COMPUTER BUSINESS SIMULATION DESIGN: NOVELTY & COMPLEXITY ISSUES

Hall, Jeremy J. S. B.
Hall Marketing
Phone +44 20 7537 2982
jeremyhall@simulations.co.uk

Key Words: Simulation, Design, Metrics

ABSTRACT

Developing a business simulation is a time consuming, costly and potentially risky task. And, the risk of time and cost overrun is associated with the novelty and complexity of the simulation. This paper suggests the aspects of simulation novelty and complexity that are the sources of design risk and explores these in the context of a range of actual simulation designs so that designers can take action when forecasting costs, the development schedule and starting the design and development work.

The areas of design novelty and difficulty are described and discussed in the context of five representative simulations. Following this the impacts on actual design difficulty and complexity are discussed and analyzed for the simulations. Finally, a design time model is proposed that empirically links model size with duration and novelty and design time with model size. In conclusion, the paper suggests that design time can be forecast from simulation duration provided the forecast is adjusted to take into account simulation novelty.

INTRODUCTION

Five representative simulations were analysed to develop a model of the areas of novelty and complexity that may affect the development of a business simulation and hence design time and risk. The simulations analysed had a range of durations (from two to nine and a half hours) and covered a wide range of learning needs, simulation type and industry (consumer goods, banking, distribution, entrepreneurship and engineering).

When designing these simulations, the designer used an existing platform that provided standard functionality and typically represented 85 to 95% of the simulation software (simulator). The use of a platform meant that the development risks explored in the paper are constrained to the development of the simulation model and it's novelty.

Hall (2005) suggests a simulation methodology that takes the design through several stages. The first two of these (Needs Definition and Simulation Specification) take the design process to the point where a client will require a cost estimate (or quotation) and a development schedule. And it is at this point that the designer must be able to identify risk areas and concerns before starting the simulation design. Thus it is at this point that the novelty of the simulation must be assessed and a forecast of how this may affect the design.

Initially this paper assesses the novelty of the simulations. It then proceeds to an investigation of the impact of novelty and simulation duration on the simulation model, simulator and associated data. Finally, the impact of the size of the simulation model and novelty on design time is investigated and a design time model proposed.

THE SIMULATIONS

The range of simulations investigated were those where the designer was able to articulate the novelty and design issues identified after the Needs Definition and Simulation Specification stages but before the simulation software was designed.

Product Launch is a short simple simulation exploring basic marketing, financial and team-working concepts. It is designed for a wide range of learners and uses. Participants range from school and MBA students through junior managers and specialists on marketing and financial appreciation course to all levels of staff at business conferences. Participants launch a new fast moving consumer product (the *Souper Hot* self-heating soup). During the simulation, participants manage the *self-heating soup* over the Product Life Cycle (making decisions for eight simulated quarters). The main design issue was the short (two hour) duration and this was reflected by the fact that the simulation involves just three decisions (price, promotion and production level).

Modern Banking is a total enterprise simulation that allows participants to manage a complete commercial bank. Participants take over a bank managing two market sectors (retail & commercial) using two levels of staff (junior & senior) making decisions about interest rates, promotion, staffing levels, staff quality and productivity improvement. The simulation is designed for junior managers, graduate employees and functional specialists (making decisions for six to eight simulated years). There were two design issues associated with the design of this simulation. First, the accounting structure and financial focus of a bank is different from an industrial/commercial company. Specifically, instead of the Income Statement being the focus, the Balance Sheet is the focus. Secondly, there was a very tight development schedule.

DISTRAIN is a total enterprise simulation that allows the exploration of the issues associated with a distribution company. It was developed as an extended version of the Distribution Challenge simulation to allow sales people from a major manufacturer to better understand the issues facing their customers. Participants make decisions for six to eight simulated quarters covering pricing, purchasing, promotion, staffing and business improvement. There were three issues associated with the design. First additional decisions (and complexity) had to be added to an existing simulation without lengthening its duration. Second, the simulation had to be recalibrated to reflect the (lower) margins in the client's industry. And, third the simulation was to be run by client staff.

