SUMMARY OF DRILLS

Use a short title page.

Provide complete directions and allow the learner to return to them at any time.

Use the organization principle of memory as well as the repetition principle.

Use item types that both enhance response economy and accomplish your aims. Mouse selection, single-word, and numeric answers are usually good.

Consider mixed-mode presentations and responses for generalization and transfer.

Item presentations should be lean, have good layout, and use proper spelling, grammar, and punctuation.

Use graphics as a context, as prompts, as feedback, and as a motivator.

Keep item difficulty fairly constant in a drill session.

Use variable interval performance (VIP) queuing for item or algorithm selection.

Retire items based on mastery, such as elimination from the future queue in VIP queuing.

Terminate drills permanently based on performance, such as complete exhaustion of the future queue.

Allow temporary termination at any time based on learner request, and allow restarting where the learner left off.

Allow help requests and requests to see the answer.

Judge intelligently.

Give format feedback when the response format is wrong. Don't consider the response incorrect; rather, give another try.

Use answer markup for constructed responses that are partially correct.

Give a short confirmation when the response is correct.

Give immediate corrective feedback when the response content is incorrect.

Keep feedback short and positive.

Keep the pace of a drill quick.

Design drill sessions to last about fifteen minutes.

Provide special feedback and queuing for discrimination errors.

If subdrill-grouping is being used, select items for each subdrill based on equal difficulty, to minimize discrimination errors and to take advantage of the organizational principle of memory. In a drill session, select items from a single subdrill group except for some review items from completed subdrills.

The endless-continuum technique may be used as an alternative to subdrill-grouping. It avoids the problems of repetitious drill endings and adjusting session length. Drill sessions should still be terminated at about fifteen minutes, and learners should be kept informed of progress.

Increase motivation by using games, cooperative learning, competition, setting reasonable and relevant goals, progress reporting, display and response variety, adjunct reinforcement, and short drill sessions. Be careful that competition does not discourage poorer learners.

CHAPTER 7

Simulations

Introduction

Multimedia simulation is an increasingly popular method for learning. Simulations are perceived as more interesting and motivating than many other methodologies, a better use of computer technology, and more like "learning in the real world." This chapter describes various types of simulations, discusses their advantages, analyzes factors critical to their design, and suggests some activities unique to their development.

Confusion and disagreement about what is and what is not an educational simulation are considerable. An educational simulation can be defined as a *model* of some phenomenon or activity that users learn about through interaction with the simulation. This definition of simulation *embodies* several new techniques, such as some types of microworlds (Brehmer & Dorner, 1993; Li, Borne, & O'Shea, 1996; Rieber, 1996; Underwood, Underwood, Pheasey, & Gilmore, 1996), virtual reality (Milheim, 1995; Psotka, 1995), and case-based scenarios (Jarz, Kainz, & Walpoth, 1997). Proponents of these new approaches often maintain that they are not simulations, pointing out one difference or another, such as that microworlds may represent imaginary or impossible phenomena. Nevertheless, if a program has the critical features of simulations, the designer should consider it a simulation and should apply simulation theory and design principles.

Our definition of simulation also *excludes* some formats, such as movies, animations, and many types of games. Although these often contain some imitation or representation of reality, they typically are not based on an internal model (which movies lack, for example), or lack the main goal of users learning about a model (as is the case with many games). Many programs are called *simulation games* (Faria, 1998; Herz & Merz, 1998). This is an apt term when the program meets the definition of simulation (learning through interaction with an underlying model) and has the characteristics of a game (competition, rules, winning and losing). However, a program that uses a model or activity solely as a motivating device (for example, being a detective in the *Carmen Sandiego* programs) is usually classified solely as a game or drill.

A simulation doesn't just replicate a phenomenon; it also simplifies it by omitting, changing, or adding details or features. This is a critical point. Many simulation designers suggest that the more accurate the representation, the better the simulation (Duchastel, 1994). Although this may be true for simulations in engineering and research, it is not the case for educational simulations. Using simplified models, learners may solve problems, learn procedures, come to understand the characteristics of phenomena and how to control them, or learn what actions to take in different situations. In each case, the purpose is to help learners build their own mental models of the phenomena or procedures and provide them opportunities to explore, practice, test, and improve those models safely and efficiently. This can be done more effectively when the model is simplified.

In addition to simplifying models, educational simulations may *add* elements not present in the real world. Coaching (Acovelli & Gamble, 1997), providing feedback or hints, and similar techniques help make complex phenomena or procedures easier and more comprehensible to beginning learners. The technique of adding or highlighting particular elements has been termed *augmentation of reality* in the simulation theory of de Jong and van Joolingen (1998).

The topic of simulations for learning is challenging and fascinating. Simulations are more difficult to design and develop than methodologies discussed previously, but the benefits in terms of user satisfaction and learning can also be much greater.

