Developments in Business Simulation and Experiential Learning, Volume 31, 2004

COMPUTER SIMULATION: A DESIGN ARCHITECTONIC

Jeremy J. S. B. Hall
Hall Marketing
jeremyhall@simulations.co.uk

ABSTRACT

This paper explores the development of a computer simulation design approach that is grounded in a three level architectonic that defines market needs & constraints, core values, and design elements. From this the paper explores the architectural needs of business simulations, describes a software architecture, and experience using the architecture to design seven new simulations and re-engineer nineteen other simulations dating from the 1970s, 1980s, and early 1990s. Finally outcomes are discussed in terms of future proofing, design flexibility, customizability, and speeding simulation development.

Keywords: design, computer, simulation

INTRODUCTION

Although having designed several computer simulations for management development and business training, by the early 1990s because of on-going technology driven change, I felt that I could and needed to re-engineer my design approach to:

1. ***better meet training needs***
2. ***speed development***
3. ***future-proof the designs.***

Instead of incremental designs changes, I felt that I could to take my own experience and knowledge and that of others, link this to adult learning theory (Knowles, 1998) and translate this into rationales, design models and computer software that would produce a step-wise improvement in my simulation development process.

The project began with a review of why and how trainers used simulation and a literature search of the use of simulation in management development and business training. This exposed several *threads* and it was apparent that I needed a way of organizing and structuring this information. This lead to the development of a design/business model or architectonic to define, structure, summarize, and develop design needs. From this architectonic, an architecture was developed and implemented in computer software.

THE ARCHITECTONIC

With an objective of ***better meet training needs*** the design must be grounded on market needs and this led the architectonic's outer ring (Hall, 1995b) - the objective definition of these needs. Central to the architectonic there are a few, core values that were distilled from market (customer) needs, wants and values. Linking these two is central ring defining the design elements of simulations that are to be implemented in the software architecture. Thus the architectonic has three parts (Figure 1):

- **Market Needs & Constraints**
- **Core Values**
- **Design Elements**

And from these derive:

- **Architectural Needs**
- **Architecture**
- **Experience with the Architecture**
- **Outcomes**