S.E.E.D. (Strategic Exploration of Entrepreneurial Directions) provides an in-depth exploration of an entrepreneurial opportunity. It covers the marketing, financial and operational plan together with the market research and an exploration of entrepreneurial purpose. The plan is created over five to seven simulated months. To explore marketing, finance and operations in sufficient depth there were a wide range and large number of decisions and the model would be complex. The prime audience for the simulation was to be science, engineering and medical undergraduates who did not have prior business knowledge. Further, the project partner required the simulation to last no more than six hours. The design issues were the probable conflict between the duration and the simulation's complexity and the fact that the participants would have little or no prior business knowledge.

Prospector provides a detailed exploration of a *stage-gate* process where participants build a portfolio of engineering projects. Initially, participants search for projects that complement the portfolio, will be profitable and risk free. At the next stage gate they qualify projects further by obtaining more detailed information about them. Next they explore profit opportunities against *virtual competitors.* And finally, progress projects to negotiate payment terms and the work schedule. At the end of the simulation the portfolio of projects is executed and the actual profits, durations and project completion time are assessed. Instead of stepping a business through several time periods, individual projects are progressed to an appropriate stage. When developing their project portfolio, participants recurse repeatedly to the first stage to search for additional projects and to other stages to progress projects forward. The target audience for the simulation were middle managers from an engineering, technical and architectural background. The design issues were the novel structure (that meant that no existing simulation platform existed) and also, unlike the other simulations that are *deterministic*, Prospector is a *stochastic* simulation where the projects and their characteristics were developed on a random basis.

SIMULATION SUMMARY

Simulation	Product Launch	Modern Banking	DISTRAIN	S.E.E.D.	Prospector
Simulation Type	Concept	Total Ent.	Total Ent.	Planning	Process
Duration	2 Hours	8 hours	9.5 hours	6 hours	6 hours
Delivery Mode	Direct	Tutor	Tutor	Direct	Direct

Figure 1: Simulation Summary

Simulation Type: The simulations were classified based on a taxonomy suggested by Biggs (1990) and extended by Hall (1996). Concept Simulations are those that explore a particular concept (Product Launch – the Product Life Cycle). Total Enterprise simulations involve participants running a complete business with decisions covering marketing, finance and operations (Modern Banking involved running a complete bank and DISTRAIN involved running a complete distribution business). Planning Simulations involve participants developing a business plan (S.E.E.D. - an entrepreneurial plan). And, Process Simulations involve the exploration of a business process (Prospector - a stage-gate process). (Note: Process Simulations replace Hall's Analysis Simulations in his taxonomy.)

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

Delivery Mode: This is who will use the simulator (software) - the participants (direct) or the trainer (tutor). Who uses the simulation has a bearing on the participant and tutor support needs. Where the participants use the simulation directly, there is the need to incorporate a Participant Support System that provides help with software use, tasks and definitions of terms. Where the trainer uses the software only, the support needs may be substantially less. If the trainer has significant business knowledge, he or she will not need definitions of business terms. And, if he or she is an experienced computer and simulation user, he or she will not require help with the software.

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 refers to the extent to which the designers have industry knowledge and 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 the Foundation Challenge simulation (a simulation that simulates the operation of a not-for-profit organisation) where the designer had no experience of working in the sector.

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 specific to the client's customers (distributors of electrical equipment).

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. In contrast, for the other simulations no models existed and 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 where the simulation is complex (addressing strategic issues). This is because a short, simple simulation will focus of a few, basic issues and so will not require extensive, in depth industry knowledge.

Simulator/Process Novelty refers to the extent to which the designers have had experience of the particular form of the 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 participant(s) (and, if appropriate a 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, and DISTRAIN are examples of this form. Other simulations involved producing and evaluating a series of plans (S.E.E.D.) or taking the participants through a business process (Prospector). Each of these involves different processes **-** both computing and learning.

The learning process 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, all the simulations were to be run in a single session. In contrast Prospector was to be run in four sessions through a course as a 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 participants' ability to reflect (consciously and unconsciously) and the opportunity to intersperse and link the simulations to other learning activities.

The interactions between the participant(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. The extent to which a trainer is involved in the process. Here, in contrast to some e-learning simulations, the simulations were to be run in a classroom setting with a trainer facilitating and managing the experience.
3. Who uses the simulation software (simulator) – either the participants entering their decisions directly or the trainer entering decisions on behalf of the participants. Both DISTRAIN and Modern Banking had decisions entered by the trainer. All the remaining simulations involved the participants 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 2).

