Designing Simulations for Learning

Vivek Williams
formerly of the University of Southern Queensland
Toowoomba, Queensland Australia
Email: vivek.williams@townshend.cz

Introduction

When creating an interactive simulation specifically geared towards the achieving of learning objectives, there are a number of interface design options that need to be chosen carefully. Frequently these decisions are made ‘intuitively’ by designers who resort to the traditional guidelines of ‘direct manipulation’, without adequate consideration of how overall cognitive processes may be affected. Results of several studies have indicated that, to the contrary, interfaces that are less ‘user-friendly’ may, in some instances, be more conducive to concept learning.

This paper examines such research and associated theory. Major features of a simulation-learning environment are examined systematically, and, in each case, implications for design are discussed and summarized. The framework used is one that places the interface at the centre of interactive learning, mediating as an information flow portal between human cognition and a programmed model. Variations in the directness of manipulation and timing of feedback are highlighted as vital parameters, and, in concluding, a call is made for further research to elicit the effects of such changes specifically on concept acquisition.

Briefly: Simulation and Interaction

> *“For better or worse, simulation is no mere fad. Indeed, to think of simulation games as mere entertainment or even as teaching tools is to underestimate them. They represent a major addition to the intellectual repertoire that will increasingly shape how we communicate ideas and think through problems.”* (Starr, 1994)

Since the popular adoption of the PC for educational and recreational purposes, the terms ‘simulation’ and ‘interaction’ have attributed numerous and diverse meanings and have often been used in contexts indicating beneficial uses of the new technology. When thinking about simulations, people reflexively cite examples such as Simcity (Maxis, 1989) and Microsoft Flight Simulator (Microsoft, 2000).

In computer-based learning any system can be simulated, providing that its relevant attributes can be expressed in terms of an algorithmic model. The system being modelled may be derived from the 'real-world' or based primarily on fantasy. Generally, in the educational setting, attributes of physical, social, economic and political domains are replicated in simulation models. For the purposes of this discussion, the following definition will be used:

A simulation is a modelled interactive environment that reflects particular attributes of another system.

The term 'interactive', by virtue of its literal definition: *'mutually or reciprocally active'* (Webster's, 2001), has been widely used in sales pitches to describe software ranging from simple productivity tools to large multimedia encyclopaedias. For applications dedicated to learning, this broad, somewhat jingoistic definition has proven to be totally deficient, and as a result, several taxonomies have been formulated in an attempt to afford pedagogical significance to aspects of interaction. Whether deep or surface learning occurs with user's involvement with an application, was addressed by Jonassen (1988) in framing his five levels of interaction. The learner's mental engagement was identified as being crucial within three dimensions of interactivity, classified as 'Levels', 'Functions' and 'Transactions' (Schwier & Misanchuk, 1993). More recently, this mental engagement, has been considered in conjunction with the learner's ability to manipulate and navigate through the environment, with calls for greater attention to be paid to the design of context-based interaction (Sims, 1997).

The ubiquitous presence of game consoles bears testimony to the popular appeal and acceptance of the simulation as a vehicle for learning and entertainment. The efficacy of adopting simulators in education and training has been well studied (Carlsen & Andre, 1992; Randel, Morris, Wetzel, & Whitehill, 1992; White, 1993), and although it is not strongly conclusive whether concept learning is improved by the use of simulators as compared to traditional methods, there certainly does not appear to be evidence to the contrary.

> *"Simulated experiences have the potential to become powerful instruments of cognition. They support both experiential and reflective processes: experiential because one can simply sit back and experience the sights, sounds, and motion; reflective because simulators make possible experimentation with and study of actions that would be too expensive to try in real life."* (Norman, 1993)

On a cognitive level, sensory input (usually visual, auditory or haptic) is frequently updated in response to user initiated-commands. This results in a feedback loop that is controlled and guided by the learner, through which they are able to observe changes, make inferences, and test emerging ideas about the model in question. Pedagogically, the process appears ideal, however, concerns remain.

> *"Tim's approach to 'SimLife' is highly functional. He says he learned his style of play from video games... Tim is able to act on an intuitive sense of what will work without understanding the rules that underlie the game's behavior. His response to 'SimLife'—comfort at play, without much understanding of the model that underlies the game—is precisely why educators worry that students may not be learning much when they use learning software."* (Turkle, 1997)

What is essentially suggested here is that computerized models can be so complex that the user has little to no hope of understanding the relationships between key underlying attributes. The 'black box' concept (Starr, 1994) of the algorithm controlling the simulation, has been the cause of much concern, where users are unable to query or debate crucial assumptions forming the basis of the program. This criticism highlights the point that designing for educational impact requires the consideration of several layers, from the algorithm to the interface, to the nature of the learning task itself.

A model that defines these layers as well as represents their juxtaposition in an interactive environment is the essential starting point for any systematic design process.

A 'Conveyor' model

Represented in the diagram below is a model where the interface assumes a central position, with the human and program at opposing poles.