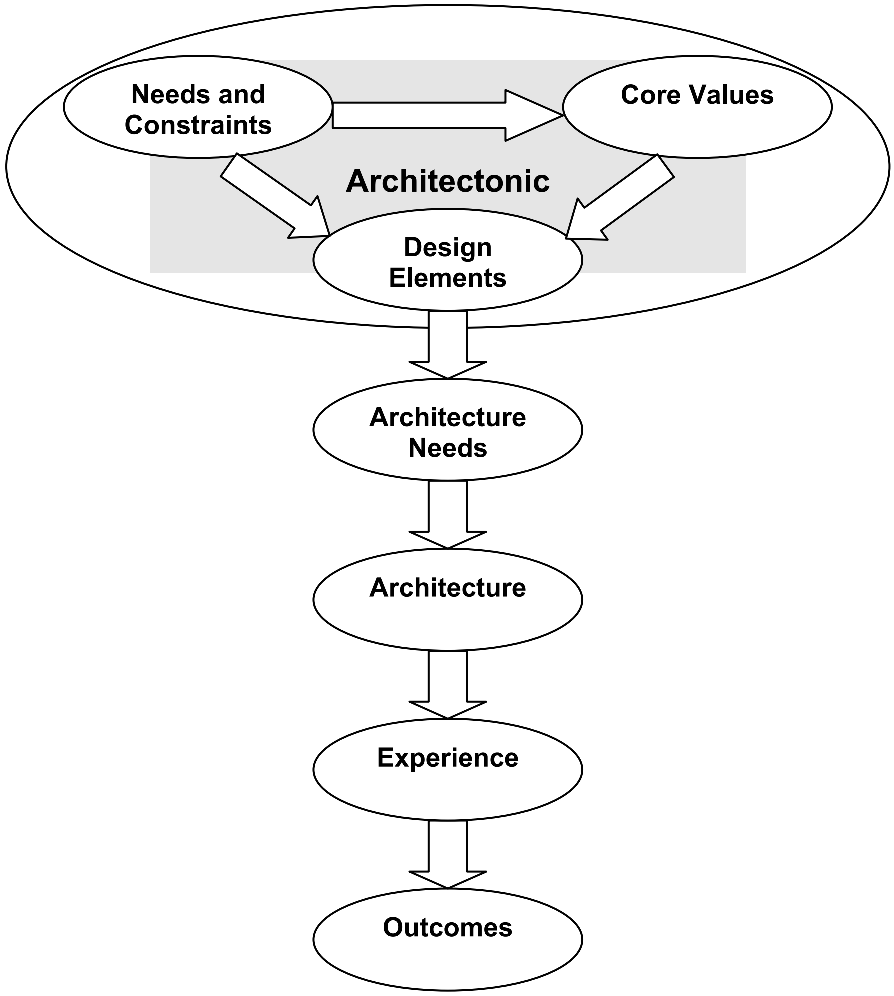

Figure 1: Architectonic & Architectural Development

MARKET NEEDS & CONSTRAINTS

An analysis of why trainers and organizations used simulations for management development and business training suggested that there were four areas of needs, wants, and constraints:

- **development (learning)**
- **reasonable duration**
- **target audience**
- **manner of use**

The development (learning) and manner of use needs were developed based on an analysis of some two thousand runs of simulation and discussion with trainers and training providers in the UK, Europe and the US (Hall 1998). And, although described and discussed separately these are not independent of each other.

Development (learning) needs subdivided into

- **knowledge exploration**
- **skills practice & development**
- **motivation**
- **assessment**
- **learning enhancement**

As a generalization, this dimension defines ***product purpose*** (rather than product features and functions).

Duration is a common, perhaps universal concern of trainers and training providers and so the ability to provide simulations with short durations is a prerequisite. As a generalization, this dimension defines the key ***cost element***.

The **Target Audience** subdivides into:

- **training providers**
- **trainers**
- **trainees (learners)**
- **organizations paying for training**

As a generalization, this dimension defines the ***people*** involved in the purchase and use of the product. It allows the study of their objective and subjective disposition both pre and post sale and exposes the links and associations.

Manner of Use describes the way companies use business simulation and subdivides into two sets (training and other use) and eleven subsets defining how the simulation would be used.

Training Use	Other Use
1. Course Finale	7. On a conference
2. Course Theme	8. Spare-Time learning
3. Course Starter	9. In graduate recruiting
4. Course Break	10. For assessment
5. To reinforce learning	11. As a promotional contest
6. Standalone seminar	

Figure 2: Ways simulation used by companies

As a generalization, this dimension defines the ***usage needs*** of the product.

CORE VALUES

To an extent the core values (**effective, efficient, and consistent learning**) (Hall, 1995b) are like mother-hood and apple pie - good things. However, they do provided a series of *touchstones* extracted from and linking to the market needs and serve to focus design effort.

Effective Learning measures the way the simulation matches and fulfills the development (learning) needs and is impacted by the target audience and manner of use and constrained by duration needs.

Besides looking at the effectiveness of learning from the learner's viewpoint it is also necessary to look at it from the point of the view of the other audience types (training providers, trainers and organization paying for training).

Efficient Learning

Efficient Learning measures the *cost dimensions* of learning. And, although acquisition and usage cost were important, the main factor (linking to the duration need in the outer ring) was the amount of learning that could be done in a given period of time.

For corporate training every learner must learn and, besides consistency within a course, every course must **consistently deliver learning**.

These core values provide ***touchstones*** that when linked to market needs provide design direction and focus.

DESIGN ELEMENTS

These link the Market Needs to the Core Values and provided a starting point for the architecture and the product development. For the computer simulations there were four design elements:

- **The Simulation Model**
- **Delivery Dynamics**
- **Tutoring Needs**
- **Diversity of Need**

Simulation Model

The conventional view of "good" simulation design focused on the simulation model (Miller & Leroux-Demers, 1992), yet the model's scope and complexity has a major impact on effective and efficient learning.