Types of Simulations

Several approaches have been suggested for categorizing educational simulations (Gibbons, Fairweather, Anderson, & Merrill, 1997; Goodyear, Njoo, Hijne, & van Berkum, 1991; Reigeluth & Schwartz, 1989; Towne, 1995; van Joolingen & de Jong, 1991). We divide simulations into two groups according to whether their main educational objective is to teach *about* something or to teach *how to do* something. The *about something* group can be subdivided into two subcategories, *physical* and *iterative* simulations, and the *how to do something* group into two subcategories, *procedural* and *situational* simulations. This can be summarized as follows:

About something simulations
  Physical
  Iterative

How to do something simulations
  Procedural
  Situational

This classification into four categories is useful for several reasons. First, because there is disagreement as to what is meant by simulation, it helps clarify the terminology. The word *simulation* has different connotations to people of different disciplines. When civil engineers or economists refer to a simulation, they likely mean an iterative simulation. Psychologists and businesspeople typically mean situational simulations whereas training professionals generally mean physical or procedural simulations.

Second, categorizing simulations allows us to identify and discuss what factors are of greater or lesser importance for each category, which assists designers of simulations by placing emphasis and effort on what should yield the greatest benefit.

Some simulations are difficult to assign to single categories. For example, flight simulator programs appear to fit in both the physical and the procedural categories because they simulate an aircraft as well as the procedures involved in flying the aircraft. Classification is easier if you begin by identifying the educational objective (learning *about* versus learning *how*). This helps clarify whether a simulation is physical or procedural, or iterative or situational. For example, flight simulator programs would generally be classified as procedural simulations because the learner's primary objective is learning to *operate* the aircraft.

Although categorizing simulations is beneficial, a potential drawback is creating the impression that the categories are clearly distinct. In truth, many simulations do not fall neatly into just one category but are a synthesis of more than one type. It is not uncommon for the learner to be learning about a device (how it works), as well as how to operate that device, as might be the case for an aircraft mechanic. Nevertheless, a classification system does provide guidance to both simulation users and designers. We now describe and give examples of the four subcategories.

Physical Simulations

In *physical* simulations a physical object or phenomenon is represented on the screen, giving the user an opportunity to learn about it. Many examples are in the physical and biological sciences (gravity, optics, chemical bonding, photosynthesis, weather), in engineering (internal combustion engines, transmission of electricity through power lines, computer logic circuits), and in some social sciences (economics, urban planning, and psychology).

A popular example of a physical simulation is *SimCity Classic* (Electronic Arts, 1996a) and its related family of programs such as *SimEarth* (Electronic Arts, 1998) and *SimFarm* (Electronic Arts, 1996b). Although these are marketed as games, they are powerful simulation programs that allow learners to explore difficult topics, such as urban planning and ecology. *SimCity Classic* is illustrated in Figure 7.1. Free from any time constraints, you can build cities of varying size with different layouts of businesses, homes, roads, parks, mass transportation, and services. You can observe and analyze the effects of such layouts on population, economics, resident satisfaction, and traffic patterns. *SimEarth,* illustrated in Figure 7.2, allows you to design and modify entire planets, including landforms, oceans, and living species; analyzing the effects on populations, weather, and many other outcomes. Such phenomena take place over periods of time too lengthy to be observed in *real time* and, of course, are impossible in most cases for learners to manipulate. The fact that some of these programs contain game elements (for example, in *SimCity Classic* your city might be attacked by Godzilla, after which you must rebuild) does not detract from them being simulations. Each program has an underlying computer model of a system (a city, the earth, a farm), and the objective is to learn about those models.

Another example of a physical simulation is *Future Lab: Circuits for Physical Science* (Simulations Plus, 1998a), illustrated in Figure 7.3. In this simulation, you can

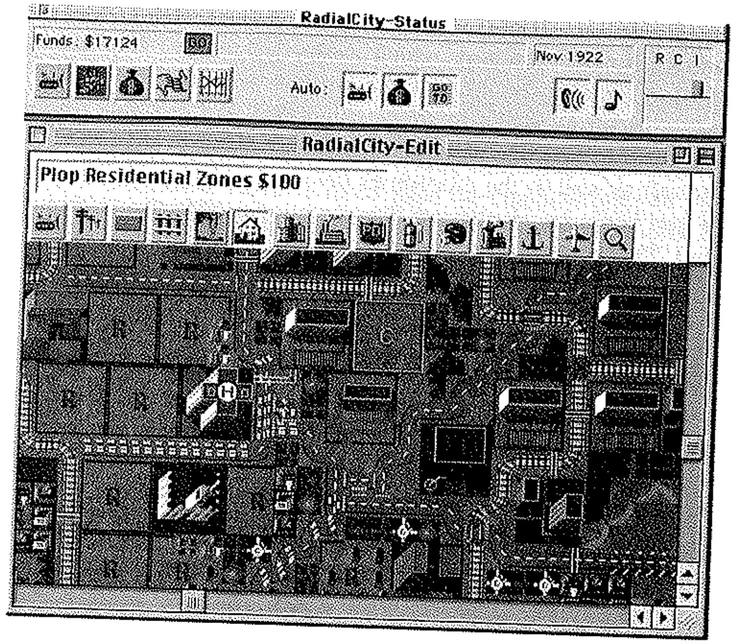


FIGURE 7.1

***SimCity Classic,* an Example of a Physical Simulation**

Courtesy of Electronic Arts, Inc. Copyright © 1996 by Electronic Arts, Inc. SimCity Classic is a trademark or registered trademark of Electronic Arts, Inc. in the U.S. and/or other countries. All rights reserved.