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

Figure 2: Novelty & Design Issue Characteristics

Scenario Novelty: This is the extent to which the business modeled might cause problems. Product Launch, DISTRAIN and Modern Banking all had scenarios that were easily described and modeled. The S.E.E.D. and Prospector had scenarios that were substantially more novel. For S.E.E.D. 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 modeled.

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 participants 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. And, although S.E.E.D. 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.

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

The S.E.E.D. 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 participants allows them to work *asynchronously* and this shortens the simulation.) However, for S.E.E.D, despite introducing decisions in stages and the direct use of the simulation, there was still a major mismatch between duration and decision complexity.

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. So, it was felt that the Prospector simulation *might* have delivery mode (software use problems). And it was certain that the S.E.E.D. simulation would have problems. Because Direct Use simulations are used by the participants who have no prior knowledge of the software delivery mode issues are generally higher than those for Tutor Mediated Simulations (where only the trainer uses the software). This is because the trainer can spend time beforehand becoming familiar with the software and may use the simulation regularly whereas the participants cannot do this.

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

Participant Support Need: This depends on the type of simulation and the prior knowledge and experience of the participants. It is only required when the simulation is used directly by the participants (Delivery Mode - Direct). Were the software is used by the trainer (Delivery Mode – Tutor) online participant 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 participants have reasonable business experience and knowledge or the concepts covered by the simulation are simple the need is average. However, where the participants have little business knowledge or experience (S.E.E.D.) or where the simulation is complex (Prospector) participant support needs are high.

Tutor Support Need: This defines the amount of support required by the trainer and like the participant support need depends on the simulation type and the business and industry experience and knowledge of the trainer. Thus it is only required 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. And, this meant that is was necessary to incorporate comprehensive Tutor Support covering using the software and definitions of terms and explanations of reports.

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. Also, 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 the requisite business and simulation use experience a Train the Trainer course was necessary. And DISTRAIN had a Train the Trainer course as part of piloting the customized simulation. Although the S.E.E.D. and Prospector simulations were be complex, they would be run by trainers who were very experienced and who were involved in their development and so a Train the Trainer course was not 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 specialized focus of DISTRAIN, Prospector and S.E.E.D. meant that it was not appropriate to consider multiple versions.

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 3 and discussed next.

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

Figure 3: Probable Design Complexity Issues

Decision Complexity: This is based on the number, type and structure of decisions made by the participants. It ranged from three (for Product Launch) to several dozen (for S.E.E.D.) 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. So, for example, as for S.E.E.D., there was a conflict between the number of decisions required and the duration this led to high Decision Complexity.

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 categorized as follows:

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

Reports for participants 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 – in use they allow the trainer to answer participant questions about how results are calculated and during the design stage they help validate the models.

Reports to help managing the experience 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 S.E.E.D. 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 the random samples used by the simulation performed realistically. To reduce the duration of S.E.E.D. *intelligence* was built into the simulation model and it was necessary to attune these models during the design and this required reports to help with the design. This is summarized in Figure 4 where for DISTRAIN there is an apparent conflict between low complexity and the number of reports. This reason for this disparity lies in the fact that DISTRAIN was developed from an existing simulation and many of the reports existed and some of these were superfluous (to the new simulation).

Simulation	Product Launch	Modern Banking	DISTRAIN	S.E.E.D.	Prospector
Reporting Complexity	Low	Low	Low	High	High
Number of Reports	57	137	209	133	112

Figure 4: Reporting Complexity and Numbers

Model Complexity: This is the size and structure of the Simulation Model. Besides producing the Business Reports for the participants, the model must produce outputs to show interim calculations, analyze business performance, 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 S.E.E.D. the Model Complexity was high because to simplify and hence shorten the simulation *intelligence* was built into the model. Figure 5 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.)

Simulation	Product Launch	Modern Banking	DISTRAIN	S.E.E.D.	Prospector
Model Complexity	Low	Average	Low	High	High
Relative Model Size	27%	78%	87%	137%	169%

Figure 5: 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 compares the number of variables used to control the simulation, for decision entry and reporting and a significant number of variables are vectors. (Thus one variable might be price but existing as a vector containing the prices for each market.) Further, the number of variables does not include variables used for interim calculations. Thus the actual number of variables used by the simulations is significantly higher than shown.