The simulation model relative to learning needs can be viewed as two overlapping sets (Figure 3). One set (A + B) represent the issues raised by the model and the other (B + C) represents the learning needed.

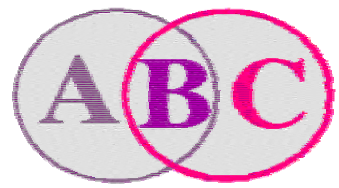

Figure 3: Model and Learning Need Sets

These sets and the overlap (B - learning provided by the model) reveal the impact of model complexity (size) on the core values of effective and efficient of learning. For learning to be efficient and as learning needs are defined by the B + C set, B must be large compared to C. As duration correlates with model complexity (Hall & Cox 1994), duration is defined by the A + B set. Thus, for the simulation to be efficient, B must be large compared to A. Further, if A is large compared to B learners may be confused by the complexity and (adult learners) may question the relevance of the simulation. Finally, as development time correlates with model size (A + B), a complex unfocused model is uneconomic in development terms as it incorporates aspects that do not contribute to learning needs.

Although much of the *received wisdom* is that the "goodness" of a simulation has a high positive correlation with complexity (Miller & Leroux-Demers, 1992) and this leads to a design that focuses on modelling the *"real world"*, this conflicts with the design of *lean products* that ***"deliver value to the customer - and nothing more. There is no design overshoot. There are no features which are technologically interesting but which the customer does not value"*** (Cloke, 2000).

As a generalization, this design element defines the attributes of **the *basic tangible product offering*** and how it is positioned between needs and values.

Delivery Dynamics

The literature seems to have few references to the dynamics of simulation use and how this impacts learning. Yet the experiential learning cycle (as described by Kolb 1984) that is a characteristic of simulation is analogous to the feedback process of control systems (Hall & Cox, 1993). For simulations this leads to a systems dynamics model consisting of three dynamics:

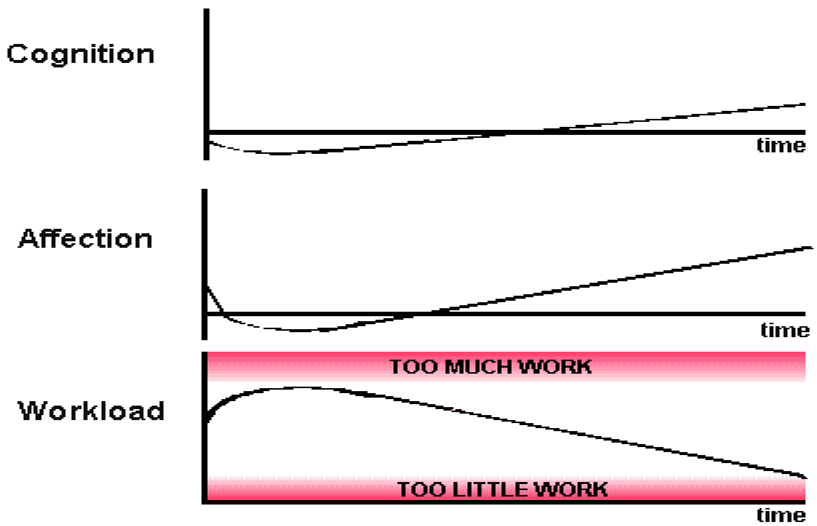

Figure 4: Delivery Dynamics

- **Cognition**
- **Affection**
- **Workload**

Over the course of the simulation these change and the typical pattern is shown in Figure 4

Typically, the **Cognition Dynamic** starts with the learners somewhat confused with the task and the business situation facing them. Then as time passes and they make decisions and review these understanding grows and learning takes place.

Typically, the **Affection Dynamic** starts with the learners enthused about the activity. Then as they discover the task is more difficult than envisaged, learners become slightly disaffected, but as they gain command of the situation and learn, affection increases.

