Structuring Innovation: A design architectonic

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

ABSTRACT

Innovation is to do with being different but this differentiation must be meaningful in terms of market needs, wants and perceptions. Thus innovation without a strong focus on market needs and wants is meaningless. This paper explores using a structure (architectonic) that links market needs through design needs to market wants and the outcomes in terms of product architecture and development.

This exploration uses, as a case study, the development of a computer simulation architecture that recently won an innovation award and is being used to reduce simulation development times by some eighty percent while incorporating leading-edge learning delivery paradigms.

Although the *design architectonic* is explored in the context of an actual innovation, it is believed that its structure may be used for other innovations to ensure that they meet market needs and wants in a focused and structured way.

KEY WORDS

Innovation, Computer, Simulation

INTRODUCTION

I run my own, one-man, micro-business developing and providing computer simulations (management games) for the management development and business training of business people. Like the flight-simulator used to train pilots, simulations place the businessperson in charge of a *simulated* company that they run, without risk, in an accelerated manner. The process involves teams of four or five business people working on the problem guided and coached by a trainer and using a computer to simulate the impact of their decisions.

I concentrate on the corporate training market rather than the academic education market as the two are very different in terms of needs and simulation design (Hall, 1995a)

In the early 1990s, I felt that, because of on-going technology driven change, I could and needed to re-engineer my design approach to:

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

Instead of incremental changes in my designs, I wished to take my experience and knowledge and that of others, link this to adult learning theory and translate this into theories, design models and computer software that I would use to produce a step-wise improvement in my simulations.

I began the project by reviewing why and how trainers used simulation and by searching the literature on the use of simulation in management development and business training. This exposed several *threads* and it soon became apparent that I needed a way of organising and structuring this information. This lead me to developing a design/business model or architectonic to define, structure, summarise and develop my design needs. And from this I developed an architecture and implemented it in computer software.

THE ARCHITECTONIC

With a business and marketing background, I believe strongly that any design, especially, an innovative design must be grounded on market needs. This led the outer ring of my architectonic (Hall, 1995b) - the objective definition of market needs. I also felt that central to my design, I needed a few, core values and these were distilled from market (customer) needs and wants. Finally linking these two is a central ring that defines the elements that must be designed into the simulations and are implemented in the software architecture.

The architectonic has three parts:

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

And the from these derive:

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

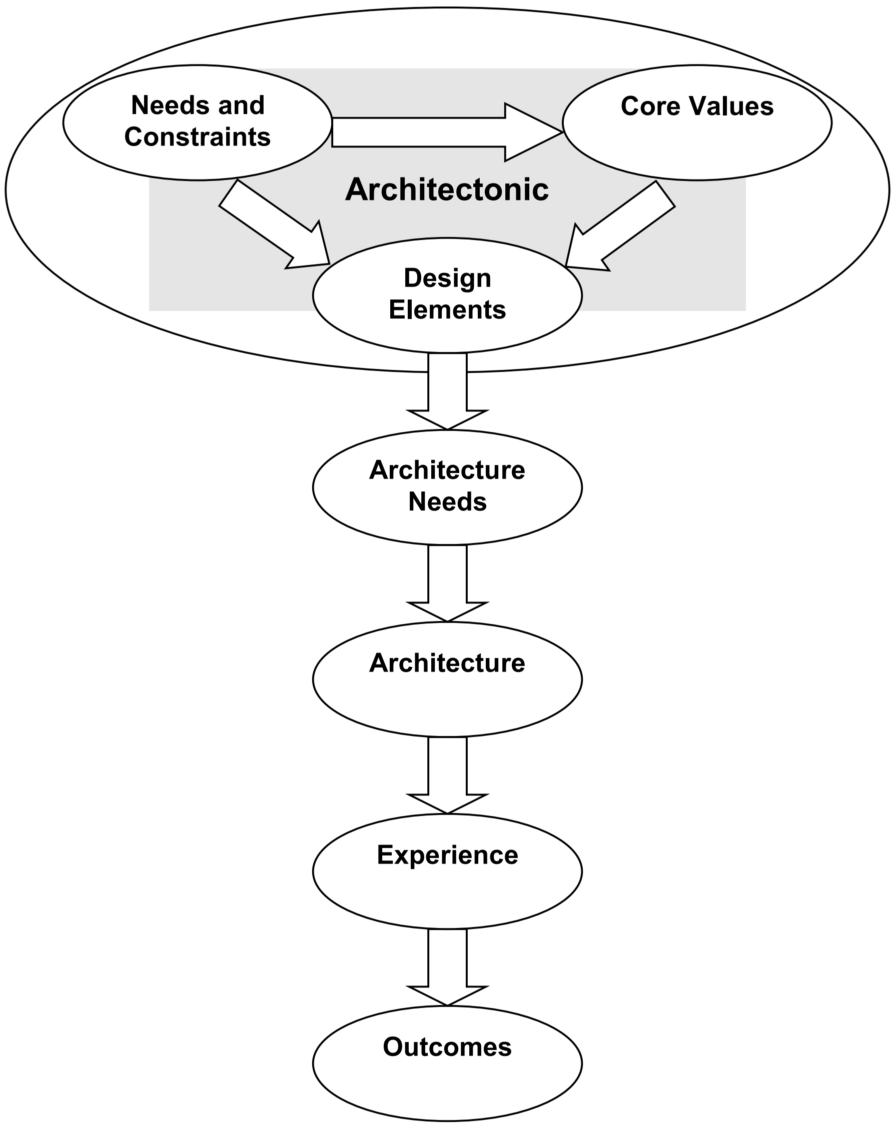

Figure 1: Architectonic & Architectural Development

MARKET NEEDS & CONSTRAINTS

From an analysis of why trainers and organisations used simulations for management development and business training I felt that there were four areas of need:

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

Because of my firm's market concentration these needs are for simulations used in corporate training where they are used on the continuing professional development of working business people. 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).

Further, although described and discussed separately these needs are not independent of each other.

Development (learning) Needs

These subdivided into

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

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

Duration

This 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 generalisation, this dimension defines the key ***cost element***. Obviously product price is one aspect of this but it is only one element of the cost to the buyer.

Target Audience

This subdivides into:

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

As a generalisation, 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

This describes the way companies use business simulation and subdivides into two sets 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 generalisation, this dimension defines the ***usage needs*** of the product.

CORE VALUES

To an extent the core value I identified (**effective, efficient** and **consistent learning**) (Hall, 1995b) are like mother-hood and apple pie - good things. However, I found that they provided a series of *touchstones* extracted from and linking to the market needs

Effective Learning

Effective Learning measures the way the simulation matched and fulfilled the development (learning) needs and was impacted by the target audience and manner of use and constrained by duration needs. For adult managerial learners this is a difficult value to gauge.

Adult learners are concerned with being able to use learning rather than just knowing (Knowles et al, 1998). And, in particular for managerial learners the ability to make those *wise decisions* that lead to business success.

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 organisation 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.

Consistent Learning

Unlike academic learning where the purpose is to differentiate between distinctions, passes and fails, for corporate training every learner must learn. Further, 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 then the product development. For my 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. This is illustrated by the fact that the majority simulation design literature is about model design and virtually none about their use, the processes and needs. Yet the model's scope and complexity has a major impact on effective and efficient learning.

The model relative to learning needs can be analysed in terms of 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.

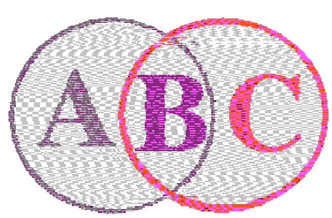

Figure 3: Model and Learning Need Sets

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.

Yet the *received wisdom* is that the "goodness" of a simulation has a high positive correlation with complexity (Miller & Leroux-Demers) and this leads to a design that focuses on modelling the *"real world".* But, 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 generalisation, 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 has 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:

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

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

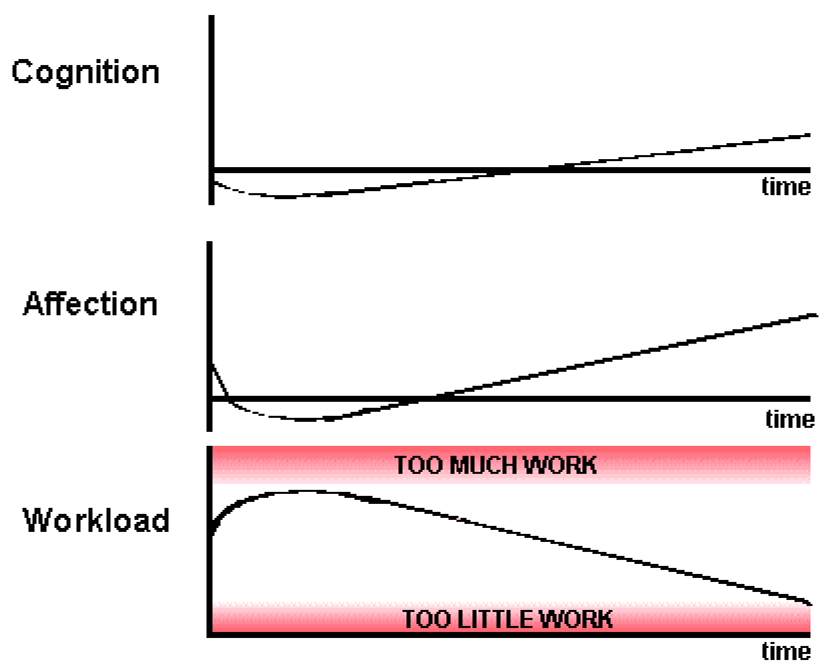

Figure 4: Delivery Dynamics

Typically, the **Cognition 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 then as they gain command of the situation and learn affection increases.

Typically, the **Affection 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 **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 falls.

These patterns show problems and opportunities. If workload is maintained during the simulation, then more learning (cognitive development) is delivered (Figure 5).

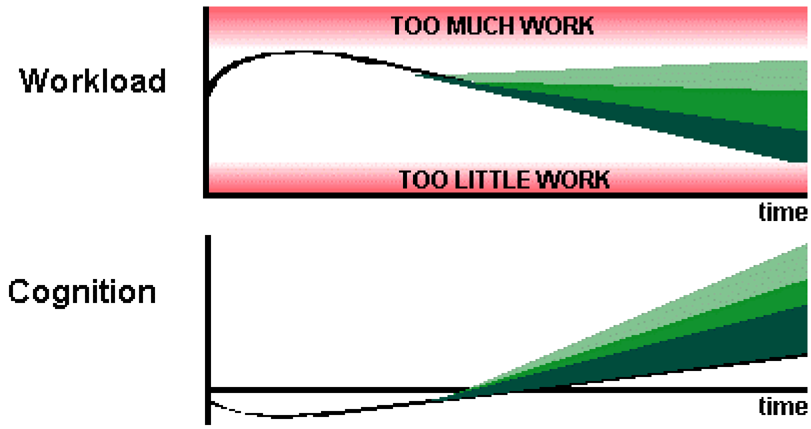

Figure 5: Effect of maintaining workload.

As a generalisation, 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* support of learning (the identification of learning needs and opportunities).

As a generalisation, 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 provide versions to match different market needs and markets. In other words, although the simulation model may be the same it is desirable to provide a range that address different learning needs, with different durations, for use in different ways and to be used by different customer, different types of learners and trainers with different levels of experience.

As a generalisation, 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 delivers:

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

Model Development

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

Creating simulation model software focusing on meeting learning needs is a *problem-solving activity* (Guindon, 1990) that is an iterative process and the simulation architecture must support this design process in a flexible, efficient yet rigorous way. In other words, like many software products, it is not possible to define and specify needs at the start of the design process (Poppendieck, 2003).

To generalise here we are defining ***product functionality*** but in terms of customer needs and benefits.

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 are introduced evaluating products customers or markets on a profit or investment centre basis. Alternatively, decisions that change products or production methods can be introduced. These raise new issues, stimulate discussion and cause additional cognitive development.

Tutor Intervention involves the trainer analysing 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 (as discussed so far). 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 behavioural style of the trainer. Also (as described later), if feedback is in the form of qualitative comments, initially these should emphasise strengths. Then later, these comments can cover weaknesses.

Figure 6 shows how these process improvement impact the dynamics.

