Authoring Tools for Educational Simulations

Directed Reading Report
August 8, 2005

Submitted by:

Denise Withers
301015312
dwithers@sfu.ca

MSc. Candidate
School of Interactive Arts & Technology
Simon Fraser University
# TABLE OF CONTENTS

1 INTRODUCTION .................................................................................................................. 5

1.1 DEFINING AND DIFFERENTIATING EDUCATIONAL SIMULATIONS ........................................ 6

1.1.1 Simulations ................................................................................................................. 6

1.1.2 Models ....................................................................................................................... 7

1.1.3 Games ....................................................................................................................... 7

1.1.4 Microworlds ............................................................................................................ 9

1.2 SIMULATION TYPES & ELEMENTS .................................................................................. 9

1.3 TOOLS .......................................................................................................................... 11

2 LEARNING WITH SIMULATIONS ..................................................................................... 12

2.1 EXPERIENTIAL LEARNING .......................................................................................... 13

2.2 CONCEPTUAL MODELS .............................................................................................. 15

2.3 MOTIVATION ............................................................................................................... 16

2.4 MODEL-BUILDING FOR LEARNING ............................................................................ 17

2.5 LEARNING WITH SIMULATIONS ............................................................................... 17

3 CHALLENGES TO CREATING EDUCATIONAL SIMULATIONS ......................................... 18

3.1 ECONOMIC .................................................................................................................. 19

3.2 IDEOLOGICAL ............................................................................................................. 20

3.3 TECHNICAL ............................................................................................................... 21

3.4 PEDAGOGICAL .......................................................................................................... 22

3.4.1 Conceptual model .................................................................................................... 24

3.4.2 Instructional strategy & type of learning task ......................................................... 25

3.4.3 Support for learning .............................................................................................. 27

3.4.4 Degree of control & interaction design ................................................................. 28

3.4.5 Motivation ............................................................................................................. 29

3.4.6 Feedback ............................................................................................................... 30

3.4.7 Reflection .............................................................................................................. 30

3.4.8 Performance & Evaluation .................................................................................... 31

4 STATE OF THE ART ........................................................................................................ 31

4.1 CUSTOM SIMULATION DEVELOPERS ........................................................................ 32

4.2 AUTHORING TOOLS .................................................................................................... 33

4.3 TOOL REVIEW ............................................................................................................. 34

4.3.1 Experience Builder (Experience Builders) ............................................................... 34

4.3.2 Force Ten (Eedo Knowledgeware) .......................................................................... 37

4.3.3 RealCall (SIVOX) .................................................................................................. 39

4.3.4 Unreal Tournament (Epic Games) ....................................................................... 41

4.3.5 Feature review ...................................................................................................... 43

5 FUTURE RESEARCH ........................................................................................................ 51

REFERENCES ....................................................................................................................... 53
LIST OF FIGURES

Figure 1. Experience Builders simulation authoring environment (ExperienceBuilders, 2005). 36
Figure 2. Experience Builders sample simulation report (ExperienceBuilders, 2005). 37
Figure 3. ForceTen simulation authoring environment (Eedo, 2005). 38
Figure 4. RealCall simulation authoring environment (SIVOX, 2005). 40
Figure 5. RealCall simulation editing environment (SIVOX, 2005). 40
Figure 6. ICT Leaders Unreal Tournament simulation/mod (ICT, 2005). 43

LIST OF TABLES

Table 1. Pedagogical features. 45
Table 2. Administrative features. 48
ABSTRACT

As part of their on-going search for affordable and effective tools to support learning, all types of organizations are exploring the potential of computer-based simulations to deliver engaging and effective education and training. In theory, such simulations can provide powerful experiential learning. However, custom simulations can be expensive and time-consuming to produce. As such, they remain out of reach for most educators. To make this technology more accessible, researchers and commercial vendors have developed easy-to-use authoring tools that support the creation of educational simulations. With a view to understanding what characteristics a good authoring tool should have, this paper explores the nature of educational simulations; their connections to contemporary learning theories; and some of the challenges inherent in their design. It then reviews four currently available tools, according to criteria developed from the discussion of challenges. Finally it offers suggestions for future work, with a particular emphasis on investigating how better to support effective pedagogical design in the creation of such authoring tools.
1 INTRODUCTION

As part of their ongoing search for affordable and effective tools to support learning, all types of organizations are exploring the potential of computer-based simulations to deliver engaging and effective education and training. In theory, such simulations can provide powerful experiential learning. Their non-linear, interactive design correlates directly to instructional strategies described in many popular contemporary learning theories. On paper, they have the potential to be powerful learning tools.

In practice, however, the use of simulations for learning today is quite limited. Few educators and trainers have the technical, pedagogical and financial resources needed to create effective and engaging simulations. To build potent, truly interactive programs that offer dynamic visualizations, meaningful feedback, engaging content, instructional support for learning objectives, effective assessment tools and immersive experiential learning, educators must outsource the development to specialty companies. This process is generally costly and time-consuming. The resulting simulations often require expensive hardware or network access to use; and cannot be modified or up-dated for re-use.

In response to these limitations, researchers in academia and the commercial marketplace are creating authoring tools to enable anyone - even an educator without a knowledge of programming - to develop his own educational simulation.

Although these tools overcome many existing barriers to creating engaging and effective simulations, the industry is still in its infancy. Most such programs lack proven instructional design models and strategies. It is still expensive and time-consuming to produce rich media elements for inclusion in the simulations. Limits to artificial intelligence and natural language processing abilities severely curtail interaction capabilities. As well, many educators and those holding their purse-strings suffer from a general lack of understanding about how such simulations can best support different kinds of learning tasks.
This paper explores the current state of simulation development for learning. I attempt to differentiate between types of programs that are commonly lumped into the “simulation” category. This is followed by a review of how simulations can support learning, according to various learning theories. Based on the benefits and liabilities of using simulations identified, I outline many of the current challenges developers face in creating such programs. Using these challenges as evaluation criteria, I then review and compare four different commercially-available simulation authoring tools for soft skills training. The paper ends with a discussion of possible future work in this field.

By exploring how currently available simulation authoring tools support educators, I hope to begin to develop an understanding of what is working, what areas need more research and development and how we might improve our use of such technologies in the future.

1.1 Defining and differentiating educational simulations

As one begins to review the literature and market reports on educational simulations and authoring tools, it quickly becomes apparent that one man's simulation is another's game is another's microworld. It seems that each of these "genres" of computer-based learning media offers strengths and characteristics that support different learning styles and objectives. What follows is an attempt to identify these elements, with a view to exploring how they support learning in the next section.

1.1.1 Simulations

The one element that seems to differentiate simulations from other similar programs is that a simulation models or represents a real or imaginary system in action (Banks, 1999; Rieber, 1996; Williams, 2003). Such systems may be technical, physical, social, theoretical or human. The purpose of the simulation is to enable users to explore interactions between the elements, observe system operation over time and ask what if questions about the effects of changes to any of the system elements or attributes (Banks, 1998, 1999; Sauve, Renaud, & Kaufman, 2005).
Simulations are dynamic and interactive (Williams, 2003). This means that the learner can change attributes of the system and the changes are reflected in the behaviour and output of the simulation (Sauve et al., 2005; Thomas & Schnurr, 1998). It also means that there is a passage of time; the model is not static as in most math and statistical simulations (Banks, 1998).

An object-oriented simulation (OOS) models the behaviour of interacting objects over time (Banks, 1998). The idea of OOS has great intuitive appeal because it is easy for us to view the real world as being composed of objects or things. Most of the authoring tools currently available fall into this category. In an OOS, objects are collected into a library or database; then are accessed and assembled by the simulation as needed. This modular approach supports reusability of objects and learning elements. Object collections are called classes; they describe the characteristics and functionality of common objects included in the simulation (Banks, 1998).

1.1.2 Models

A model is also a representation of a system. However, models are generally static. They may vary in complexity from simple – just representing key parts of the system – to complex, including great detail on the relationships between system elements (Banks, 1998, 1999).

Choosing how to represent a system in a model or a simulation can be deceptively difficult. For abstract systems, a visual metaphor is often used. Such representations are subject to the learner’s interpretation and may be affected by prior learning and experience, cultural influences or other perceptual factors.

1.1.3 Games

Recent research into the use of computer games for learning has revealed significant benefits and liabilities associated with this genre. There is great interest among educators and instructional designers in discovering how to adapt the motivational power of games to learning. However, the way to do so remains unclear.
The terms “game” and “simulation” are often used interchangeably in education. Yet the literature reveals one distinct difference between the two: all games include some level of competition. This may manifest as competition against others, the system or one’s self; but the object of the game is always to win (Rieber & Noah, 1997).