Typically, the **Workload Dynamic** starts high as the learners become familiar with the task, their fellow learners and the business that they are to run. But as time passes and the participants learn to handle the task workload tends to fall.

These patterns show problems and opportunities. If workload is maintained during the simulation, then more learning (cognitive development) can be delivered (Figure 5). As a generalization, this design element defines the ***dynamics of product use.***

Tutoring Needs

Both the learning needs and the learners predicate the need for a trainer to run the activity. And, the trainer has three major areas of work (Hall, 1994b) and these are:

- **Administration**
- **Facilitation**
- **Learning Management**

Where **administration** is concerned with the smooth running of the activity, **facilitation** with the *reactive* support of the learners and **learning management** with the *proactive*

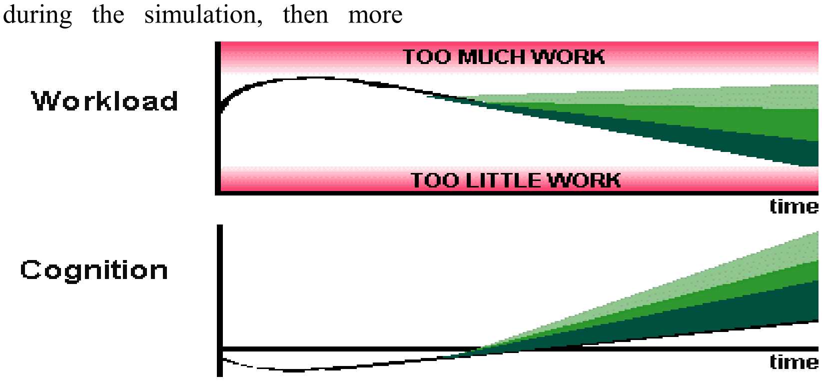

Figure 5: Effect of maintaining workload.

support of learning (the identification of learning needs and opportunities and driving learning forward).

As a generalization, this design element defines the characteristics of the ***human usage (ergonomic*** *and* ***emotional needs).***

Handling Diversity

As simulations are expensive to develop there is an economic need to design the simulation to have versions to match different market needs and markets. In other words, although the simulation model may be identical it is desirable to provide a range of simulations that address different learning needs, with different durations, for use in different ways and to be used by different customers, different types of learners, and trainers with different levels of experience.

As a generalization, this design element defines the ***range of products needed*** to ***fit market sector needs***.

ARCHITECTURAL NEEDS

Having defined the design elements these translate into a product architecture that supports:

- **Model Development**
- **Delivery Process**
- **The Tutor**
- **Multiple Versions**

Model Development

There are two starting points for simulation model design. The first is where a real world business situation is modeled and the second where only the elements that are required to produce the cognitive processing required for learning are modeled. Metaphorically speaking, modelling the real world can be describes as an *hunter-gatherer paradigm* and providing a simple and stylized abstraction to meet learning needs is an *engineered paradigm* (Hall, 2001).

Creating simulation models that focus on meeting learning needs is a *problem-solving activity* (Guindon, 1990) that is an iterative process (Ballard, 2000) and the simulation architecture must support this in a flexible, efficient yet rigorous way. In other words, like many software products, it is not possible to fully define and specify needs at the start of the design process (Poppendieck, 2003) and so the simulation architecture must support iterative, flexible and agile development while ensuring quality and robustness.

To generalize here we are developing ***product functionality*** but only in terms of customer needs and benefits, producibility, and quality assurance.

Delivery Process

The systems dynamics model of the delivery process leads to the following ways of improving learning effectiveness and efficiency:

- **Economic Calibration**
- **Ramped Complexity**
- **Tutor Intervention**
- **Feedback Style**

Economic Calibration involves calibrating the simulation so that business *difficulty* increases as the simulation progresses. For example, the business may move from being "cash-rich" to one with liquidity problems or the market situation may change.