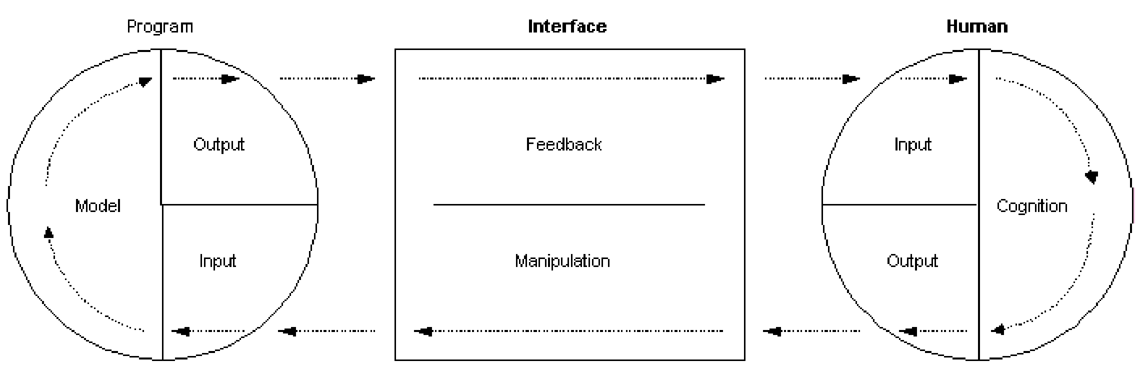

Diagram1: Conveyor model overview

Beginning the process at the section representing human input, the learner usually sees or hears a stimulus from the interface – this undergoes integration in a cognitive process – the resultant is an output such as a planned course of action – and so certain elements of the interface are manipulated – resulting in the definition of the inputs for the underlying program – which are then processed by program algorithms defining the model – creating output data – that the interface represents as feedback to the user – and so the cycle continues.

Using this framework, the subsequent sections examine the human, interface, and program ends, particularly placing emphasis on the design options available at each, and their significance within the system.

The Human End

Learning processes

Human sensory input occurs through sight, hearing, touch, taste, smell and vestibular mechanisms. Most commonly the design of a computer interface is concerned with sight and hearing, with a small but steadily growing industry dedicated to haptic (touch) feedback. Visual input results from the attention a user gives to the interface. It is needed to guide actions (e.g. hand movements), to make decisions about actions, and most significantly in this case, to learn about the behaviour of an underlying model.

A learning process can be defined as *“cognitive transactions of the learner that are meant to transform information into knowledge”* (Goodyear, Njoo, & Hijne, 1991). The definition is largely born out of an information-processing approach, which grew from theory that was a result of research within the field of Artificial Intelligence, where much endeavour centred around creating programs able to simulate human cognitive processes.

The learning theory that is often applied to simulations is ‘discovery learning’. Described formally by Bruner (1961), interest in discovery learning became resurgent after the advent of personal computers in the early 1980’s. This was largely as a result of the computer’s data processing power enabling educators to simulate aspects of an environment that were previously unreplicable, and also because of the increasing educational emphasis being placed on constructivist approaches (Jonassen, 1991). Since then, several variants and elaborations of the basic theme have been proposed (Joolingen & Jong, 1997; Klahr & Dunbar, 1988; Qin & Simon, 1990).

In discovery learning, one of the ways in which learning can take place is when the subject reflects on the outcomes of their actions, makes inferences and verifies them with further action. This is implicit within the notion of a simulation. De Jong & Njoo (1992) identify four features for the instructional use of simulations: *the presence of formalised, manipulable underlying models, the presence of learning goals, the elicitation of specific learning processes, and the presence of learner activity*. The human end is concerned with the latter two. The learning processes described by those authors includes hypothesis generation, prediction and model exploration.

It has been suggested that there are two fundamental 'problem spaces' (Klahr & Dunbar, 1988) that are searched in the process of discovery learning, the 'hypothesis' space and the 'experiment space'. The hypothesis space is essentially a bank of possible hypotheses relating to the problem, which the learner searches in order to explain what is observed. It may be informed by prior knowledge and the outcomes of experimentation. The 'experiment space' is a bank of possible experiments that may be conducted, and may or may not necessarily be guided by a relevant hypothesis. The model proposed by the same authors is called 'Scientific Discovery as Dual Search' (SDDS), and features the key processes of searching the hypothesis space, testing the hypothesis and evaluating the evidence.

In relation to computer simulations Reimann (1989) suggests an inductive model of learning featuring the following phases:

- Testing and modifying the hypothesis;
- Designing an experiment;
- Making a prediction;
- Evaluating the prediction;
- Evaluating and/or modifying the hypothesis.

It largely equates to the SDDS model, with the addition of a 'prediction' phase, seen as being distinct from the hypothesis. However, the process is still considered to be an iterative one, whereby the underlying model is progressively discovered by the learner through sequential modification of an hypothesis. Ultimately, the goal of this discovery process is for the learner to establish a hypothesis, or set thereof, that accurately reflects the conceptual model being simulated.

Discovery learning has at its core the notion of an iterative cycle, represented in the following diagram:

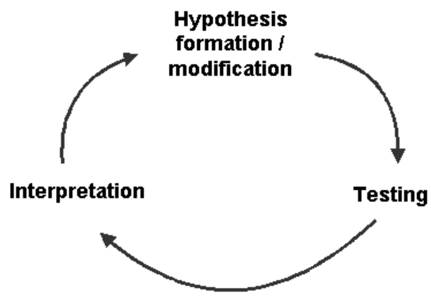

Diagram 2: Discovery learning cycle

If the environment permits, in the early part of exposure to an interactive system the subject undergoes an orientation, where familiarity is gained with controls, feedback areas and other relevant features of the interface. After this initial period, the user may start to apply themselves to the particular task, progressively acquiring 'domain' knowledge in the process. If the goal of arriving at a hypothesis/set aligned to the programmed model is to be achieved, the learner must progressively advance from a rudimentary understanding of the system to a refined notion of what it represents. Subsequently, as domain knowledge builds and becomes more sophisticated, the learner engages in deeper reflective thought processes, which may be manifested in less frequent and more deliberate interactions.