Another difference is that games function according to a set of pre-defined rules (Rieber & Noah, 1997; Sauve et al., 2005). Ben Sawyer, founder of the Serious Games organization, writes extensively on games for learning. He identifies the defining qualities of a game as follows:

- The player must be able to tangibly affect the outcome of the game.
- There must be an overriding goal/challenge as well as sub-goals and challenges to the player with positive and negative outcomes based on their actions.
- It must require mental or physical skill.
- The outcome must be uncertain at the outset.
- It must require the player to develop strategies in order to win or succeed. Those strategies needn’t be apparent at the outset; in fact the discovery element of gaming is one of its most important strengths.
- It must offer multiple paths to success. Linear games tend to take the form of puzzles, which, while useful and entertaining, are primarily about figuring out a specific question and not necessarily about formulating strategies.
- Players must be able to ultimately overcome most obstacles in the game. Only under certain circumstances does it make sense to provide a game that is not at some point "winnable."
- It must be interesting and fun (relevant to its audience) to inspire repeated play (Sawyer, 2005).
1.1.4 Microworlds

Microworlds are not easy to differentiate from simulations. Much of what we know or think about microworlds links back to work by Seymour Papert (1980) who conceptualized microworlds as explorative learning environments that facilitate discovery. Rieber (1996) goes on to qualify this definition, stating that microworlds present the simplest representation of the system being simulated to the learner. He also asserts that microworlds need no instructions to operate; the way to proceed and interact with the program is intuitive. Microworlds may be re-shaped as the learner progresses to offer increasingly complex and sophisticated representations of the system.

Although many describe Virtual Reality simulations as microworlds, such programs may or may not fit this definition. Much would depend on their immediate usability and representational flexibility.

1.2 Simulation types & elements

During my research for this report, I drew extensively on a recent publication by Bryan Chapman and the staff of Brandon-Hall (2005), which compiled information on key vendors of custom simulations and authoring tools for learning. Chapman separates the simulation market into three categories according to the type of learning tasks addressed: software skills, hard skills and soft skills. To provide clarity and continuity, I use the same three classifications in this paper.

Software simulations teach the learner how to use a specific program or application. They are often created by recording the desired interactions with some kind of screen capture program and supplementing this with instruction and tutoring (Chapman, 2005).

Hard skills simulations address training in areas such as troubleshooting or diagnostics; procedural walkthroughs; physical systems operations; emergency response actions; concept explorations (such as understanding a schematic); and virtual world or spatial explorations (as in virtual reality interactions and flight simulators) (Chapman, 2005).
Soft skill simulations involve learning about human systems. Examples include role play, business skills and analysis development, story-based or scenario-type training, problem-solving and sales process simulators (Chapman, 2005). Since human systems are difficult to describe, model and re-create, they can also be notoriously difficult to simulate. Many other systems, such as those in the software and hard skills categories, can be represented by numbers and equations, which translate nicely into computer-based models. Human systems can be much messier. To create valid models of such dynamic systems, developers require powerful resources in key areas such as artificial intelligence, graphics, sound and connectivity.

Simulations for learning in any of these domains generally have three major design components or layers. The first is the underlying or conceptual model that represents the system being simulated. The second layer is the scenario or context in which the model operates. The third layer, unique to this genre of simulations, is the instructional overlay or strategy. This embodies the developer’s learning methodology and supports the delivery of information, features, tutoring, help, coaching, etc. to facilitate achievement of the learning objectives (Rieber, 2002).

Agents are becoming increasingly common elements in many simulation types, especially those addressing soft skill development. An agent is an object that represents a entity (which may or may not be human) that has the ability to determine its own behaviour. These objects are often key elements of the simulation interface, speaking to the learner and facilitating interaction with the program. One important feature of autonomous agents is that their behaviour is not strictly structured according to a pre-determined architecture or outcome; they are governed by artificial intelligence (Summers, 2004).
Agents that use Natural Language Processing (NLP) can be particularly powerful tools for facilitating learning. NLP is a specific type of artificial intelligence that enables the simulation to recognize and understand the voice or text input of the learner and respond to it appropriately (Summers, 2004). This demands significant computing power and can be costly to implement. In addition, technology to support NLP is still relatively young and limited in capability. As such, most available custom simulations and authoring tools default to multiple choice or branching style interactions that limit learner input and interaction, instead of using NLP.

1.3 Tools

As discussed, creating engaging, effective, dynamic simulations for learning can be expensive and time-consuming. Commercially available authoring tools offer an option to developers and instructional designers, by providing template or wizard-style programs to make the process easier and more affordable. Many such tools also conform to the latest standards for LMS’s, learning object repositories, human resource and other systems; this supports the current trend towards interoperability and reusability (T. Murray, 2003b; Summers, 2004).

While these tools do put authoring power and control into the hands of practicing educators, most are limited in their ability to build and support complex simulations and the type of sophisticated graphical and special effects much of the audience expects. By contrast, many are limited only by the instructional expertise and creativity of the educators using them.

The first major players in the authoring market emerged in the 1990’s and focused on creating standalone products that could be delivered via media such as CD-ROM (Paris, 2003). With the increasing availability of the Internet and corporate Intranets, most tools now work in some kind of HTML platform and deliver the final product in a browser-friendly format.
Many developers are also using the “mod” capabilities of popular game engines such as Unreal Tournament by Epic Games to create educational simulations. In Nakamura et al., (Nakamura, Tori, Bernardes, Bianchini, & Jacober, 2003), the authors define a game engine as “a set of integrated, reusable software components designed to facilitate the development of computer game applications. A game engine can optionally have a set of auxiliary software tools for data manipulation and content creation” (p. 3).

Although these engines generally lack specific support for instructional design strategies and methodologies, they do have the benefit of tapping into the motivational appeal inherent in entertainment-based computer game play. Such game engines typically include state-of-the-art capabilities in artificial intelligence, graphics, sound, animation and networked collaboration; as such, they offer tremendous aesthetic power to developers.

Authoring tools to create Intelligent Tutoring Systems (ITS) do not necessarily support the development of full-blown educational simulations and may be part of larger programs. However, they can provide enhanced functionality for such programs by monitoring the learner’s interactions with the system and creating a model of her current knowledge and skills. This model may then be used to help the simulation software determine when and how to offer her learning support through coaching, tips, etc. (T. Murray, 2003b).

2 LEARNING WITH SIMULATIONS

There is a general assumption among educational theorists and instructional designers that simulations can provide powerful learning opportunities. However, to understand how to create effective simulations, developers need to understand what it is specifically about simulations that makes them valuable for learning. In particular, we need to know how educational simulations differ from other simulations, such as those created for scientific evaluation or system planning. In addition, we should have a grasp of what types of learning tasks simulations support best and when they offer an appropriate learning strategy.
2.1 Experiential learning

Simulations deliver a learning environment that is unique in many ways. One of the fundamental benefits they offer that is exclusive to simulations (as opposed to other computer-based learning media) is the ability to learn by doing. This hands-on, experiential type of learning is one of the key features of constructivist learning theory (de Jong et al., 1998).

In constructivism, learners construct knowledge themselves, as opposed to receiving and processing information that is given to them, a feature of other cognitive and information processing learning theories. Constructivism as originally discussed by Swiss psychologist Jean Piaget involves transferring some degree of responsibility for and control over learning from the educator to the learner (de Jong et al., 1998; Fisher, 1991; Hung, 2001). This learning structure encourages active participation by the learner in the simulation – doing versus collecting and processing information. Facilitating such engagement by having the learner perform a task or action directly enhances the learning experience (Greeno, 1996; Kirriemuir & McFarlane, 2004; Bruner, as cited in Mantovani, 2003; Ponder et al., 2003). Knowledge acquired through such interaction is not just a mental state, but rather an experience representing relationships between things and concepts (Dewey, as cited in Hung, 2001).

By their nature, simulations offer some flexibility and variability in the path learners follow through the program. In math or equation-based simulations, such as those using Excel, this path may be completely open-ended. Most soft-skill simulations use some form of branching path that offers learners options or decision points. Giving learners this measure of control over their learning experience is also thought to support more effective learning (Hung, 2001; Kirriemuir & McFarlane, 2004). While some educators believe that the branching format is too constraining, others point out that exposure to different options via multiple choice may introduce the learner to new ideas or responses that she may not have considered before (Summers, 2004).
By contrast, creating a simulation that is completely open and offers no guidance to the learner may be a hindrance to the knowledge building process. Research indicates that some structure and learning support is necessary to facilitate learning; otherwise learners can become lost in the simulation and fail to achieve their objectives (Plowman, Luckin, Laurillard, Stratford, & Taylor, 1999; Rieber & Noah, 1997).