Simulation	Product Launch	Modern Banking	DISTRAIN	S.E.E.D.	Prospector
Data Complexity	Low	Average	Average	High	High
Number of variables	100	537	476	534	588

Figure 6: 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 standardized and then the size of the help database just depends on the additional Participant and Tutor Support Needs.

The two simulations where the delivery mode was the tutor (Modern Banking and DISTRAIN) illustrate extremes of 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 trainer's knowledge, Modern Banking had a large help database (that explained tasks and reports and defined terms).

Where the simulation is used directly by the participants (Product Launch, S.E.E.D and Prospector) the online help need is greater. For S.E.E.D. the lack of participant 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 participants meant that their help databases were of average size.

Simulation	Product Launch	Modern Banking	DISTRAIN	S.E.E.D.	Prospector
Online Help Complexity	Average	High	Low	V. High	Average
Number of Help Screens	254	554	166	668	261

Figure 7: 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 behavior 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 S.E.E.D. and Prospector had a significantly higher level of calibration complexity. For S.E.E.D there were three entrepreneurial opportunities and none of these could be 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 participants to narrow these down to half a dozen or so in a reasonable time. Thus both S.E.E.D. and Prospector had a high level of Calibration Complexity.

Simulation	Product Launch	Modern Banking	DISTRAIN	S.E.E.D.	Prospector
Calibration Complexity	Average	Average	Average	High	High

Figure 8: 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 and DISTRAIN all used existing platforms and no additional code was required. For S.E.E.D. it was necessary to provide 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	S.E.E.D.	Prospector
Platform Availability	Complete	Complete	Complete	Most	Some

Figure 9: Extent of Platform Availability.

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

Simulation	Product Launch	Modern Banking	DISTRAIN	S.E.E.D.	Prospector
Relative Novelty	Average	Average	Average	V. High	V. High

Figure 10: 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 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). And design time is forecasted by dividing the expected number of program lines by the expected number lines that can be programmed a day (Figure 11).

Duration ê Model Size ¸ Lines/Day ê Design Time

Figure 11: Design Time Forecast Model

And, 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's (1994) showed that complexity was strongly correlated with duration and Keyes (1977) suggested that complexity is linked to model size. Thus it was hypothesised that simulation duration and model size would be correlated and the size of the simulation model could be 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 tested 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 the same organisation as the representative 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).

For the eighteen homogenous simulations it was found that model size correlated strongly with simulation duration (Figures 12) ($p<.01$).

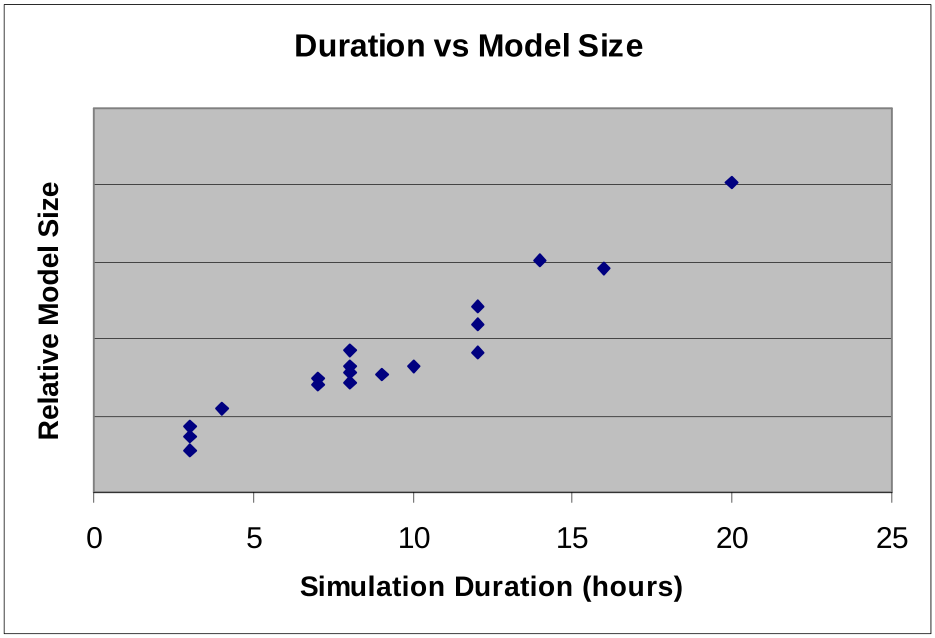

Figure 12: Graph linking Duration with Model Size

Model Size and Novelty: The regression model was derived from group of simulations with an average level of novelty. And, if the regression model is used to forecast model size for the simulations, this compared with the actual size gives an indication of the relative impact of novelty (Figure 13).