Considering the discovery cycle in the diagram above as being that of a 'top down' or aerial view of the process, viewing it from the 'side' shows how the series of iterations may be represented as a function of time.

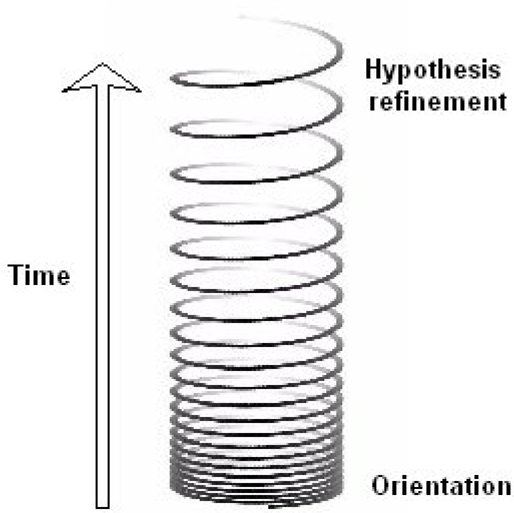

Diagram 3: Discovery iterations as a function of time

The ‘spring’ is compressed during orientation, followed by the relative infrequency of iterations/coils during hypothesis refinement. It is over-simplistic to suggest that this is a universal pattern of interaction, as differences in individual learning behaviour account for a diverse range of approaches. Klahr and Dunbar (1988) suggest that users apply themselves as ‘experimenters’ or ‘theorists’, signifying predominant activity within experiment or hypothesis spaces respectively. ‘Experimenters’ would demonstrate a greater number of iterative cycles compared to ‘theorists’. Orientation, also, may or may not be extensive, depending on how familiar the user is with the system, and how intuitive the interface appears to them.

Information processing limitations

Within an interactive simulation the flow of activity taking place during learning can be considered in terms of an information-processing model. Classically, the approach categorizes memory into sensory, short-term, and long-term stores. Information enters initially via the senses (sensory memory) and then proceeds to short-term memory and subsequently, under certain conditions, to long-term memory (Atkinson & Schiffrin, 1968).

Over the past fifty years researchers in the field of Artificial Intelligence (AI) have attempted to simulate aspects of human cognition using computer systems. The models created are based on the ‘declarative’ and ‘procedural’ knowledge dichotomies that are allied to similar distinctions in other theories of knowledge and learning. Piaget’s separation of ‘concepts’ and ‘schemes’ (Piaget, 1952), and Schema-theory’s ‘objects’ and ‘events’ (Gagne, 1985) are examples.

Although popularly adopted as a model of cognition, some authors have argued that short-term and long-term memory are essentially part of a single memory system. The ‘*levels of processing’* framework (Craik & Lockhart, 1972) suggested that information is processed at different levels concurrently, depending on its particular characteristics, and deeper processing results in more information that can be remembered. These deeper levels require analysis of meaning, which could involve thinking of associations, images, and past experiences.

Short-term memory, from the Atkinson-Schiffrin model, has two important limitations. Firstly, it can hold at any given time 7 (+/-) 2 "chunks" of information (Miller, 1956). Secondly, its holding ability is approximately 20 seconds. Such limitations appear to be of limited relevance within the context of a simulation based on reasoning, where there is little need for the recall of strings of words or numbers. Aspects of greater significance can be found in the contemporary cousin of short-term memory, ‘working memory’ (Baddeley, 1986). This is considered as a dynamic system, active in the execution of higher-level cognitive tasks such as learning and reasoning. It does so by being a system for the temporary storage and manipulation of information via two types of components: a *storage* and a *central executive*. The storage system is considered to be passive, and mainly responsible for the transient storage of verbal and visual information, whilst the central executive, is regarded as being actively involved in encoding, storing, and retrieving information. The concept of a central executive (Baddeley, 1990) was preceded by the comparable *supervisory attentional system* (Norman & Shallice, 1980), which also was seen as having limited capacity, and active in tasks involving decision making and problem solving. Demands on resources vary during the learning process; for example, the executive function is likely to require greater processing power when a subject is presented with a new task or environment, as compared to when they perform familiar routines.

The notion of limited capacity is fundamental when designing activity flow as represented by the conveyor model. If resources are limited and need to be shared by the demands of a storage and central executive, then it must bear consideration that a balance should exist between any new inputs and the time and resources needed to process them. In a simulation, a prime mediator for establishing this balance is the interface.

The Interface

Keyboards, monitors, voice activation, and mouse devices are classically considered as the interface between man and machine. In the Conveyor model, rather than focusing on hardware, the interface is viewed as the software intermediary between the human and the model.

Graphical metaphors constitute the most common way to contextualize a simulation for the user, and are often used to implicitly convey a basic paradigm of operation. Most simulations rely on visual representations to provide the setting, and display feedback and manipulation environments.

Emphasis placed on the realism, or fidelity, of a simulation has resulted in increasingly rich graphical environments that aim to reproduce the look and behaviour of an alternate system (usually a real-world system). The term ‘virtual reality’ conjures up the notion of being totally immersed in an artificial world, where the user moves and acts as they would in ‘real reality’. Fidelity has been categorized as being *physical* and *functional* (Hays & Singer, 1989), where physical fidelity refers to how authentic the interface feels through manipulation and feedback, whilst functional fidelity is a measure of how faithfully the system being simulated is represented by the model.