One benefit of using simulations for learning is that they support active reflection on the part of the learner as part of the interaction process. As she works through the simulation, she reflects constantly on what she knows, has learned and experienced; makes inferences from that reflection; and tests her inferences and theories through her actions and decisions (Williams, 2003). Although this type of learning is often categorized as tacit in that it exists below or beyond the learners’ consciousness and occurs without deliberate effort or attention (Rieber & Noah, 1997), such a process makes the learner’s knowledge tangible through her action (Galarneau, 2005).

This relates to learning theories on the development of expertise, in which experts build knowledge through activity, experience and practice, as opposed to memorization and the use of formal, syntactic rule systems (Greeno, 1996).

Learning through simulations also supports the process of learning via situated cognition. According the theorists, knowledge is built through interactions between the learner and her environment (including the people around her). Simulations permit learners to build knowledge through interactions with their virtual environment. Experiencing and exploring the context and relationships between objects, concepts and the world are key elements in facilitating such knowledge construction (Duncan, 1995; Hung, 2001).

For software and hard skills training, simulations can offer unlimited opportunities to practice, which supports learning via the behaviorist approach of drill and repetition.

Simulations also offer the learner the opportunity to test new knowledge, strategies, skills and techniques in a virtual, risk-free environment. Many educators believe that experiencing failure such as this is a critical component in the learning process (Schank, as cited in Galarneau, 2005).
2.2 Conceptual models

Research suggests that the way in which the system being simulated is represented can affect learning significantly (Banks, 1999; Carson, 2003; Dalgarno, Hedberg, & Harper, 2002; Rieber, 2002; Romme, 2004; Winn, Windschitl, Fruland, & Lee, 2002). The conceptual model created to represent the system should portray the relationships between key elements and attributes accurately. The model should also include enough detail about the system for the learner to understand how it functions; but it should not be so complex as to induce cognitive overload.

Among others, two theories which should be considered when developing conceptual models are Dual Coding Theory and Mental Models. The former, best supported by the work of Paivio, suggests that pictures are superior to words for remembering concrete concepts (Paivio, as cited in Rieber, 2002; Ware, 2000). This indicates that developers should maximize their use of effective graphics, video and animation. In addition, providing information and data to learners both verbally and visually should enhance their ability to build knowledge.

Mental model theories connect to the idea of simulations as microworlds, in that people form mental models of the physical world in order to interact with it. The conceptual model presented by a simulation can directly affect the mental model the learner constructs of the system being simulated (Greeno, 1996; Mandler, 1984; Moessinger, 1978; Norman, 1988; Rieber, 1996). Mental models are thought to be dynamic, changing constantly as the learner builds new knowledge.
2.3 Motivation

In their marketing literature, most vendors of custom simulation products and authoring tools assert that using simulations for learning promotes motivation and engagement among learners. Both these sensations are critical to facilitating effective learning (Arnone, 2003; Bangert-Drowns & Pyke, 2001; M. Csikszentmihalyi, 1997; Lepper & Cordova, 1992; Mayer, 2002; Rezabek, 1995; Rotto, 1994). However, the marketing literature does not identify which aspects of simulations trigger this response. Many assume that the “bells and whistles” of animation, stereo sound and 3D graphics automatically generate engagement. There is no evidence to support this; yet there is evidence to the contrary (Milne, 2003; Pausch, Snoddy, Taylor, Watson, & Haseltine, 1996; Winn et al., 2002), that technology alone cannot sustain engagement.

Such assumptions may stem from a belief that the extrinsic, goal-oriented motivation generated in the context of a computer game is also automatically a by-product of an educational simulation (Kirriemuir & McFarlane, 2004). Since this element of competition and goal-achievement is not inherent in simulations, this assumption is false.

Others may be confused by the engagement that can be generated through the use of storytelling or scenarios. While such an approach is a common feature and instructional strategy of many simulations, it is not necessarily present in all programs.

The potential to generate engagement and motivate learners does exist in the construction of educational simulations. However, it is important for developers to understand that simulations do not automatically achieve this; creating motivation and engagement must be explicit design goals of the instructional methodology.

Using strategies that incorporate the development of cognitive conflict in the learner is one way to do this. Facilitating play is another. Piaget, among others, considered play to be an important element in helping learners assimilate new knowledge (Mihalyi Csikszentmihalyi, 2000; Piaget, as cited in Rieber, 1996). However, developers need to be aware that too much play or fun can inhibit reflection and may harm the learning process (Paras & Bizzocchi, 2005; Rieber, 1996).
2.4 Model-building for learning

In addition to the use of pre-constructed simulations for learning, educators may also use the process of simulation construction itself as a learning strategy. Working through the process of designing the model, its underlying systems and representation can provide a powerful educational experience (Rieber, 2002). The availability of authoring tools for simulations makes this approach more feasible today than ever before.

2.5 Learning with simulations

The relationships between various iterations of learning theories and the use of educational simulations is complex and only somewhat understood. As our use of simulations grows, research should provide us with more insight on how, when and why to use simulations.

For now, Hung (2001) offers an effective compromise. Based on his review of available research and literature, he suggests that there is a role for behaviorist, cognitivist, constructivist and social constructivist models of learning in the design of educational simulations.

“Adopting the general premises of both [constructivism and social constructivism] schools of thought, we have: (1) Learning is an active process of constructing rather than acquiring knowledge; (2) Knowledge can be socially constructed where the social interactant may include just oneself; (3) The interpretation of knowledge is dependent on (a) the prior knowledge and beliefs held in one’s own mind and (b) the cultural and social context through which the knowledge was constructed” (Hung, 2001, p. 283).

The use of simulations may also address current concerns about the changing ways in which children and young adults learn. Some of the tensions between new and old methods are described as:

- Twitch speed vs. Conventional speed.
- Parallel processing vs. Linear processing.
- Graphics first vs. Text first.
• Random access vs. Step by step.
• Connected vs. Standalone.
• Active vs. Passive.
• Play vs. Work.
• Payoff vs. Patience.
• Fantasy vs. Reality.
• Technology as friend vs. Technology as foe (Kirriemuir & McFarlane, 2004).

Simulations seems to offer great potential to enhance the way we all learn. But building programs that are effective and evergreen on tight timelines and limited budgets presents substantial challenges to developers.

3  CHALLENGES TO CREATING EDUCATIONAL SIMULATIONS

As experiential learning tools, simulations can help users build understanding, knowledge and skills in almost any type of learning task. Though their deployment in academia and organizational learning is growing, they are still not in common usage (Chapman, 2005) and represent only 2% of the e-learning industry (Hoban, as cited in Summers, 2004). In addition, many of those that do exist do not come close to realizing the full potential of the genre as an educational medium.

What follows is a brief review of some of the major challenges, issues and barriers educators and developers face in designing, developing, delivering and evaluating simulations for learning. Finding ways to manage these challenges will help developers create better custom simulations. Understanding these concerns is also a necessary prerequisite to being able to develop effective and affordable authoring tools to enable educators to produce their own programs.
In some respects, creating simulations for learning offers more complex challenges than building simulations for research or analysis. The developer must create not only the conceptual model to represent the system being simulated and the computer-based representation of it; but he must also integrate an instructional model with pedagogical support and administrative features, tools and standards into the package.

For the purposes of this paper, I have separated the challenges developers face in creating effective simulations into four categories: economic, ideological, technical and pedagogical.

### 3.1 Economic

If you ask an educator why he does not include simulations in his learning plan, the answer most often given is that they are too expensive to create, buy and maintain (Chapman, 2005; O'Keeffe, Lunness, Kilpatrick, & Balzac, 2005; Thomas & Schnurr, 1998).

It is difficult to find dollar values for average costs of custom simulation production, partly because every program is different. Some companies suggest that a cost ranging from $50,000 - $150,000 USD is average (Chapman, 2005). Aldrich (2004) uses the number $100 - $1000 USD per user. Most cost metrics are tracked according to development time. According to the Brandon-Hall 2005 report (Chapman, 2005), the ratio for creating an e-learning course that is not a simulation is 220:1 (development time:finished hours). By contrast, the average ratio calculated from focus group research revealed that the time to create a simulation for e-learning is from 750:1 to 1300:1.