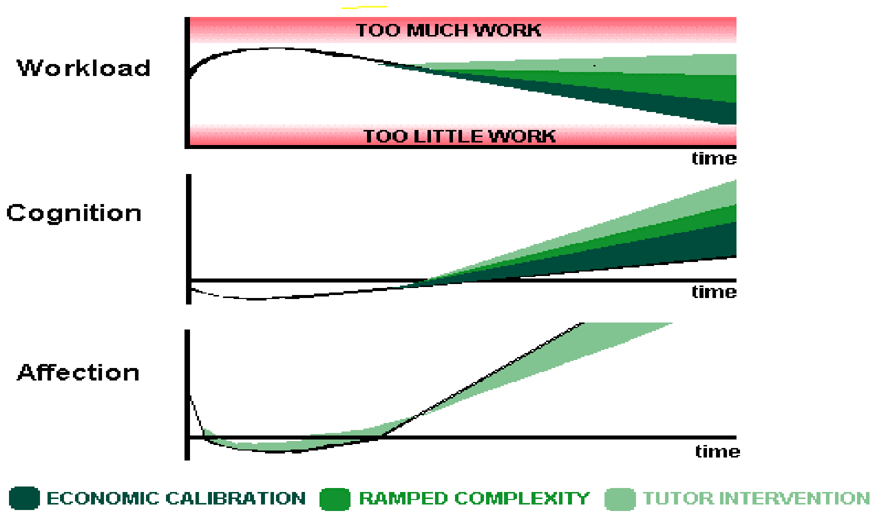

Figure 6: Design Impact on Dynamics

To summarise and generalise these elements improve ***product dynamics***.

Tutor Support

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

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

Table I: Tutor Support System

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 are *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 or 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 summarise and generalise, this area of the architectonic defines the way the product and ***ancillary services*** make the product easier and safer to use.

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.

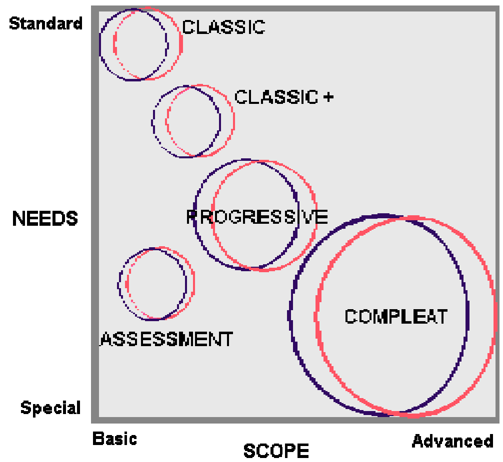

Figure 7: Versions

Besides addressing market needs, an architecture that supports multiple versions allows the simulation to be available using different business terminology and in different languages. And, it allows different versions of the simulation to be offered to different market sectors at different prices.

In general there is a need to try to meet client needs by providing a range of different versions of the product. And software products or physical products incorporating software provide a low cost ***differentiation opportunity*** and the ability to market upgrades (Hall 1978).

ARCHITECTURE

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

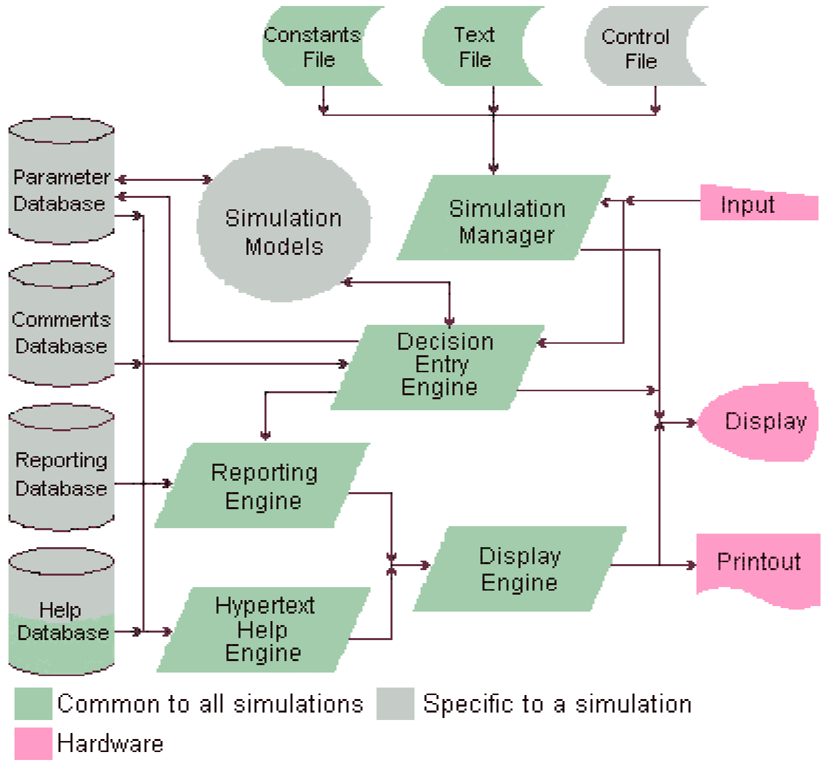

Figure 8: Software Architecture

With some forty products in my range and the regular need to develop simulations for clients it was decided to implement the architecture as a shell. This meant that the shell was common to many simulations and where a specific simulation only differed in its simulation model and associated data. (Each component of the architecture is described in Appendix 1.)