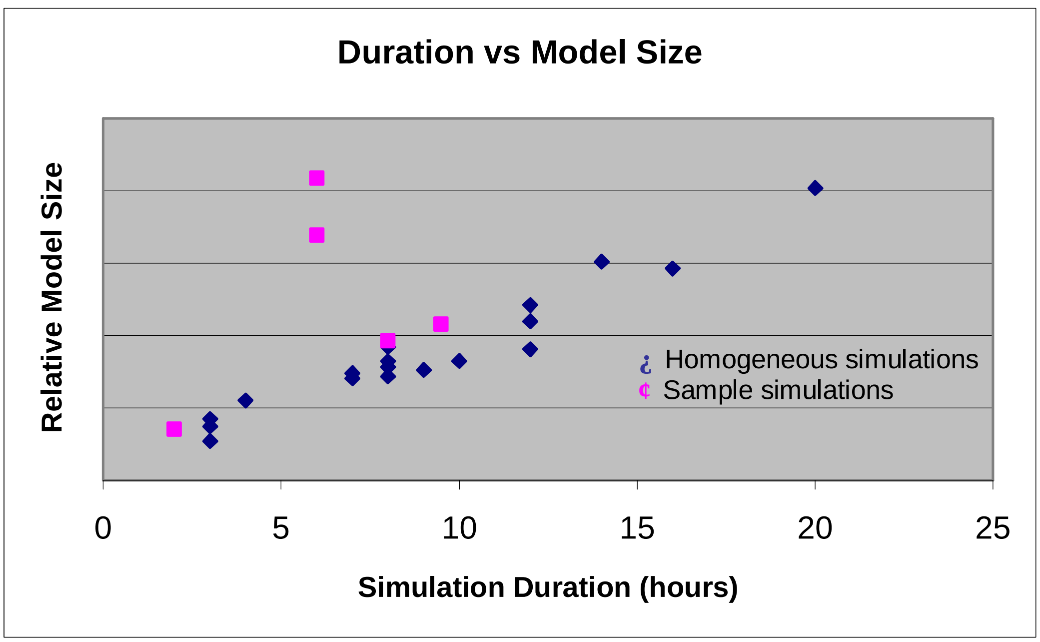

Figure 13: Duration/Model Size Graph including simulations.

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. And, if novelty is average, duration can be used to forecast model size. But if the simulation is perceived as novel, then the model size is likely to be significantly larger than that forecast from duration. This is illustrated in Figure 13 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 S.E.E.D. and Prospector this lead to a significantly larger model size and longer development times.

Reflecting its novelty, the S.E.E.D. model size is nearly three times the size expected based on the duration-based forecast. And, 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 14a and b 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 14a: Single Line Price Response Algorithm

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

Figure 14b: Multiple Line Price Response Algorithm

Physical vs. Logical Lines: Figure 14 shows another aspect of measuring program size – the difference between physical lines of code and logical lines of code. Although Figure 15a is only one line of code it contains two logical elements (the if-then conditional and the calculation). Likewise Figure 15b is three lines of code but still only contains 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 15b.

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 the designer had already designed a significant number of simulations and so was experienced. For inexperienced designers, the number of program lines per day will be less and 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. And Sommerville (1992) suggests that "particularly for smaller systems" the effort curves are almost straight lines. Further for simulations, the model consists of a large number of sub-models (algorithms) each of which has a similar level of structural complexity. As a simulation model consists of an aggregation of sub-models, 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, 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. And, 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 Hall's design methodology (Simulation Design, Simulator Development, Simulation Validation and Design Finalization). Figures 15 show 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 ($p<.01$).