This translates into a cost that is out of the reach of most educators. As well, creating a simulation that offers a moderate level of complexity and pedagogical support, and features rich media such as animation, video and audio can take months or even years to complete, eliminating any possibility of quick or timely delivery.
Once a custom simulation is finished, the content is generally locked; any attempts to make changes or updates may result in having to re-program considerable parts of the software.

Other economic factors that inhibit the use of simulations include the cost of acquiring and maintaining equipment for learners to use the program. This is particularly true for simulations that require the use of special equipment, such as HMD’s or CAVE’s to support Virtual Reality.

Many educators are also interested in exploring the potential of collaborative and multi-user learning with simulations online or over a network, such as the type of interaction found in massive multiplayer online games. However, the expense of network access is generally too great to permit such applications (Sawyer, 2005).

### 3.2 Ideological

For those fortunate few who can afford to buy or develop educational simulations, there is still an ideological barrier to overcome in convincing clients and administrators that simulations can provide effective learning and are not simply an excuse to play. Summers cites such cultural resistance as the second greatest obstacle to adopting simulations for e-learning.

Part of this bias stems from difficulties in measuring the learning that happens during a simulation (Becker, 2005) and relating that to ROI (Return on Investment). Simulations often facilitate tacit knowledge building and the development of expertise, as opposed to explicit or declarative learning. The former is more difficult to measure and learners are often not aware of their new knowledge. In addition, since learners may choose different paths through the simulation or vary its parameters in different ways, not all of them will have the same learning experience, nor will they learn all of the same things (Kirriemuir & McFarlane, 2004).
Older educators and those managing their organizations may also harbour fears about using technology for learning, as they do not fully understand how to tap its potential. Since they often make the decisions about how curriculum will be developed and delivered, this bias may cause them to veer away from tools that employ simulations (Aldrich, 2004). Such uncertainties may also stem from a lack of research validating the applicability of simulations for learning. Much more data is available on the use of games and on how children learn from such media than on using simulations for adult learning.

Finally, the reality of the e-learning boom is that it has been a bust for many educators; this is reflected in their distrust of new trends and technologies (Aldrich, 2004; Chapman, 2005; Summers, 2004).

### 3.3 Technical

Despite the rapid advances in technology and our capabilities, the technical challenges inherent in creating an educational simulation are tremendous. This is the area where growth is most rapid and most research seems focused.

Variations in media formats, systems standards, platforms, networks, programming languages and delivery formats are just a few of the issues that arise. In an effort to manage these, there is a current trend towards creating simulations that are modular or object-oriented and support interoperability between Learning Management Systems (LMS), platforms and delivery formats.

Simulations created for learning can include tremendous amounts of data and huge media files that need to be processed or compiled at high speeds. Storing, processing, transmitting and managing the enormous files that result presents overwhelming difficulties for many educators. This process may be complicated by requirements to deliver the simulations over networks with limited bandwidth or on systems with weak hardware and software capabilities.
Ensuring usability is a particularly important challenge, as simulations that present a learning curve that is too steep or difficult to navigate can create cognitive overload and harm the user’s ability to build skill and knowledge (Caird, 1996; Thomas & Schnurr, 1998).

It is important to note that these technical issues that are related to pedagogical and administrative design are all supplementary to those inherent in non-educational simulation creation. Developers who build simulations for educational use share additional technical challenges with those building programs for research and analysis purposes. Some of these include: designing effective visualization techniques; managing data import and export standards; creating debug tools; conceiving and implementing the interface; ensuring compatibility between programming languages; facilitating and translating user input and interaction; and providing appropriate tools for dynamic analysis (Banks, 1998).

3.4 Pedagogical

Creating an educational simulation that is pedagogically effective is perhaps the most important challenge developers face, since the instructional design directly affects the learners’ experience (Galarneau, 2005). Yet this is also arguably the most difficult part of the process, since it is the least well understood.

Many models exist for the design and development of simulations for research, analysis or experimentation. But few authors offer methodologies specific to the creation of effective and engaging interactive simulations for learning. Most current design strategies rely on what we have learned about interactive media design, which may or may not share the dynamic and experiential elements of a simulation.
Barker (1994) offers a summary of the outcomes that should be the goals of an interactive learning environment that seems appropriate to educational simulation design: knowledge acquisition, skill development, skill rehearsal, problem solving and self-realization. Some things he suggests designers should consider to achieve these outcomes include: the learning theory mix; instructional position mix; machine character mix; environmental factors; mode of use; locus of control; extent of instructor intervention; aesthetic features; content; and role of technology.

Norman (1993), known for his innovative approach to effective and usable design, offers seven basic requirements of a learning environment that seem appropriate to simulation design as well. These share many elements of Flow Theory, as described by Csikszentmihalyi (1997).

- Provide a high intensity of interaction and feedback.
- Have specific goals and established procedures.
- Motivate.
- Provide a continual feeling of challenge that is neither so difficult as to create a sense of hopelessness and frustration, nor so easy as to produce boredom.
- Provide a sense of direct engagement, producing the feeling of directly experiencing the environment, directly working on the task.
- Provide appropriate tools that fit the user and task so well that they aid and do not distract.
- Avoid distractions and disruptions that intervene and destroy the subjective experience. (Norman, 1993)

For the purposes of this paper, I have broken down the issues affecting the pedagogical design of educational simulations as follows: conceptual model; instructional strategy and type of learning task; support for learning; degree of control and interaction design; motivation; feedback; reflection; and performance and evaluation. These same general groupings are used as criteria to evaluate the authoring tools reviewed in Section 4.
3.4.1 Conceptual model

The design of the conceptual model is a crucial part the simulation development process. Good models facilitate learning by helping the user build knowledge about and form an accurate mental model of the system being simulated.

Ideal representations should include important, critical features of the simulation and exclude those that are irrelevant. They should be well-suited to the learner, with respect to prior knowledge and ability level. And they should match they type of learning task (Norman, 1993).

Choosing an appropriate representation for physical or concrete systems may be as simple as reproducing the system visually.

Creating a conceptual model for soft skills and social simulations, as well as those representing abstract systems can be considerably more difficult. This often involves the use of metaphor, in an attempt to give the learner something familiar to anchor her learning. This can be dangerous however, as ill-chosen metaphors can lead to misconceptions (Winn et al., 2002). These types of simulations may require more learning support and scaffolding to help users develop valid mental models and knowledge. The use of agents and avatars here, as well as scenarios and storytelling, can provide tools and structure to strengthen the model (Schank, 1997; Summers, 2004; Towle, 2000).

As part of this design process, developers must decide how much detail and complexity to include in the model, or what level of fidelity it will have. Fidelity describes the degree to which the simulated system mirrors its real-world counterpart (Choi, 1997). Many novice designers automatically assume that a high degree of realism will correlate to a highly effective learning experience. Yet there is little evidence to support this notion. There is on-going debate in the literature about whether complex models overwhelm novice learners and whether over-simplified models dumb-down the learning process. In both cases, a negative impact on transfer of learning to the real system may occur (Alessi, 1988).
The critical issue for designers should not be how real the simulation appears, but that it represent what is and not what should be. This is often a challenge for developers who are under pressure to create simulations that are politically correct, idealistic or based on theory instead of reality. When the learner tries to map the knowledge and experience she acquires in the simulation to the actual system, the discrepancies between the two can thwart the learning transfer (Aldrich, 2004; Winn et al., 2002).

Fidelity is also used to describe the visual treatment and presentation of the simulation; how much does it look like the real system? Choices facing designers here are many: 2D or 3D; static or animated; video or CGI; image or text or audio or all three; immersive VR, desktop or PDA? Many aspects of visualization theory can be applied here.

Although the current state of technology and expertise in the industry makes the production of visual elements such as graphics, animation and video relatively accessible, few simulation developers seem to be making effective use of visualizations. Many simulations produced with authoring tools and in custom production environments still rely heavily on the use of text and numbers. This has implications for the learning effectiveness of the simulations, especially with respect to younger learners, who show strong preferences for visual learning (Kirriemuir & McFarlane, 2004; Rieber, 2002; Williams, 2003).

### 3.4.2 Instructional strategy & type of learning task

Identifying an appropriate instructional strategy to achieve the specified learning objectives is a key aspect of the design of any learning media, not just simulations. Technology is just a tool to support learning; the real power of the experience comes from the instructional strategy and skillful application of learning theory (Osberg, as cited in Dalgarno et al., 2002; Mantovani, 2003).
For example, despite current trends to employ 3D representations in educational media, it appears that 3D may not add enough value to the learning process to justify the increased expense. In their work with a simulation to teach concepts of fluid dynamics, Winn and his colleagues (2002) found that the use of an immersive, 3D environment helped learners understand spatial issues better, but did not facilitate their learning on other, non-spatially related concepts.