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 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 these do not need to be predefined.
b) A Parameter Database that documents the variables used by the simulation.
c) A Reporting Database that allows reports and data entry templates 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 to allow changes to the Economic Parameters as the simulation progresses.

b) The Control File together with the Reporting Database to allow 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 to provide tutor support information coupled with the ability to intervene using ad-hoc reports that are 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 to 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 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. 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.

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 the addressed a wide range of situations - *business, marketing, sales, operations and financial appreciation and management* (see Appendix 2).

OUTCOMES

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

- § **Future Proofing**
- § **Flexibility**
- § **Customisability**
- § **Simulation/Shell Proportions**
- § **Speeding Development**

Future Proofing

Although computer platforms have changed and are continuing to change significantly over the years, 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 and I can maintain a comprehensive product range. Two developments illustrate this. One is a simple, short marketing simulation (Product Launch) that was originally developed in 1977. One current user is a major management school on its executive MBA. A second example is a complex sales management simulation (SMITE). Developed in 1984, it was moved into the shell and customised for use in the American 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 situation occurred for the Profess simulation after a year's use when the client changed their business focus and strategy, Because the reports and decisions are held in the Reporting Database it took less than half a day to realign the simulation.

Customising

The need for customisation 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 (Service Challenge) was customised for use by the Football Association by changing the market and resource terminology in the databases in a few minutes.
b) A simulation (Executive Challenge) 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 (Retail Challenge) was customised for a US 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. (See Appendix 2 for details.)

Speeding Development

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

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

Table II: Competitors' Design Times

Notes

1. Chadwick, Jonathan (2002) Integrating a New Strategy and Developing Key Performance Indicators Business and Simulation Games Conference, London
2. de Hoog, Robert (2002) KITS A Knowledge Management Simulation Game Business and Simulation Games Conference, London
3. *E-Learning Simulations:* Tools and Services for Creating Software, Business, and Technical Skills Simulations (2002) Brandon-Hall.com
4. Summers, Gary J. (2003) The Business Simulation Industry

Simulation	Description	Development Hours/Hour	Model %
SEED	Entrepreneurial Planning[1]	60:1	16%
Foundation Challenge	Not-for-Profit Business Appreciation[2]	25:1	8%
Constructive Negotiation	Sales Negotiation [3]	10:1	2%

Table III: Design times using the shell

Notes:

1. A one-day simulation developed in partnership with Imperial College of Science, Technology and Medicine.
2. A one-day simulation developed on behalf of Henley Management College for the Wellcome Trust
3. Role-play supported by simulation model developed for Carillion PLC

SUMMARY

To summarise and generalise, structured innovation consists of the following steps:

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

REFERENCES

Ballard, Glenn (2000) Positive vs Negative Iteration in Design, Proceedings Eighth Annual Conference of the International Group for Lean Construction, IGLC-6, Brighton, UK, July 17-19.

Cloke, Bill (2000) Lean Products in Advanced Manufacturing Magazine, March 2000.

Guindon, R. (1990) Designing the Design Process: Exploiting Opportunistic Thoughts Human-Computer Interaction, 1990, V5, pp. 305-344

Hall, Jeremy (1978) Marketers and the Electronic Chip in Marketing Magazine October 1978, pp 69-73.

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

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

Hall, Jeremy 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, Kogan Page, London, pp. 76-88.

Hall, Jeremy 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, Kogan Page, London, pp. 76-88.

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

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

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

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

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

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

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

AUTHOR

Jeremy J. S. B. Hall, Churchill Fellow
Hall Marketing,
Studio 11, Colman's Wharf, 45 Morris Road, London, E14 6PA, UK
Phone +44 (0)20 7537 2982
E-mail jeremyhall@simulations.co.uk
Web www.simulations.co.uk

APPENDIX 1 - Common (Shell) and Singular (Simulation) Components

The Shell

This consists of five main program components:

- **Simulation Manager**
- **Display Engine**
- **Hypertext Help Engine**
- **Reporting Engine**
- **Decision Entry Engine**

And three data files

- **Constants File**
- **Text File**
- **Help Database**

The **Simulation Manager** drives the simulator and customises the shell to a particular form of simulation. The simulation forms are as follows:

- Direct Use
- Tutor Mediated
- Decision Support
- Planning
- Negotiation Role-plays

Direct Use simulations are those where individual teams run the simulation directly on their own microcomputer. **Tutor Mediated** simulations are those where the trainer processes decisions on behalf of the teams of learners. **Decision Support** simulations are those where it is appropriate for each team to have their own decision support tool. **Planning** simulations are those where instead of running a business for several time periods, teams use the simulation model to prepare a business plan. **Negotiation Role-plays** utilise a simulation model to assess the costs of proposals in a negotiation.