In simulations where skilled operation is a desired outcome, high levels of fidelity will result in the user being able to more readily transfer learned procedures to the real working environment. High fidelity may also be an important factor in learner motivation, and in most cases, where the costs and technology permit, it is recommended as good design practice (Reigeluth & Schwartz). However, as is discussed below, at times it may prove beneficial to sacrifice high physical fidelity for an interface that promotes planning and reflection.

Manipulation

Manipulation of an interface usually takes the form of operating sliders /dials /buttons (using keys or a mouse), or entering numbers and characters within specified fields. The term 'Direct manipulation' was coined by Ben Shneiderman, and essentially features the following three criteria (Shneiderman, 1998):

1. Continuous representation of the object of interest.
2. Labeled button presses used instead of command line syntax.
3. Operations whose impact on the object of interest is immediately visible.

These guidelines, due to their obvious synergy to real life experience, became fundamental parameters in developing any graphical user interface. Norman (1988) identified the *'gulf of execution'* which refers to the distance, or difference, between one's intentions to the actions that must be carried out in acting through the interface. Bridging this gulf through direct manipulation has been established for several years as a fundamental tenet of good interface design, and it follows logically that the case should be no different when constructing an educative simulation.

With this assumption in mind, and therefore somewhat counter-intuitively, findings of recent studies (Golightly, 1996; Schär, 1996; Svendsen, 1991) revealed some evidence to the contrary. Results indicated significant improvements to problem solving performance when using less direct forms of manipulation, such as command line interfaces, or having to act on alternate representations of the object needing manipulation.

Researchers attempted to explain these effects in a number of different ways. Svendsen (1991) proposed that the verbalization employed during the use of a command line interface resulted in subjects developing a deeper explicit knowledge of rules, therefore resulting in less moves taken to problem completion. Alternate explanations have suggested that increased 'implementation costs' lead to a greater 'planfulness' during the problem-solving process (O'Hara & Payne, 1998).This means that from a perceptual-motor standpoint, the burden to the user of using a command line is significantly greater than the click of a mouse, and hence, to avoid the excessive execution of such operations, the user chooses to plan each move more carefully. This corresponds to the ‘rational analysis’ of ACT-R theory (Anderson, 1993), where it is proposed there exists a cognitive tradeoff between maximizing goals and minimizing implementation costs. In another study (Fu & Gray, 2000) involving subjects placing blocks in particular pre-determined configurations, constrained by visual and memory cost factors, this tradeoff between memory costs and perceptual-motor costs, predicted by Anderson’s rational analysis, was again supported.

Limiting the number of key presses or moves, as well as providing set goals for interactions have also shown to improve overall learning of an interface (Trudel & Payne, 1995). The studies indicate that if manipulation is unlimited and too easy to perform, the user will tend to operate without thinking enough about the process. A similar condition may eventuate when learners fail to adequately establish goals for the interaction, and spend excessive amounts of time ‘roaming’ the interface. Although some exploration is essential for orientation within the environment to take place, aimless interaction beyond a certain point could lead to boredom, frustration and even abandonment of the system.

Some studies (Rieber, Tzeng, Tribble, & Chu, 1996) have attempted to evaluate concept learning in a simulation within the context of dual-coding theory (Paivio, 1991). Interface design complementing the referential processing suggested by this theory was shown to significantly enhance the explicit understanding of the physical scientific principles under review. Manipulation of the interface varied from visual representations of objects to numeric displays.

Given the implications of these studies, the following items would be worthy focus points for discussion when designing interface manipulation:

- Establishing a goal or set of goals for the interaction that will guide the manipulations of the user;
- Allowing for a conversational interface (i.e. typing in words or commands) if verbalization could assist in concept learning;
- Providing opportunity for orientation early in the user’s contact with the interface;
- Imposing ‘costs’ or burdens on actions to stimulate reflection at key times.
- Costs may include key press/ move limits, time constraints, manipulation of alternate representations of an object, or deliberately cumbersome procedures.

It will become apparent, from the ensuing section, that none of these summary items can be detailed in isolation, since they are also integral to establishing the feedback parameters of a system.

Feedback

The feedback most commonly encountered in learning simulations is visual feedback, and can be categorized as synchronous or asynchronous. Synchronous feedback is ‘real-time’ in nature, and changes instantly, in synchrony, with user manipulations of the interface. It is faithfully representative of our movement in the physical world, as our actions reveal immediate and visible consequences. Asynchronous feedback, on the other hand, is feedback that is delayed or modulated in some way. It is instituted by programmed time delays or additional operations that the user must perform to reveal the outcome of previous actions.

As stated previously, direct manipulation requires as one of its *conditions ‘Operations whose impact on the object of interest is immediately visible’*, whilst Norman’s (1988) second gulf, the *‘gulf of evaluation’,* refers to the difficulty a user has in determining whether goals have been achieved. These have been well-established guidelines for interface design for several years.

Thus, at face value, the design decision for feedback in simulations appears to be quite straightforward: provide feedback that most closely resembles what happens in the system being simulated. However, again, as with manipulation parameters, there is evidence to suggest that this does not necessarily produce optimal learning outcomes.