Hard skill and software simulations that require the learner to master a particular, demonstrable skill might use strategies that include demonstration, practice, model progression and performance assessment.

Soft skill development, which may involve helping the learner build knowledge about a specific concept, may benefit from an approach that employs discovery, experimentation, scenarios or reflection.

Other popular instructional models employ Intelligent Tutoring Systems (ITS), which use artificial intelligence to monitor a learner’s progress, create models of her behaviour and provide appropriate support at appropriate times (Ainsworth, 2003).

Goal-based scenarios and problem-solving, such as being placed in charge of a new company or having to find the most effective way to get to the moon can also provide effective approaches for soft skill development (Schank, 1997).

Deciding on an architecture or structure for the simulation is another important part of the instructional design process. Traditionally, simulations for research or analysis support an “open” structure; that is, the data is entered and various mathematical equations run when the user changes a parameter. The results of those equations produce new states which are unknown and therefore cannot be programmed in advance. This approach is possible for some types of simulations, such as those based on Excel. Many educators claim that this type of structure maximizes opportunities for learning creativity and experimentation (Summers, 2004).
However, most soft skill simulations produced today tend to adopt some sort of closed or branching approach. This offers the instructional designer more control over the outcomes learners achieve. But such simulations can be more work to create, since the path to the outcomes and media or representations of the results must all be scripted in advance (Summers, 2004). It also limits the creativity of the learner, in that she will only have a prescribed number of options to choose from. There is no room in a branching structure for original or unique user input.

3.4.3 Support for learning

One of the findings that emerged from research into the past decade of e-learning is that learners cannot be completely self-directed in their pursuit of their learning objectives; they need support or scaffolding (Grow, 1992, as cited in Merriam & Caffarella, 1999; Plowman et al., 1999). Choosing how much support to provide, when and how to provide it are decisions that test the skill of every instructional designer (Aldrich, 2004). There are no proven rules or methodologies to support educators in this area, save the knowledge that such support must be provided.

Research shows that adults in particular are uncomfortable with discovery-based learning environments, which can cause anxiety. They desire structure and guidance (de Jong et al., 1998; Rieber, 2002). It is best if such support comes when the learner is engaged in the simulation process: not before starting or after completing the program.

Too much support can quash the meaningful learning that occurs when users discover and build knowledge in their own way and at their own pace; too little can leave the learners frustrated and cause them to quit (Aldrich, 2004; Plowman et al., 1999). Some level of frustration can be productive and many consider it to be a desirable part of the learning process (Schank, 1997). But too much “mal-learning” can cause conceptual confusion and may do more damage than good (Barker, 1994).
There is a wide range of tools and techniques educators can use to deliver coaching and support. This list grows daily, with new technologies being developed by both industry and academia. Examples include in-line artificially intelligent agents and avatars, chat rooms, links to external resources, expert databases, tutorials, case studies, exercises, natural language processing, voice recognition, virtual reality and collaboration, live networked coaches and help functions (Summers, 2004).

3.4.4 Degree of control & interaction design

As described earlier, in reference to the relationship between constructivism and educational simulations, the amount of control given to learners over the learning activity in simulations is often touted as one of the genre’s biggest strengths (Aldrich, 2004).

For designers of such simulations, choosing how much control to give away evokes many of the same challenges discussed in the previous section on learning support. If the learners have too much control, they may get lost; too little and they may end up simply clicking through a linear presentation (Aldrich, 2004).

The level of interactivity offered to users relates directly to the architectural structure of the simulation. Popular branching-style simulations reduce interactivity to clicking the mouse to make a choice at decision points. By contrast, open, math-based simulations offer the learner more flexibility and freedom in the type and amount of input she can enter. However, all simulations must necessarily limit the number of parameters that may be manipulated to make the program understandable and manageable.

According to Murray (1998), true interaction or agency occurs when the actions of the learner affect or change the outcome of the situation. This implies a dynamic relationship; the learner takes an action and the simulation changes. The learner responds to the change with another action and the simulation state changes again. Such agency is rarely found in educational simulations. Yet it is highly desirable, as it richens the learner’s experience, making it more enjoyable, meaningful and memorable (Sastry and Boyd, as cited in de Jong et al., 1998; Mantovani, 2003).
When developers shift control to learners, there is also an inherent shift of responsibility for the learning process and outcomes (Rieber, 1996). Many learners, adults in particular, are uncomfortable with this shift (de Jong et al., 1998; Summers, 2004). This can result in an increased need for motivation; the more they want to learn, the more likely they will be to take on this new responsibility and the extra effort associated with it.

### 3.4.5 Motivation

Motivation or lack of it, can make or break a learning experience (Galarneau, 2005). Unmotivated learners are incapable of engaging in the process of knowledge construction. Highly motivated learners will invest time and energy beyond the minimum requirements to extend their learning (Gagne & Briggs, 1979; Schiefele, 1991).

Many instructional designers count on the inherent interactivity of educational simulations to generate engagement and motivate learners. But as discussed above, most simulations do not support high levels of interactivity. As such, developers need to include strategies for engaging and motivating learners in their model development (Galarneau, 2005).

Methods for creating motivation encourage active participation by the learner, giving them more control, facilitating exploration and structuring opportunities for reflection. None of these design feature occur by accident; all must be deliberately created.

Many educators are turning to the entertainment industry for help in this area, drawing on the expertise of screenwriters, filmmakers and game developers. “Entertainment people”, as they are sometimes called (Lindheim & Swartout, 2001), approach the task of instructional design differently. Rather than focusing on the learning objectives, they focus on creating engagement, through the use of storytelling and character development. Their rationale is that the emotion and affective learning generated through these techniques can have tremendous impact; if they can generate emotion and engagement, it will be much easier for the learner to achieve the objectives (Iuppa, Weltman, & Gordon, 2004; Lindheim & Swartout, 2001; Ponder et al., 2003).
Other effective frameworks for motivating learners come from Flow theory (M. Csikszentmihalyi, 1997) and the Keller’s ARCS Model of Motivational Design (Paras & Bizzocchi, 2005). Both describe the need for making the learning experience relevant to the learner. Both also address the need to adapt the challenge so that it is neither too easy nor too difficult for the learner to achieve.

### 3.4.6 Feedback

Providing timely, effective feedback to learners on their strategies and actions is an essential part of any learning process (M. Csikszentmihalyi, 1997; Laurillard, 2002). Video games do this quite well; the player always knows his status and the consequences of his actions.

Many of the newer simulation authoring tools support this concept, providing functionality for real-time feedback and coaching. Some have even adopted techniques from the game industry and include status bars and dynamic scorecards that are always visible to the learner.

### 3.4.7 Reflection

Critical, active reflection is an important part of the learning process (Greeno, 1996), one that is often overlooked in the design of educational simulations.

As mentioned earlier, tacit reflection is a natural part of the interaction process in a simulation. Each time the learner receives feedback on her action, she reflects on the feedback and processes it, along with her knowledge, to make new decisions and take new actions.

It is more difficult to facilitate explicit reflective activities. One reason for this is that they interrupt the learner’s focus or flow, by asking her to consciously step back and think about what she is learning (Paras & Bizzocchi, 2005; Rieber, 2002; Rieber & Noah, 1997).
Techniques that facilitate explicit reflection may be included in ancillary exercises. These may include journaling, teaching others what the learner has discovered, chatting or debriefing with peers or a coach, creating quizzes on the simulation, sharing stories about what happened during the simulation or creating concept maps to represent the simulation system (O'Keeffe et al., 2005).

3.4.8 Performance & Evaluation

Devising ways to evaluate learning and performance during a simulation is considered a necessary evil by many developers. Measuring learning in something like a soft skill simulation can be a challenging process; often the learning does not take effect or become fully realized until long after the learning experience has ended (Aldrich, 2004). In addition, there is increasing debate among educators concerning the validity of such measurements (de Jong et al., 1998). Instead, many are attempting to shift the emphasis away from evaluation and outcomes to focus on the process of learning (Mantovani, 2003).

Most simulations offer support for measuring and recording computer metrics, such as time on task, task completion, number of correct responses to quizzes etc. These can often be reported directly into a standardized LMS. More complex programs monitor and model the student’s progress through the simulation and interpret that as a measure of learning success.

Until more effective measurement and evaluation tools and techniques are developed, many learning organizations will continue to shy away from using simulations for education; it is simply too difficult for them to relate the results to ROI.