Ramped Complexity involves introducing additional reports or decisions as the simulation progresses to introduce new learning. For instance, reports may be introduced evaluating products, customers or markets on a profit or investment center basis. Alternatively, decisions that change products or production methods can be introduced. These raise new issues, stimulate discussion and provide opportunities for additional cognitive development.

Tutor Intervention involves the trainer analyzing the situation, identifying learning needs and problems and providing suitable feedback. This is desirable because both Economic Calibration and Ramped Complexity are pre-defined and can not take into account differences between individuals, teams and courses. Because it is proactive, tutor intervention ensures consistent learning and takes advantage of learning opportunities. For instance, the tutor is able to introducing new reports and (perhaps) decisions to stimulate discussion and cognitive development and adjust the economic pressure (to make "life" easier or harder).

Feedback Style addresses the Affective Dynamic rather than the Cognitive Dynamic. At the beginning, participants are generally confused and feel overworked and thus need encouragement. Later, if they feel that they are doing exceptionally well, participants may become *manic* and will need to be challenged. In the context of Tutor Intervention this defines the behavioral style of the trainer. Also (as described later), if feedback is in the form of qualitative comments, initially these should emphasize strengths. Then later, these comments can cover weaknesses, threats and opportunities. Figure 6 shows how these process improvement impact the dynamics.

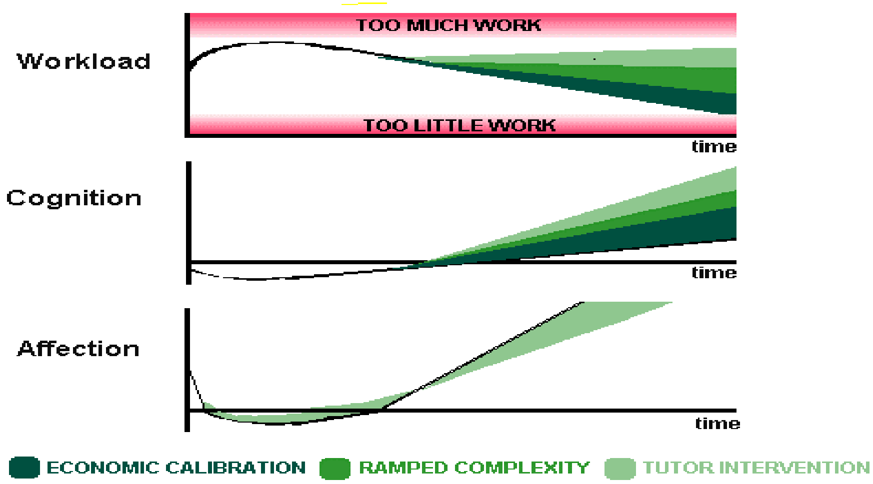

Figure 6: Design Impact on Dynamics

To summarize and generalize these elements improve ***product dynamics***.

Tutor Support

Because of the complexity of business simulations and in the interest of consistent, effective and efficient learning, there is a significant and necessary role for the trainer (Hall, 1994b) and the simulation architecture must support this. Table 1 shows ways that the administrative, facilitation, and learning management training tasks may be supported.

The **Help System** supports both administration and facilitation by providing context sensitive help with software use, the current task and, if appropriate, definitions and an on-line-manual.

The **Decision Screen** checks and validates decisions as they are entered. It rejects illegal decisions and flags unusual and sophistic decisions. Thus it protects against mistakes and misunderstandings, warns of radical and arbitrary decisions, and identifies possible learning problems and opportunities.

Explanations provide a way of clarifying how the accounting and operational calculations were done and so help the trainer answer questions about these.

Comments are qualitative comments about teams' strengths, weaknesses, decision problems and market news. These replicate feedback from staff, customers, suppliers etc. Because they can be *fuzzy* they necessitate discussion and interpretation and so ensure deep cognitive processing. Also, for the less numerate learner they provide a respite from the quantitative business and financial reports. Finally, as they come from the simulation software rather than the trainer they are not seen as an irrational criticism!

Tutor's Audit compares and explains differences between teams. Thus it tells the trainer why teams differ and suggests which teams need coaching and which need challenging.