Some research has demonstrated that synchronous feedback (sometimes termed continuous feedback) can result in the inducement of an implicit learning mode, whilst asynchronous feedback (similarly termed discontinuous feedback) may be optimal for the production of declarative knowledge (Schär, Schluep, Schierz, & Krueger, 2000).

A skill-based simulation has the intention of promoting reflexive actions, experiential processes (Norman, 1993) or procedural skills through implicit learning (Reber, 1992). ‘Shoot-em-up’ arcade games are good examples. Synchronous feedback is desirable in such instances, where skills are mainly developed ‘unconsciously’ during the process of interaction. This type of implicit learning is often illustrated by a person learning to ride a bicycle, where they respond reflexively, but cannot articulate explicitly what knowledge has been acquired.

Concept-based simulations have an underlying model, informed by observed relationships, and defined by rules, conditions and actions, which need to be explored and uncovered by the learner. These types of simulations promote reflective processes (Norman, 1993), explicit learning (Reber & Squire, 1998) and the formation of declarative knowledge. Asynchronous feedback can thus be effectively employed to create enforced delays, encouraging the learner to engage in the deeper thought processes demanded by the simulation.

Modulation of feedback also decreases the concurrent information being processed by the user. A learner fully focused on planning and executing a manipulation may be distracted by a continuous stream of feedback, especially if it is visually or audibly intrusive. Reducing the effects of this interference is another benefit of an asynchronous feedback loop.

The application of asynchrony in feedback can be achieved in several ways. The user entering data into a field may be required to press ‘enter’. In addition, there may be a time delay before the result is made evident. Alternately, whilst dragging a slider control with the mouse, change in output would only be displayed upon reaching the ‘mouse-off’ state. Another method could be to necessitate the use of a ‘show me’ button, which the user clicks on to find out the result of a previous series of manipulations.

Studies, grounded in the framework of information-processing theory, have primarily sought to explore the effects of the nature and timing of feedback in computer-based instruction (CBI). The type of feedback addressed in CBI studies ranges from that associated with discrete student responses to the more informative feedback used to enlighten the student on their progress towards a particular goal. It is common to find reference to two forms of feedback (Kulhavy & Stock, 1989), ‘verification’, whether an answer is right or wrong, and ‘elaboration’, the provision of guidance to the learner. Similarly, Overbaugh (1994) described four such levels of feedback as lying on a continuum of usefulness for learning enhancement.

Although superficially appearing to differ from the types of feedback present in simulations, there are several valuable parallels that can be drawn. CBI research has shown that immediate feedback, which could be likened to synchronous feedback, may be more effective for lower-level knowledge acquisition (Gaynor, 1981). Conversely, delayed feedback, allied to asynchronous feedback, has proven more effective for the comprehension of higher level concepts (Jonassen & Hannum, 1987).

In summary, for a concept-based simulation, options for the design of feedback should include:

- Encouraging explicit learning modes by delaying feedback.
- Designing delays by displaying the resultant of a manipulation only upon ‘mouse-off’, by clicking a ‘show me’ button, or by time lapse.
- Reducing interference by the spatial separation of visual feedback from manipulation areas, coupled with an unobtrusive representation of outputs.

As was mentioned in concluding the preceding section, these options need to be closely considered in conjunction with the types of manipulation envisaged for the simulation. For example, if substantial ‘costs’ have been already imposed by instituting a command line interface, it may be counter productive to further create delays by the addition of a 'show me' button. In many such instances it is possible to obtain the benefits of asynchrony and manipulation costs by the design of a single feature.

The Program End

Learning simulations have at their core a set of algorithms or models that reflect the operation of another system. The distinction between these rules and the supporting software that they are programmed in needs to be clearly made. The model represents the fundamental conceptual basis of the interaction, whilst the software simply 'houses' or facilitates the process.

A model is composed of variables and their mutual relationships, and it is the discernment of these variables and relations that is the heart of a concept-based simulation.

In further developing SDDS, attempts were made (Joolingen & Jong, 1997) at describing the structure of the hypothesis and experiment space as well as the search within these spaces. The hypothesis space was subdivided into spaces for 'variables' and 'relations', variables were ordered from general to specific within a hierarchical tree structure, and relations were similarly represented according to their level of precision. The diagram below displays the structure.

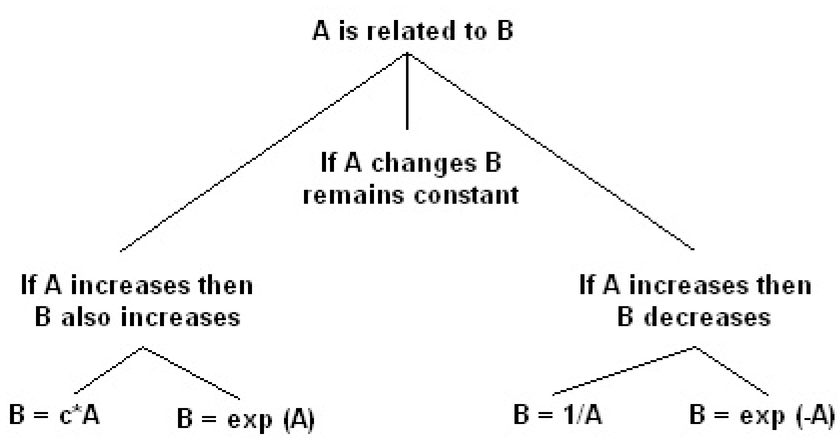

Diagram 4: Example of a relation hierarchy, adapted from (Joolingen & Jong, 1997)

In searching the hypothesis space the learner first constructs a set of hypotheses by searching both the variable and relation spaces. Subsequent search operations continue to be characterised by activity in both these spaces as well as those that alter the set itself. The authors proceed to categorize each type of search operation in a classification that is representative of the various parameters of a hierarchical tree structure, for example, *'abstraction of a hypothesis'* is seen as a relation space search operation, and is described as moving from a more to a less precise relation. This equates to moving 'up' within the tree structure.

