reduced significantly they were not quite as low as the real world and it was easier to increase profitability – an example of stylisation to ensure engagement.

Outcomes

The customised work increased the simulation model by 46%, the number of parameters increased by 63% and the number of reports produced by 55%. However, by using existing models (from other simulations), the customisation took a relatively short time (12 days).