Team Commentary provides additional reports and analyses on a team by team basis. They allow team performance to be assessed in depth and provide reports that can be fed back to teams as part of learning management

To summarize and generalize, this area of the architectonic defines the way the product and ***ancillary services*** make the product easier and safer to use.

	Administration	**Facilitation**	**Management**
Help System	Y	Y	
Decision Screen	Y	Y	Y
Explanations	Y	Y	
Comments			Y
Tutor's Audit		Y	Y
Team Commentary		Y	Y

Table 1: Tutor Support System

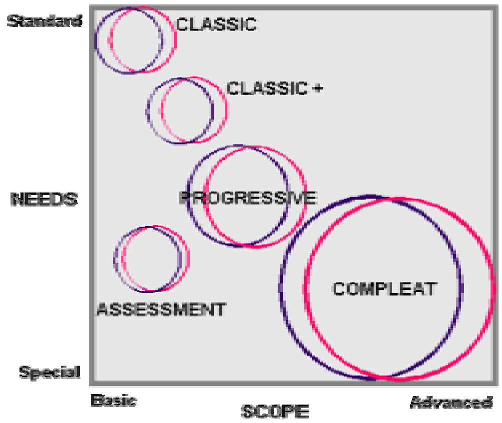

Figure 7: Versions

Multiple Versions

Multiple Versions, as illustrated in Figure 7, allow a simulation to address several sets of development needs, target audiences and manners of use. Having several versions of the simulation mean that it is better able to deliver effective, efficient and consistent learning.

Besides addressing market needs, an architecture that supports multiple versions allows the simulation to be available using different business terminology and in different languages. It allows different versions of the simulation to be offered to different market sectors at different prices. In general there is a need for any product to meet client needs by providing a range of different versions of the product.

ARCHITECTURE

Having specified the product needs, they were translated into an architecture. Figure 8 shows this and the links between the components.

With some forty products in the range and the regular need to develop simulations for clients it was decided to implement the architecture as a shell that was common to many simulations and where a specific simulation only differed in its simulation model and associated data.

The Architecture and Modelling

To facilitate ***lean design*** the *architecture* must facilitate the creation of only the models necessary to fulfill 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) A Parameter Database that allows variables to be added to the model as needed and that do not need to be predefined.
b) A Parameter Database that documents the variables used by the simulation.

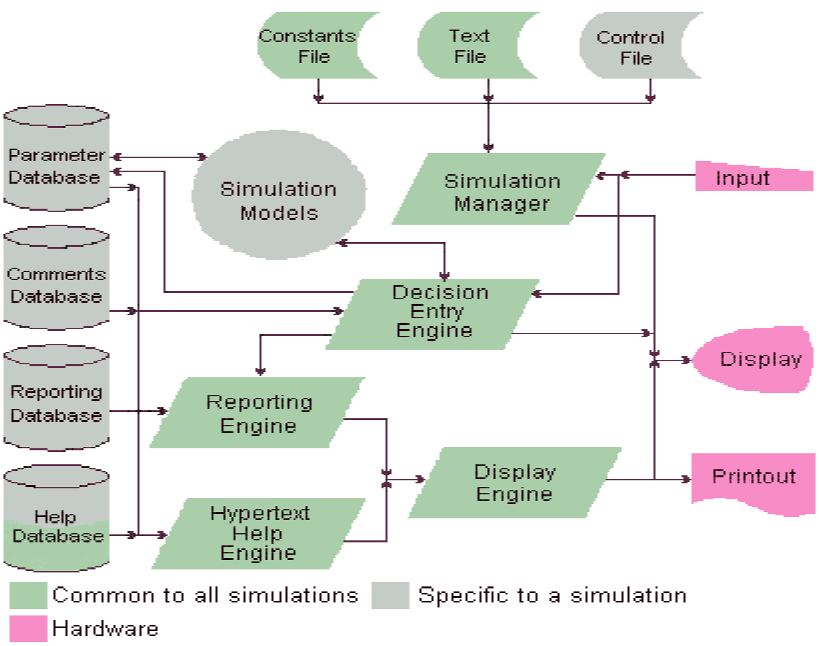

Figure 8: Software Architecture

c) A Reporting Database that defines reports and decision entry templates and allows these to be modified, augmented and restructured.
d) A 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 in association with the Parameter and Reporting Databases allows 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 preplanned qualitative feedback.