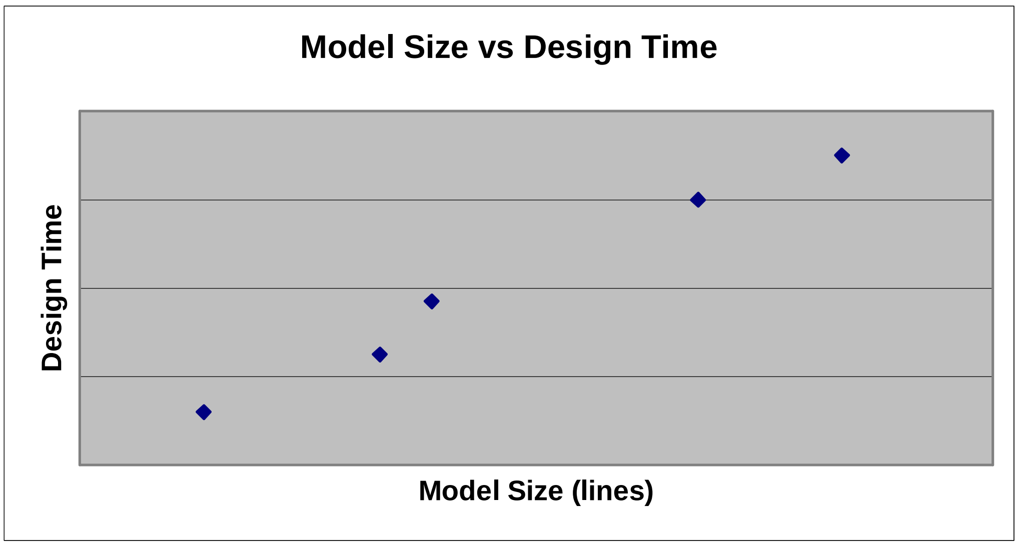

Figure 15: Graph linking Model Size and Model Design Time

PLATFORM AVAILABILITY AND DESIGN TIME

If a simulation platform does not exist then there is also 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. And this means that the time to create the platform does is not roughly proportional to duration (as is the case for models with an average level of novelty).

CONCLUSIONS

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 relatively 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)**

REFERENCES

Albrecht, Allan J. (1979) “Measuring Application Development Productivity”, Proceedings of the IBM Applications Development Symposium, Monterey, CA, 1979

Albrecht, Allan J. Gaffney Jr., John E, (1983) “Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation”, IEEE Trans. Software Eng 9(6): 639-648

Biggs, William D. (1990) “Introduction to Computerized Business Management Simulations” in *Guide to Business Gaming and Experiential Learning* ed James W. Gentry, Association for Business Simulation and Experiential Learning

Boehm B.W. (1981) *Software Engineering Economics,* Englewood Cliffs NJ: Prentice-Hall

Gold, Steven (2003) “The design of a business simulation using a system-dynamics-based approach”, *Developments in Business Simulation and Experiential Learning,* Volume 30, 2003

Hall, Jeremy J. S. B., Cox, Benita M (1994) “Complexity: is it really that simple”, *Developments in Business Simulation and Experiential Learning,* Volume 21, 1994

Hall, Jeremy J. S. B. (1996) *Simulation: Virtual Business Experience* Hall Marketing, London, England

Hall, Jeremy J. S. B. (2005) “Computer business simulation design: the rock pool method”, *Developments in Business Simulation and Experiential Learning,* Volume 32, 2005

Jones, T.C. (1986) “The SPR Feature Point Method”, Software Productivity Research Inc.

Keys, J. Bernard (1977) “Total enterprise Games: Computerised” in *The Guide to Simulations/Games for Education and Training,* ed R. E. Horn Cranford NJ: Didactic Systems Inc.

Longstreet, David (2004) Function Points Analysis Training Course, Longstreet Consultancy Inc. www.SoftwareMetrics.Com

Park, Robert E. (1992) “Software Size Measurement: A Framework for Counting Source Statements”, Software Engineering Institute, Carnegie Mellon University, Pittsburg, Pennsylvania.

QSM – Function Point Languages Table – Qualitative Software Management,Inc.Htm, Version 3 3.0 April 2005

Sommerville, Ian (1992) *Software Engineering*, Addison-Wesley