4 STATE OF THE ART

State-of-the-art is a somewhat nebulous phrase, often misused by marketers to lure clients into buying the latest technologies, regardless of how well the technology suits their needs. I use it here to describe the most advanced or leading edge work in the field of educational simulation production.
Unlike many other fields, in which research advances are achieved primarily in academia, much of today’s state-of-the-art technology comes from industry. This is due largely to the tremendous growth, popularity and commercial success of the computer and video game market. Current game offerings incorporate the most powerful technologies in the fields of artificial intelligence, animation, graphics and collaborative networking.

Many of these technologies are in use in educational simulations today; however, their application is restricted primarily to custom programs created by specialty simulation and e-learning companies. Again, this relates directly to barriers of cost for development, implementation and delivery.

Other notable achievements not related to the game industry include the development of natural language processing and speech recognition technologies. In the next section, I describe some of the more advanced applications of these technologies in educational simulations.

### 4.1 Custom simulation developers

When one surveys the offerings of custom simulation production for soft skills, several organizations emerge as clear leaders. These include: the Institute for Creative Technologies at the University of Southern California, WILL Interactive, Ninth House, Cognitive Arts and Simulearn.

These groups are blending actors and avatars, 3D and virtual reality, theatre and film, game and adventure, fantasy and reality. Their simulations use NLP, life-size 3D autonomous agents, artificial intelligence, collaborative VR, Hollywood storytelling and haptic interfaces. The result is simulations that are highly engaging and effective – and incredibly expensive to produce.
4.2 Authoring tools

The challenge in creating strong authoring tools for educational simulations is to come as close to the achievements of such custom simulations, while providing a tool that is affordable, easy for a non-programmer to use, based on modular or object-oriented principles and interoperable with common computer and administrative learning management systems.

Although such authoring tools are relatively new on the market, researchers started working in this field more than a decade ago. Strangely, I could find little academic research on new simulation authoring tools dated after 1998.

One of the most often cited is SIMQUEST, which bears many similarities to those commercially available today. This was one of the first working tools that specifically provided cognitive support for learning in the discovery process (de Jong et al., 1998). A SIMQUEST simulation includes a simulation model, user interfaces, a collection of instructional supports and specifications of the control flow of the learning environment (Kuyper et al.). It functions much like a wizard, helping the educator select learning objects from a library for inclusion in the simulation and using these objects to create structure around the desired flow.

Other research-based authoring tools include: RIDES and VIVIDS, which provide some low level automatic support for detecting student operations (Munro, 2003); Eon, which offers a suite of domain independent tools for authoring all aspects of a knowledge based tutor (T. Murray, 2003a); and the MultiVerse User Environment (MUE), which supports the inclusion of multimedia and in which the simulation interface is developed separately from the computational model driving the simulation (Thomas & Schnurr, 1998).
Today, many authoring tools are commercially available for producing e-learning programs and multimedia. But only 40 or so specifically support educational simulation production. Most focus on generating programs for software training; about a third support soft skills simulations and less than a quarter are for hard skills simulations (Chapman, 2005). Almost all generate programs most suitable for desktop delivery, either via standalone media or a network.

4.3 Tool review

Since the demand for soft skills simulations is growing (Chapman, 2005) and this genre of educational simulation seems to offer some of the most daunting challenges for developers, I have chosen to focus on authoring tools for this type of simulation for my review.

Three of the four tools I chose were represented in the Brandon-Hall report. These are: Experience Builder from Experience Builders; ForceTen from Eedo Knowledgeware; and RealCall from SIVOX.

These seem to offer some of the most innovative overall support to non-technical educators, while attempting to incorporate some of the latest technological advances. The fourth is actually the game engine from Unreal Tournament, produced by Epic Games. I chose this because of the current trend by educators to attempt to tap into the huge popularity of video games in the creation of their simulations.

In Tables 1 and 2 on pages 45 and 48 respectively, as well as the section that follows, I evaluated each of the tools according to the criteria and categories developed in Section 3.

4.3.1 Experience Builder (Experience Builders)

The Experience Builder is a web-based, collaborative authoring tool that is designed to facilitate the creation of online role-playing simulations (ExperienceBuilders, 2005). It differs from many other tools offering similar functionality in a number of key ways.
The authoring process is structured according to the company’s own Learning Design, which comprises four elements: identifying and recreating common situations found on the job; building a behavioural model that prescribes specific best practices for employees to learn via the goal-driven scenario; designing enough “simulation space” for the learner to try different strategies during the simulation; and developing context sensitive tutoring to assist her during the program. The behavioural model developed during this process functions as the design anchor. This model-centered approach differs from that found in most other tools, in which developers build the simulations page by page from start to finish.

To purchase the tool, clients buy an annual or per-project license that gives them access to the vendor’s server, which is where the tool resides. All authoring is done on the server. Although the client can download and install the simulation at any time and can edit it thereafter with any HTML tool, he never owns the tool.

The demos available on the company’s website use video as an integral part of the simulation; although most others claim that their tools also support the use of such rich media assets, few demonstrate this capability.

Experience Builders is also one of the few companies to include audit and debug tools as part of its authoring package.

Finally, the company claims that the tool supports simultaneous authoring on the same simulation, which could help development teams shorten their development time frame.
Figure 1. Experience Builders simulation authoring environment (ExperienceBuilders, 2005).
Figure 2. Experience Builders sample simulation report (ExperienceBuilders, 2005).

4.3.2 Force Ten (Eedo Knowledgeware)

Although Brandon-Hall includes ForceTen in their simulation tool review, it is not designed to support simulations specifically, but can be used to author many different types of e-learning programs. ForceTen is actually a suite of tools that includes a Learning Content Management System (LCMS).

Some of the unique features provided by this package are a Workflow Management tool, Storyboard facility and WebClass real-time conference tool for virtual collaborative learning; plus a searchable knowledgebase and knowledge sharing tool to support learners during and after the simulation.
The tool suite focuses on meeting technical challenges, such as compliance with standards, supporting an object-oriented architecture and offering extensive reporting capabilities. There appears to be little or no guidance built in to help educators manage the pedagogical challenges of simulation design.

Like most other vendors, Eedo provides little public information on pricing. The suite is sold according to the server size and configuration desired as follows:

- $40,000 USD for single server CPU
- $75,000 USD for dual server CPUs
- $130,000 USD for four server CPUs (Eedo, 2005).

*Figure 3. ForceTen simulation authoring environment (Eedo, 2005).*
4.3.3 RealCall (SIVOX)

RealCall is an authoring tool for Call Centre training. It is one of the few tools that uses natural language processing, with speech recognition and text-to-speech capabilities. The system listens to learners during a simulated call and analyzes everything from their voice tone, to the words they use, to the speed of their delivery. During this analysis, if the system finds a mistake or determines that the learner needs to be corrected, it interrupts the call in the guise of a Real-Time Coach to offer advice and corrections. The system is dynamic in that both the system and the learner can stop and start the call again at any time (SIVOX, 2005).

Agents who require deeper learning are sent to the Knowledge Center where they can acquire a better understanding of what to say and how to say it by reviewing best practices, taking quizzes and using study guides.

The tool also includes the capability to create software simulations, so that as the learner practices her voice interaction, she also learns to use the Call Centre’s live data systems.

The RealCall pricing model includes a server license for the authoring tool; user licenses; and professional services to create the initial set of simulations and conduct a knowledge transfer. This last item includes training and certification in creating effective simulations. Typical installed pricing, including development of the initial simulation set, ranges from $100,000 to $750,000 USD (SIVOX, 2005).
Figure 4. RealCall simulation authoring environment (SIVOX, 2005).

Figure 5. RealCall simulation editing environment (SIVOX, 2005).
4.3.4 Unreal Tournament (Epic Games)

The notion of using game engines to create simulations for learning is not new; educators have been working to tap this technology for almost a decade. Technically, I would not classify such engines as authoring tools, simply because they are too difficult for lay teachers, trainers and facilitators to use. They also do not provide any specific pedagogical design support or compatibility with learning systems. However, they do offer a way for educators with access to a skilled programming team to exploit some of the most powerful technologies in artificial intelligence, 3D graphics and modeling, autonomous agents, audio and networking available today.

Unreal Tournament (EPICGAMES, 2005), marketed as a first-person shooter game, offers one of the most developed, flexible, and usable engines for modding (Lewis & Jacobson, 2002; Nakamura et al., 2003). Its open source game code is written in UnrealScript, a bytecode-compiled scripting language similar to Java. However, the rendering engine, referred to as the “crown jewel of the game” (Lewis & Jacobson, 2002), is proprietary and not open to modification.