Display Engine is common to all shells and formats output for display on the screen or printing.

Hypertext Help Engine is common to all shells and is used to extract hypertext records from the Help Database, formats the text, pictures and music and sends to the Display Engine

Reporting Engine is common to all shells and is used to generate the business reports provided to the learners and the trainer. It takes data from the Parameter or Comments Database and report layout information from the Reporting Database and formats these before sending to the Display Engine.

Decision Entry Engine is common to all shells and is used for decision entry and manages the entry of decisions. The Reporting Database defines what data is to be entered and when this is done and checked it stores the data in the Parameter Database for use by the Simulation Model

Constants File is used to store constants used by the simulation shell.

Text File is used to store the standard texts used by the simulation shell and different files are used for different languages.

Help Database is used to store the help associated with the simulation together with the help, advice and explanations that is specific to a simulation.

The Simulation
A particular simulation is produced by creating a **Simulation Model** and inserting it into the shell and providing a **Control File** and **Parameters, Comments, Reporting and Help Databases.**

The Control File is specific to a version of the simulation. But, except where different terminology or languages are involved, the databases are usually common to all versions of the simulation.

APPENDIX 2 - Simulations

Simulation	Platform	Duration	Date	Purpose	Model Size
1. Teamskill[1]	MSDOS	1.5 days	1970	Operations Management	11%
2. Management Experience[1]	MSDOS	1 day	1976	Tactical Management	7%
3. Product Launch[1]	MSDOS	2 hours	1977	Marketing Concepts	4%
4. Market Strategy[1]	MSDOS	4 hours	1978	Marketing Planning	3%
5. Global Operations[2]	MSDOS	1 day	1980	Strategic Management	9%
6. Operations[2]	MSDOS	3 hours	1981	Operations Concepts	3%
7. Sales Calls[2]	MSDOS	3 hours	1983	Selling Concepts	2%
8. Sales Negotiation[2]	MSDOS	1 day	1983	Sales Negotiation	2%
9. Sales Mix[2]	MSDOS	3 hours	1983	Team Behaviour	5%
10. SMITE[2]	MSDOS	2 days	1984	Sales Management	17%
11. Financial Analysis[3]	MSDOS	4 hours	1985	Financial Appreciation	2%
12. Management Challenge[3]	MSDOS	1 day	1986	Business Appreciation	7%
13. Retail Challenge[3]	MSDOS	1 day	1987	Business Appreciation	8%
14. SMART[3]	MSDOS	1.5 days	1987	Marketing Strategy	9%
15. CISCO[3]	MSDOS	2.5 days	1988	Strategic Management	15%
16. RESERVE[3]	MSDOS	1.5 days	1988	Business Appreciation	9%
17. Service Challenge[3]	MSDOS	1 day	1989	Business Appreciation	8%
18. Commercial Negotiation[3]	MSDOS	1 day	1990	Sales Negotiation	2%
19. Distribution Challenge[3]	MSDOS	1 day	1993	Business Appreciation	8%
20. Executive Challenge	MSDOS shell	1day	1996	Business Appreciation	8%
21. PROFESS	MSDOS shell	1day	1998	Business Appreciation	8%
22. FINESSE	MSDOS shell	2 days	1999	Strategic Management	12%
23. Business Focus	MSDOS shell	4 hours	2000	Selling Concepts	4%
24. Constructive Negotiation	Windows shell	1day	2002	Sales Negotiation	2%
25. Foundation Challenge	Windows shell	1day	2002	Business Appreciation	8%
26. SEED	Windows shell	1day	2002	Entrepreneurial Planning	16%

Notes:

1. Originally developed for a Computer Time-Sharing platform and then transferred to first generation microcomputers and then MSDOS.
2. Originally developed for first generation microcomputers (Tandy Models 1 & 3, Apple 2) and then transferred to MSDOS.
3. Originally developed for MSDOS (PCs).