The Architecture and Tutor Support

To improve learning the architecture provides a system ***support for the trainer*** *and* ***the participants.*** 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 utilizing 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. And, 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.

EXPERIENCE WITH THE ARCHITECTURE

Between 1996 and 2002 the architecture was developed and coded. Initially it was prototyped using the MSDOS operating system and then the current version developed for the Windows operating system. During this time the architecture was tested and advanced through:

- **developing four new simulations using the MSDOS shell**
- **developing three new simulations using the Windows shell**
- **moving nineteen *old* simulations into the Windows shell**

These covered a spectrum of simulation complexity ranging from simple (lasting two to four hours), through intermediate (lasting a day) to complex (lasting up to two and a half days). Also, they covered a comprehensive range of simulation types - non-interactive and interactive *management games, planning simulations, and enhanced role-plays.* Finally they addressed a wide range of situations - *general and strategic management, marketing, sales, operations, and financial appreciation*.

OUTCOMES

Having developed seven new simulations using the shells and moved another nineteen simulations into the shells these were the outcomes:

- **Future Proofing**
- **Flexibility**
- **Customizability**
- **Simulation/Shell Proportions**
- **Speeding Development**

Future Proofing

Although computer platforms have changed and are continuing to change significantly, basic management development and training needs have not. In this context it means that if an existing simulation model can be transferred to the shell then the product's life cycle can be extended. Two developments illustrate this. A simple, short marketing simulation that was originally developed in 1977 has as a current user a major management school on its executive MBA. A second example is a complex sales management simulation. Developed in 1984, it was moved into the shell and customized for use in the Mid-West in about three weeks.

Flexibility

Besides providing a simulation in different versions a client's needs may change and without the flexibility to reorder and change reports and decisions the simulation will become redundant. This occurred for one simulation where after a year's use the client changed their business focus and strategy, Because the reports and decisions were held in the Reporting Database it took less than half a day to realign the simulation to the new business focus and strategy.

Customizing

The need for customization exists at several levels - changing terminology or language; altering the reports and their timing or adding models to the simulation. For example:

a) A generic service industry simulation was customized for use by the Football Association by changing the market and resource terminology in the databases and these changes took a matter of minutes.
b) Another simulation aimed at junior managers was simplified for use by school children by reducing the reports produced and limiting the decision sets - again in the matter of minutes.
c) A retail management simulation was customized for a West Coast client. This involved changing terminology and adding decisions and models that addressed the issues facing the retailer. These changes took about a week.

Simulation/Shell Proportions

The proportion of the software that is pre-defined in the shell is as much as 98% (for simple simulations). Even for very complex simulations 83% of the software is pre-defined by the shell. Typically, for a simulation with a one-day duration, 92% of the software is pre-defined by the shell.

Speeding Development

The combination of the *lean design* approach with the *shell* reduces development times significantly. This is illustrated in Tables 2 and 3 where development times of three recent simulations developed using the shells (Table 3) are compared with those of competitive developers (Table 2). These suggest that development times were reduced by eighty-percent or more.