The experiment space is seen as consisting of 'value-tuples', which are sets of variables with corresponding values assigned to them. These values may be numeric or qualitative in nature. In searching this space the learner first chooses the variable/s to be altered followed by the allocation of a value to them.

It is self evident from the limited processing capacity available to a human during interaction that the number of variables and the complexity of their relationships constitute essential elements in the design of a model. However, greater complexity of this nature does not automatically suggest greater 'difficulty' for a learner, as will be discussed when considering the inextricably linked notion of 'intention'.

Intention

The 'Tower of Hanoi', invented by the French mathematician Edouard Lucas in 1883, is a popular puzzle task that requires users to transfer discs from the left (Peg 1) to the right (Peg 3).

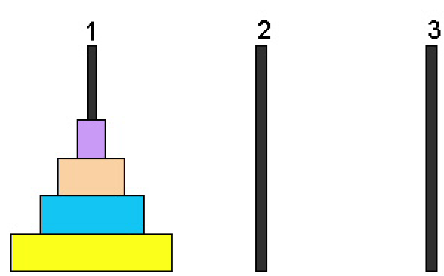

Diagram 5: The Tower of Hanoi Puzzle

The rules simply state that only one disc can be moved at a time, and a larger disc may not be placed on a smaller one.

Applied through the medium of a computer interface, the intention for which the puzzle is presented significantly determines the learning outcome for the user. For example, if rules are explicitly stated initially, and the user is simply asked to complete the puzzle, the learning outcome may be a rote memorization of sequence. If the rules are not disclosed (i.e. during the session an illegal move results in a beep, and the disc being returned to its original position), an additional learning outcome may be the discovery of the puzzle constraints. If the goal is to complete the puzzle in a minimum number of steps, a learning outcome may be the rote memorization of an ideal sequence derived through trial and error, or the formulation of an optimal algorithm.

The intention of any task or interaction cannot be formulated in isolation of the developmental and situational contexts of the users. Developmental issues include language and logic sophistication, especially when designing interactivity for young children. Situational contexts refer to the association of domain-specific knowledge to the problem, as well as to the setting and environment of the simulation.

Using the example of Simcity (Maxis, 1989), the intent for novice users may just be a superficial appreciation of a growing city, whilst students of economics may be required to uncover relationships between unemployment and crime.

Similarly, advanced computer programming students presented with the Tower-of-Hanoi problem during a lesson on algorithm production would differ considerably from primary school children in terms of the scope and depth of understanding puzzle solution paths. The intention may be for them to derive that the minimum number of steps needed to complete the puzzle equals $2^n - 1$, (where n is the number of discs present), whereas the child could simply be undertaking a recreational exercise in logic.

Largely, task intentions can be categorized as those promoting model operation or model conceptualization, which equate to the previously discussed skill-based and concept-based simulations respectively. It may be argued that in many instances model conceptualization is necessary for skilled model operation, and, less frequently, this could be conversely true. Any single ‘learning simulation’ may contain multiple or combinations of task intentions. For example, the same flight simulator may be used for operational training (e.g. sequences for takeoff and landing, and associated skills), but may be also used for teaching theory (e.g. lift/drag ratio problems). The former has the intent of model operation, the latter model conceptualization.

Learners will often attempt to achieve a particular end-goal during a simulation rather than theorizing about the model (Kuhn & Phelps, 1982). For example, in a rocket fuel simulator, they may have a goal of sending the rocket into orbit without necessarily understanding the chemical relationships of the fuel mixture. This has been referred to as an ‘engineering approach’, in contrast to a ‘scientific approach' where users systematically uncover model rules and relationships (Schauble, Glaser, Duschl, Schulze, & John, 1995). Therefore, a learning simulation may utilize operational tasks to illustrate particular behaviours of the system, in conjunction with conceptual exercises that encourage hypothesis formation.

The model operation/conceptualization dichotomy is summarized below in a table that generalizes for each; the cognitive processes involved, possible outcomes, and how these outcomes may be measured.

Task intention:	**Cognitive processes:**	**Outcomes/objectives:**	**Measured by:**
Model operation	Memory, visualization, sequence logic, planning	Efficiency, productivity, recreation, skilled system operation by being able to carry out tasks in a particular system or environment. Proficiency in model operation. Knowing *how* the model works.	Time, moves, effort
Model conceptualization	Prediction, rule induction, cognitive restructuring, insight	Hypothesis formation, understanding of concept. Generalisation to other situations and contexts. Discovery of variable/relations behaviour. Understanding *why* the model works.	Explanation and articulation of rules and concepts

Table 1: Task intention summary

In summary, some of the design issues, relevant to the program end, that require consideration include:

- Definition of the model/s to be simulated by identifying variables and relations.
- Balancing variable and relation complexity with task intention and the user's developmental stage.
- Considering situational contexts by providing sufficient support material prior to and during the interaction.
- Creating a set of guided goals to facilitate progressive discernment of model properties.
- Matching task intention to the desired outcomes.