Nakamura et al. (2003) used this engine to create an educational simulation game at the University of São Paulo, in which the learner played the role of a museum worker assigned to set up paintings for a new exhibit. It has also been incorporated into the extensive research at the Institute for Creative Technologies at the University of Southern California, which is partnered with the United States military to develop new simulation training technologies.
Nakamura et al. (2003) followed a development process that had three phases: map construction, model construction and code implementation. They found that the map editor worked well for their purposes. However, the engine does not provide modeling tools for model creation: just a conversion tool to import models created with other programs. They had difficulty finding a compatible tool, which resulted in problems creating and translating models for their new game. In the third stage, working with UnrealScript, they found that a combination of design failures and poor implementation of both the language itself and the Unreal Tournament class hierarchy, led to considerable difficulty in the creation of well-designed object-oriented code.

Based on their experience, they drafted a number of recommendations for future use of this engine, which include the addition of modeling and map creation tools, an object-oriented resource library, stricter implementation of the code and better error detection and debug capabilities (Nakamura et al., 2003).
4.3.5 Feature review

Criteria definitions:

Architecture – What types of simulation structures are available? Does it support learner input? Are the outcomes open, as in a math-based simulation in Excel; or are they closed, as in a branching, multiple choice format?

Interaction Design – How interactive can the simulations be? Do they support learner input; or is interaction limited to clicking through the simulation?

Instructional Strategy - Does the learning strategy match the type of skill being taught? Is a learning methodology built into or supported in the design process? Upon what premises or theories is the learning methodology based?
Learning Support – What types of coaching, scaffolding, help, instruction or facilitation are supported by the tool? How are they incorporated into the simulation creation process?

Interface Design – How much control does the developer have over the interface design? Is it all limited to pre-existing templates; or can the designer create his own?

Motivation – What support is included in the tool to facilitate the development of a motivation or engagement strategy in the simulation creation process?

Rich Media - What kinds of media can the tool support besides text?

Usability – What testing or validity functions does the tool offer to ensure usability?

Feedback - How does the software facilitate the inclusion of tools and opportunities for feedback to the learner during the simulation?

Reflection - How does the software facilitate the inclusion of tools and opportunities for reflection by the learner during the simulation?

Programming complexity – How easy to use is the tool for developers who lack programming and technical skills?

Accessibility – What kinds of special hardware or software does the learner need to use the finished simulation?

Modularity – How does the tool support the reuse and sharing of learning objects and other elements (including pre-existing media)?

Interoperability – With what standards is the tool compliant?

Delivery platform – What platforms are supported?

Evaluation – What performance and evaluation features and reporting functions does the tool support?
Table 1. Pedagogical features.

<table>
<thead>
<tr>
<th>PEDAGOGICAL FEATURES</th>
<th>ARCHITECTURE</th>
<th>INTERACTION DESIGN</th>
<th>INSTRUCTIONAL STRATEGY</th>
<th>LEARNING SUPPORT</th>
<th>INTERFACE DESIGN</th>
</tr>
</thead>
</table>
| EXPERIENCE BUILDER    | Branching – multiple choice. | All clicking – no input capabilities. | Content design and production are built on a behavioural model. Methodology is built into authoring process:  
* Identify common situations found on the job.  
* Design a model that prescribes best practices.  
* Create a practice environment that allows learners to try these practices.  
* Offer context-sensitive tutoring.  
Author from this behavioural model as opposed to building the simulation page by page.  
Prescriptive learning - not discovery. | Supports “just-in-time” coaching when learner makes a mistake.  
Options for help, additional coaching.  
Supports links to further resources. | Can deliver in Flash or HTML.  
Interface design can be original or customized template.  
Interface is developed separately from content; can change interface without having to modify content. |
| FORCE TEN             | Architecture can vary. Tool supports creation of all types of learning products, not just simulations.  
However, all content must be specified and created, so the structure of any simulations created would be closed or branching. | All clicking and mouse based – no input capabilities. | Specific types of objects have been pre-designed to support soft skills simulation development.  
* A slide show object.  
* A branching object.  
* A flexible multiple-choice question object.  
* A content review object. | Self service knowledgebase provides multiple access pathways to content such as learning objects, help, tutorials and other resources. | Can be customized or template-based.  
WebClass real-time tool enables virtual conferencing and collaboration with whiteboard, chat, document and application sharing. |
<table>
<thead>
<tr>
<th>PEDAGOGICAL FEATURES</th>
<th>ARCHITECTURE</th>
<th>INTERACTION DESIGN</th>
<th>INSTRUCTIONAL STRATEGY</th>
<th>LEARNING SUPPORT</th>
<th>INTERFACE DESIGN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SIVOX</strong></td>
<td>Architecture is limited in that the learner must follow a specific path. However, the voice recognition feature enables a back and forth type of progress. The learner’s path through the simulation can be interrupted anytime and re-started from anywhere in the simulation.</td>
<td>Interaction design is based largely on interactive screens imported from client’s actual system. Voice interaction design is template-driven by tool. Uses Natural Language Processing – unique capability.</td>
<td>Learners learn by doing. There are three sections in each session. Each includes instructions, training and completing a practice phone call. Trainees learn software system operation in parallel with telephone skills.</td>
<td>A speech recognition engine listens to what learners say and provides them with immediate feedback via a “Real-Time Coach.” Their data input abilities and multitasking skills are also monitored and critiqued. Learners can retry an exercise immediately after receiving coaching and feedback. Those who need more information are referred to the Knowledge Center where they can access more resources.</td>
<td>Tied to system software being emulated during training.</td>
</tr>
<tr>
<td><strong>UNREAL TOURNAMENT</strong></td>
<td>Open. Although end goals are pre-programmed, the way the learner achieves them may vary tremendously. This strength comes from artificial intelligence capabilities lacking in most other tools.</td>
<td>Completely variable and programmable. Supports customizable autonomous agents. Generally supports keyboard and mouse interactions, but may offer voice recognition capabilities for other user input.</td>
<td>No support for instructional design included in mod or tool design. Players learn by doing. Game can include increasing levels of difficulty.</td>
<td>Game rules can be provided. Help is available, as are hints. Community support forums may be created outside the game.</td>
<td>Highly customizable.</td>
</tr>
<tr>
<td>PEDAGOGICAL FEATURES</td>
<td>MOTIVATION</td>
<td>RICH MEDIA CAPABILITIES</td>
<td>USABILITY</td>
<td>FEEDBACK</td>
<td>REFLECTION</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td>-------------------------</td>
<td>-----------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>EXPERIENCE BUILDER</td>
<td>Not addressed – seems to be no support or design advice given.</td>
<td>Audio, Video, Flash, WMV, Real Player, Local or streaming media.</td>
<td>Automated audit tools for quality control. Offers the &quot;Issue Tracker&quot;, an on-line bug reporting and tracking tool.</td>
<td>Push and pull feedback – can be triggered automatically by actions or mistakes by learner. Or can be sought out independently by learner. Can include meters showing scores, usually on a 1-5 scale, for both the last decision and the overall performance during the scenario.</td>
<td>Summative reflection by coach at the end of the simulation. No support for learner or self-reflection.</td>
</tr>
<tr>
<td>FORCE TEN</td>
<td>Not addressed.</td>
<td>Audio, Video, Cult 3-D, Flash, ViewPoint, QuickTime, Real Media, Streamcam</td>
<td>Quick prototyping available. Commenting capability for collaborative authoring.</td>
<td>Push and pull feedback – can be triggered automatically by actions or mistakes by learner. Or can be sought out independently by learner. Can be completely customized (E.g. different feedback for first attempt vs. second attempt). Summative feedback with links to relevant content at the end of an exercise or test.</td>
<td>Knowledge Sharing Facility enables peer sharing by employees of personal experiences, issues, best practices, lessons learned and other resources.</td>
</tr>
<tr>
<td>SIVOX</td>
<td>Not addressed.</td>
<td>Audio.</td>
<td>Not addressed.</td>
<td>Immediate feedback provided via the “Real-Time Coach.” Can be provided aurally or by text.</td>
<td>No built-in support for reflection. However, reflection is encouraged through use of immediate feedback.</td>
</tr>
<tr>
<td>UNREAL TOURNAMENT</td>
<td>Not addressed specifically. Some motivation is inherent in the game play, which supports the achievement of game goals – to win. Learners can play against each other or the system.</td>
<td>State-of-the-art animation, graphics, audio, networking and AI.</td>
<td>Game engine is well-tested with actual game. However, no usability test or debug processes are built-in. Also, problems exist with code implementation and object-oriented class structures. Dynamic feedback on player and game status provided. Hints may be programmed.</td>
<td>No built-in support for reflection.</td>
<td></td>
</tr>
</tbody>
</table>
### Administrative features