Developer	Simulation	Development Hours/Hour[1]
Cap Gemini Ernst & Young	VECTOR - Electricity Trading Game[2]	300:1
University of Twente et al	KITTS - Knowledge Management Game[3]	3080:1
Various Developers	Various e-learning simulations[4]	750-1300:1
Strategic Management Group	Various[5]	1200-1500:1

Table 2: Competitors' Design Times

Notes

1. Development Hours/Hour show the amount of time (development hours) required to create one hour of simulation duration.
2. Chadwick, Jonathan (2002) *Integrating a New Strategy and Developing Key Performance Indicators* Business and Simulation Games Conference, London
3. de Hoog, Robert (2002) *KITS A Knowledge Management Simulation Game* Business and Simulation Games Conference, London
4. *E-Learning Simulations:* Tools and Services for Creating Software, Business, and Technical Skills Simulations (2002) Brandon-Hall.com
5. Summers, Gary J. (2003) The Business Simulation Industry

Simulation	Purpose	Development Hours/Hour	Model %
SEED	Entrepreneurial Planning	60:1	16%
Foundation Challenge	Not-for-Profit Business Appreciation	25:1	8%
Constructive Negotiation	Sales Negotiation	10:1	2%

Table 3: Design times using the shell

SUMMARY

To summarize and generalize, the structured innovation process described here consists of the following steps:

1) **Analyze and Define Market Needs**
   a) **Product purpose**
   b) **Customer cost elements**
   c) **People involved in purchase & use**
   d) **Usage needs**
2) **Extract and summarize Core Values**
3) **Explore Product, Dynamics, Usage and Variety dimensions**
4) **Translate into a Product Architecture**
5) **Develop Products**

REFERENCES

Ballard, G. (2000) Positive vs Negative Iteration in Design, *Proceedings Eighth Annual Conference of the International Group for Lean Construction*, Brighton, UK: IGLC-6.

Cloke, B. (2000) Lean Products in *Advanced Manufacturing Magazine*, March 2000 http:www.advancedmanufacturing.com/March00/informationtech.htm.

Guindon, R. (1990) Designing the Design Process: Exploiting Opportunistic Thoughts *Human-Computer Interaction,* 1990, V5, (pp. 305-344) Hillsdale, NJ: Lawrence Erlbaum Associates.

Hall, J. J. S. B. (1994) Computerized tutor support systems: the tutor's role, needs and tasks, in Armstrong, R, Percival, F and Saunders, D. (Ed.), *The Simulation and Gaming Yearbook Volume 2 - Interactive Learning* (pp. 76-88). London, England: Kogan Page.

Hall, J. J. S. B. (1995) Chalk and cheese? Executive short-course vs academic simulations, in Saunders, D. (Ed.), *The Simulation and Gaming Yearbook Volume 3 - Games and Simulations for Business* (pp. 76-88) London, England: Kogan Page.

Hall, J. J. S. B. (1995) Computer Simulations: a design architectonic *Proceedings of ASTD International Conference*, Dallas, USA: ASTD.

Hall, J. J. S. B. (1998) From One-Dimensional Management Teaching to Multidimensional Management Development, *Proceedings of ASTD International Conference*, San Francisco, USA: ASTD.

Hall, J. J. S. B. (2001) Corporate Cartooning: The art and science of computerized business simulation, *Proceedings of ASTD TechKnowledge Conference and Exposition 2001*, Charlotte, USA: ASTD.

Hall, J. & Cox, B. (1993) Computerized management games: the feedback process and servomechanism analogy, in Percival, F, Lodge, S & Saunders, D. (Ed.), *The Simulation and Gaming Yearbook 1993* (pp. 150-159) London, England: Kogan Page.

Hall, J. & Cox, B. M. (1994) Complexity: is it really that simple, in *Developments Business Simulation & Experiential Exercises*, Volume 21, San Diego, California: ABSEL.

Knowles, M. S., Holton III, Elwood F. and Swanson, Richard A. (1998), *The Adult Learner* Woburn, MA: Butterworth-Heinemann.

Kolb, D. A. (1984) *Experiential Learning: Experience as the source of learning and development* Englewood Cliffs, NJ: Prentice Hall.

Miller, R and Leroux-Demers, T (1992) Business Simulations: Validity and Effectiveness *Simulation/Games for Learning 22:4* (pp 261-285) London, England: Kogan Page.

Poppendieck, M. (2003) *Lean Software Development: An Agile Toolkit* Boston, MA: Addison Wesley.