Evaluation of the learning outcomes of a simulation can indicate how further improvements may be made to subsequent versions. With operational systems it is relatively easy to measure the time taken, or the number of moves performed by users in reaching a set goal. However, to evaluate concept simulations, the designer would need to rely upon an accurate analysis of the interaction itself, or require users to attempt formal tests of their knowledge.

Conclusions

There is a need for further research aimed specifically at the effects of simulation interface design on concept learning. Typically, research studies have employed well-defined problem solving tasks such as the ‘Tower of Hanoi’ or the ‘Eight-puzzle’ to test the effects of adjusting interface features (Golightly, 1996; Svendsen, 1991). Conclusions drawn suggest that success in the set activities was characterized by the extent to which learners reflected on their actions.

It seems plausible then to extend this notion to the learning of concepts in a simulated environment. However, there are fundamental differences between the attributes of a puzzle tasks such as the Tower of Hanoi and those of concept-based simulations that could impact on this assumption. Some significant disparities are summarized in Table 2.

Puzzle tasks	***Concept simulations***
Puzzles have an obvious end goal or solved state that the user endeavours to reach.	No end-goal may be present. Users may work with several goals to develop concepts.
Rules of operation are usually understood prior to engagement.	Rules are usually ‘uncovered’ by the user during interaction.
Puzzle behaviour is not modelled on another system.	Behaviour is based on a model that is an abstraction of a real-life system.
After a move, no information about the model’s behaviour is discerned (unless rules are withheld).	Manipulations result in new information becoming available to the user.
A move results in a physical state that could have been predicted by the user prior to making the move (unless rules are withheld).	The accuracy of user’s prediction will be representative of their understanding of the model’s behaviour.
‘Thinking ahead’ several moves is possible, requiring the use of memory.	It is not possible to plan several moves ahead, as each iteration provides new information.

Table 2: Puzzle tasks vs. concept simulations

Performance can be measured by fewer moves or the reduced time to complete certain tasks, and it has been shown that greater ‘planfulness’ results from imposing costs on interactions, leading to greater efficiency (O'Hara & Payne, 1999). What is yet unclear is how concept development, rule induction, or ‘insight learning’ (cognitive restructuring) is affected by similar conditions.

This article has deliberately not set out to review existing simulations and their common applications, or to develop yet another taxonomy based on a new set of criteria. It has intended to identify key components within the framework of a conveyor model, significant research relating to each component, and relevant design issues thus arising.

Blindly accepting the pedagogical applicability of the tenets of direct manipulation has been shown to be inadequate when making design choices aimed at maximizing learning outcomes. No simple rule of thumb can be applied to give an optimal set of interaction parameters, if fact, a common suggestion throughout has been that any aspect of the Conveyor system cannot be addressed adequately when viewed in isolation. This was highlighted when considering synchronous vs. asynchronous feedback at the level of interface manipulation, as subtle variations to manipulation design automatically and inherently alter the nature of feedback timing.

Perhaps the most significant overarching concern binding all design facets together is that of educational context. If task intention, and users' situational and developmental stages are not clearly defined, the designer can be faced with creating a costly simulation that may ultimately prove to be of limited pedagogical value, and simply serve as yet another ornamental offering to the demands of 'technology in education'.

References

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, N.J.: L. Erlbaum Associates.

Atkinson, R. C., & Schiffrin, R. M. (Eds.). (1968). *Human memory: a proposed system and its control processes* (Vol. 2). London: Academic Press.

Baddeley, A. D. (1986). *Working Memory*. Oxford: Clarendon Press.

Baddeley, A. D. (1990). *Human memory: Theory and practice*. Oxford: Oxford University Press.

Bruner, J. S. (1961). The act of discovery. *Harvard Educational Review, 31*, 21-32.

Carlsen, D. D., & Andre, T. (1992). Use of a microcomputer simulation and conceptual change text to overcome students preconceptions about electric circuits. *Journal of Computer-Based Instruction, 19*, 105-109.

Craik, F., & Lockhart, R. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning & Verbal Behavior, 11*, 671-684.

De Jong, T., & Njoo, M. (1992). Learning and instruction with computer simulations: learning processes involved. In E. De Corte & M. Linn & H. Mandl & V. L (Eds.), *Computer-based learning environments and problem solving*. Berlin, Germany: Springer-Verlag.