<table>
<thead>
<tr>
<th>ADMIN. FEATURES</th>
<th>PROGRAMMING COMPLEXITY</th>
<th>ACCESSIBILITY</th>
<th>MODULARITY</th>
<th>INTEROPERABILITY</th>
<th>DELIVERY PLATFORM</th>
<th>EVALUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPERIENCE BUILDER</td>
<td>WYSIWYG authoring.</td>
<td>No plug-in required (native to the Web browser).</td>
<td>Simulation content can be Reused.</td>
<td>AICC-compliant</td>
<td>PC, MAC, UNIX</td>
<td>Level of data automatically captured during the simulation:</td>
</tr>
<tr>
<td></td>
<td>Frequent online prototyping available.</td>
<td>Finished simulations install on any type of server or on CD-ROM.</td>
<td>Quality and version control are largely automated.</td>
<td>SCORM conformance</td>
<td></td>
<td>• Start time.</td>
</tr>
<tr>
<td></td>
<td>Tool uses HTML templates – finished simulations are therefore “open” and can be modified later in programs like DreamWeaver.</td>
<td>Developer owns the finished product.</td>
<td>Existing assets can be imported into simulation.</td>
<td>Version 1.2</td>
<td></td>
<td>• End time.</td>
</tr>
<tr>
<td></td>
<td>Tool offers web-based collaborative design.</td>
<td></td>
<td></td>
<td>Version 1.1</td>
<td></td>
<td>• Date (date last accessed).</td>
</tr>
<tr>
<td></td>
<td>Supports three levels of authoring authority.</td>
<td></td>
<td></td>
<td>Not interoperable with third-party LMS solutions.</td>
<td></td>
<td>• Completion status (passed/failed).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Composite score for simulation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Completion status and/or score for each step performed in the simulation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• A list of steps performed incorrectly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Can also track multidimensional scores.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Supports use of certification assessments to conclude training.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Learners may be able to access progress data and scores, including before and after performance.</td>
</tr>
<tr>
<td>ADMIN. FEATURES</td>
<td>PROGRAMMING COMPLEXITY</td>
<td>ACCESSIBILITY</td>
<td>MODULARITY</td>
<td>INTEROPERABILITY</td>
<td>DELIVERY PLATFORM</td>
<td>EVALUATION</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------</td>
<td>---------------</td>
<td>------------</td>
<td>------------------</td>
<td>-------------------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| FORCE TEN      | WYSIWYG authoring.      | No plug-in required (native to the Web browser). | Object-oriented system. | AICC-compliant     | PC, MAC, UNIX, Linux | Level of data automatically captured during the simulation:  
|                | Forms-based approach.   | Knowledge Sync allows learners to use courses offline, then automatically synchronizes results once reconnected to the network. | Content and structure separate, but stored together in a single Repository. | SCORM conformance | * Start time.  
|                | Drag-and drop.          | Rapid Translation Facility can translate resources and simulations into other languages. | | SCORM 2004 Version 1.2 | * End time.  
|                | Browser-based.          | | | Version 1.1 | | * Duration (how much time was spent in the simulation).  
|                | Content is delivered dynamically. | | | Interoperable with third-party LMS solutions: | | * Date (date last accessed).  
|                | Bottom-up or top-down approach. | | | KnowledgePlanet Enterprise Learning Suite. | | * Completion status (passed/failed).  
|                |                        | | | Oracle iLearning. | | * Composite score for simulation.  
|                |                        | | | Saba Enterprise Learning Suite. | | * Completion status and/or score for each step performed in the simulation.  
|                |                        | | | TRACCESS. | | Survey Facility collects user feedback on course structure and content.  
|                |                        | | | PeopleComeFirst | |  
|                |                        | | | Can import, metatag and reuse content developed in other formats: | |  
|                |                        | | | PowerPoint | |  
|                |                        | | | Dreamweaver | |  
|                |                        | | | Toolbook | |  
|                |                        | | | Export content to Word for viewing and printing. | |  
|                |                        | | | View content via PDA portal. | |  

<table>
<thead>
<tr>
<th>ADMIN. FEATURES</th>
<th>PROGRAMMING COMPLEXITY</th>
<th>ACCESSIBILITY</th>
<th>MODULARITY</th>
<th>INTEROPERABILITY</th>
<th>DELIVERY PLATFORM</th>
<th>EVALUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIVOX</td>
<td>Tool is Java and Web-based. Uses standard text inputs. Claim that no programming skills are required.</td>
<td>Requires learner to use a plug-in or thin-client player.</td>
<td>Audio segments can be moved, copied and re-used. Sound clips can be created via text-to-speech tools or captured via a microphone.</td>
<td>AICC-compliant SCORM conformance SCORM 2004 Version 1.2 Interoperable with third-party LMS solutions: Moodle Various proprietary LMS's All data is stored in Oracle and SQL compatible files.</td>
<td>PC</td>
<td>Simulation operates in two modes - Practice and Assessment. Over 110 trainee metrics can be tracked, including trainee voice quality and clarity, accuracy of system tasks and call times.</td>
</tr>
<tr>
<td>UNREAL TOURNAMENT</td>
<td>UT open source game code is written in UnrealScript, a bytecode-compiled scripting language. Graphics engine is proprietary. Mod can be created by highly experienced or expert programmers only. Development process can be divided into three phases: map construction, model construction and code implementation.</td>
<td>Need high end/state-of-the-art hardware to host and run simulation. Require broadband access for networked play.</td>
<td>Object-oriented. Modules handle input, output (3D rendering, 2D drawing, sound), and generic physics/dynamics for game worlds.</td>
<td>No.</td>
<td>Can author for PC or Mac - standalone or networked. Simulations can be modified to accommodate new display types such as CAVES or HMD’s.</td>
<td>Tracks scores, levels and goals achieved.</td>
</tr>
</tbody>
</table>
5 FUTURE RESEARCH

Research in this field still in its infancy. Although theory and intuition tell us that simulations should offer effective and engaging learning experiences, there is almost no evidence to support or tell us why this should be so. Nor do we have a solid understanding of how best to structure such environments or support the learner during the simulation (Mantovani, 2003).

From a technical point of view, Howell (2003) offers suggestions for several key areas needing further research: improving interoperability; facilitating the reuse, updating, and maintenance of objects and assets; adapting simulations to learning environments, including those involving virtual and collaborative learning; and developing navigation techniques in virtual environments.

It appears that much of the research in both industry and academia continues to focus on overcoming such technical challenges. Yet there is another aspect of simulation authoring that is not addressed by most of the currently available tools, which also needs to be investigated: how to provide support within the authoring process to help educators develop simulations that are pedagogically sound and offer effective learning experiences. Researchers know that issues such as motivation, reflection, different learning styles, conceptual models, etc. can have a profound impact on the effectiveness of the learning experience. Discovering how to manage these design elements with educational simulation authoring tools is an area in need of much exploration.

Research into the long-term effectiveness of simulation training would help educators support their case to include this type of approach in their learning strategies. Several vendors have done follow-up studies; however, these are all linked to ROI, not learning. Many different questions could be explored here. How do learners transfer their knowledge from the simulation to their real world environment? What kinds of learning transfer best? How can this transfer be supported after the learner finishes with the simulation?
Questions also arise about the use of rich media. Many authoring tools currently support the use of rich media such as video, audio and Flash animation. However, it appears that few simulation developers choose to use such media in their programs. Since these types of assets are not overly difficult or expensive to create, it would be useful to discover why developers are choosing to ignore them, when they can bring added value to the learning experience.

We also need to focus more of our research efforts in the simulation field on adult learners. Although research is currently in progress on how to use games to facilitate learning, with some consideration of simulations, most of this focuses on children and learners in formal academic settings. We need to develop a better understanding of how simulations can support adult learners, in both structured and informal environments. Adult learning knows no boundaries; building knowledge on how simulations can facilitate lifelong learning will benefit all sectors of society.

Finally, there is on-going discussion in the education field regarding a potential shift in how we think about learning. Increasingly, educators talk about viewing learning as a process as opposed to an end result or a set of outcomes (Galarneau, 2005; Sefton-Green, 2004). Such a transformation could radically alter the way we design learning experiences, affecting our perspectives on the role of content and curriculum, teachers and facilitators, performance and evaluation and the nature of learning itself. Continuing to explore these ideas could lead to significant findings that would affect our approach to education for today’s learners and those of the future.
REFERENCES