Fu, W., & Gray, W. D. (2000). *Memory versus Perceptual-Motor Tradeoffs in a Blocks World Task.* Paper presented at the Twenty-second Annual conference of the Cognitive Science Society, Hillsdale, NJ.
Gagne, R. M. (1985). *The conditions of learning and theory of instruction* (4 ed.). New York: Holt, Rinehart, and Winston.
Gaynor, P. (1981). The effect of feedback delay on retention of computer-based mathematical material. *Journal of Computer-Based Instruction, 8*(2), 28-34.
Golightly, D. (1996). *Harnessing the interface for domain learning.* Paper presented at the Conference on Human Factors and Computing Systems, Vancouver.
Goodyear, P., Njoo, M., & Hijne, H. (1991). Learning processes, learner attributes and simulations. *Education and Computing, 6*(3), 263-304.
Hays, R. T., & Singer, M. J. (1989). *Simulation fidelity in training system design*. New York: Springer-Verlag.
Jonassen, D. H. (1988). *Instructional Designs for Microcomputer Courseware*. Hillsdale, NJ: Lawrence Erlbaum.
Jonassen, D. H. (1991). Objectivism versus constructivism: do we need a new philosophical paradigm? *Educational Technology Research & Development, 39*, 5-14.
Jonassen, D. H., & Hannum, W. C. (1987). Research-based principles for designing computer software. *Instructional Technology, December*, 7-14.
Joolingen, W. R. V., & Jong, T. D. (1997). An extended dual search space model of scientific discovery learning. *Instructional Science, 25*, 307-346.
Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science, 12*, 1-48.
Kuhn, D., & Phelps, E. (1982). The development of problem-solving strategies. In H. Reese (Ed.), *Advances in child development and behavior* (pp. 1-44). New York: Academic Press.
Kulhavy, R. W., & Stock, W. A. (1989). Feedback in written instruction: The place of response certitude. *Educational Psychology Review, 1*(4), 279-308.
Maxis. (1989). Simcity. http://simcity.ea.com/us/buildframes.phtml?guide/about/history: Electronic Arts.
Microsoft. (2000). Flight Simulator 2002. http://www.microsoft.com/games/PC/fs2002.asp.
Miller, G. A. (1956). The magical number seven plus or minus two. Some limits on our capacity for processing information. *Psychological Review, 63*, 81-97.
Norman, D. (1993). *Things That Make Us Smart - Defending human attributes in the age of the machine*. Reading, MA: Addison-Wesley.
Norman, D., & Drapper, S. (1988). *User centered system design*. Hillsdale, NJ: Lawrence Erlbaum.
Norman, D. A., & Shallice, T. (1980). *Attention to action. Willed and automatic control of behavior* (CHIP Report 99). San Diego: University of California.

O'Hara, K. P., & Payne, S. J. (1998). The Effects of Operator Implementation Cost on Planfulness of Problem Solving and Learning. *Cognitive Psychology, 35*, 34-70.
O'Hara, K. P., & Payne, S. J. (1999). Planning and the user interface: the effects of lockout time and error recovery cost. *International Journal of Human-Computer Studies, 50*, 41-59.
Overbaugh, R. C. (1994). Research-based guidelines for computer-based instruction development. *Journal of Research on Computing in Education, 27*(1), 29-48.
Paivio, A. (1991). Dual coding theory. *Canadian Journal of Psychology, 45*, 255-287.
Piaget, J. (1952). *The origins of intelligence in children*. New York: International Universities Press.
Qin, Y., & Simon, H. A. (1990). Laboratory replication of scientific discovery processes. *Cognitive Science, 14*, 281-312.
Randel, J. M., Morris, B. A., Wetzel, C. D., & Whitehill, B. V. (1992). The effectiveness of games for educational purposes: A review of recent research. *Simulation and Gaming, 23*, 261-276.
Reber, A. S. (1992). An Evolutionary Context for the Cognitive Unconscious. *Philosophical Psychology, 5*(1), 33-52.
Reber, P. J., & Squire, L. R. (1998). Encapsulation of Implicit and Explicit Memory in Sequence Learning. *Journal of Cognitive Neuroscience, 10*(2), 248-264.
Reigeluth, C. M., & Schwartz, E. An instructional theory for the design of computer-based simulations. *Journal of Computer-Based Instruction, 16*(1), 1-10.
Reimann, P. (1989). *Modelling scientific discovery learning processes with adaptive production systems.* Paper presented at the 4th International Conference on AI and Education, Amsterdam.
Rieber, L., Tzeng, S.-C., Tribble, K., & Chu, G. (1996, April). *Feedback and elaboration within a computerbased simulation: A dual coding perspective.* Paper presented at the Annual Meeting of the American Educational Research Association, New York, NY.
Schär, S. G. (1996). The influence of the user interface on solving well- and ill-defined problems. *International Journal of Human-Computer Studies, 44*, 1-18.
Schär, S. G., Schluep, S., Schierz, C., & Krueger, H. (2000). Interaction for Computer-Aided Learning. *Interactive Multimedia Electronic Journal of Computer Enhanced Learning, 2*(1).
Schauble, L., Glaser, R., Duschl, R. A., Schulze, S., & John, J. (1995). Students' understanding of the objectives and procedures of experimentation in the science classroom. *The Journal of the Learning Sciences, 4*, 131-166.
Schwier, R. A., & Misanchuk, E. (1993). *Interactive Multimedia Instruction*. Englewood Cliffs, NJ: Educational Technology Publications.

Shneiderman, B. (1998). *Designing the user interface : strategies for effective human-computer-interaction* (3rd ed.). Reading, Mass: Addison Wesley Longman.

Sims, R. (1997, January 27). *Interactivity: A Forgotten Art?* [On-line]. Retrieved February, 2002, from the World Wide Web: http://intro.base.org/docs/interact/

Starr, P. (1994). Seductions of Sim. *The American Prospect, 5*(17).

Svendsen, G. B. (1991). The influence of interface style on problem solving. *International Journal of Man-Machine Studies, 35*, 379-397.

Trudel, C., & Payne, S. J. (1995). Reflection and goal management in exploratory learning. *International Journal of Human-Computer Studies, 42*, 307-339.

Turkle, S. (1997). Seeing Through Computers Education in a Culture of Simulation. *The American Prospect, 8*(31).

Webster's. (2001). *Merriam-Webster's Collegiate Dictionary* (Tenth ed.). Springfield, MA.

White, B. Y. (1993). ThinkerTools: causal models, conceptual change, and science education. *Cognition and Instruction, 10*, 1-100.