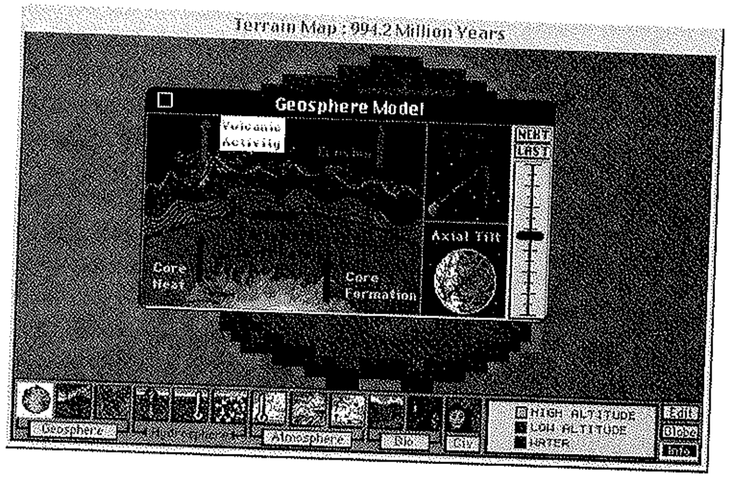


FIGURE 7.2

***SimEarth,* an Example of a Physical Simulation**

Courtesy of Electronic Arts, Inc. Copyright © 1998 by Electronic Arts, Inc. SimEarth is a trademark or registered trademark of Electronic Arts, Inc. in the U.S. and/or other countries. All rights reserved.

assemble circuits consisting of power sources, lights, switches, resistors, wires, and other components. You can turn the circuits on and off and take measurements with simulated voltmeters and ammeters. Although *SimCity Classic* and *SimEarth* exist in greatly *speeded up* time, *Future Lab: Circuits for Physical Science* occurs in real time. That is, activities take place in the same time frame that they would in a real electronics or engineering laboratory. However, working with the simulation is safer and less expensive than working in a real laboratory. Furthermore, learning is simplified because users do not have to deal with some of the subtle physical and visual manipulations, such as connecting wires and reading voltmeters.

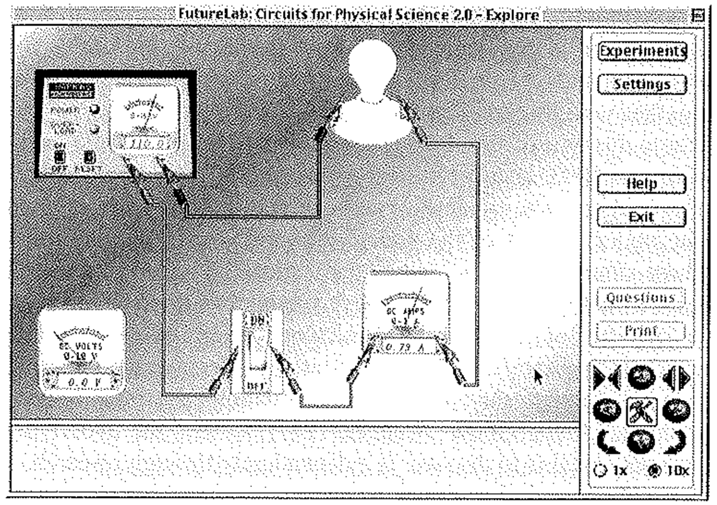


FIGURE 7.3

***Future Lab: Circuits for Physical Science,* an Example of a Physical Simulation**

Courtesy of Simulations Plus.

There are physical simulations of many other phenomena, such as the movement of stars and planets, the development of weather systems, how internal combustion engines work, the behavior of light transmitted through lenses or reflected off mirrors, how earthquakes affect the earth's surface, and so on. All such physical simulations are intended to inform learners *about* some object or phenomenon and its underlying principles. We learn from physical simulations by manipulating the various objects or variables (such as building new roads in *SimCity Classic*) and observing how the overall system changes as a result (such as the number of people who drive to work rather than take the train).

Iterative Simulations

Iterative simulations, which we previously called *process simulations* (Alessi & Trollip, 1991), are quite similar to *physical* simulations in that they teach *about* something. The primary difference is the manner in which learners interact with the simulation. Instead of continuously manipulating the simulation as it unfolds in either real or manipulated time, the learner runs the simulation over and over, selecting values for various parameters at the beginning of each run, observing the phenomena occur without intervention, interpreting the results, and then running it all over again with new parameter values.

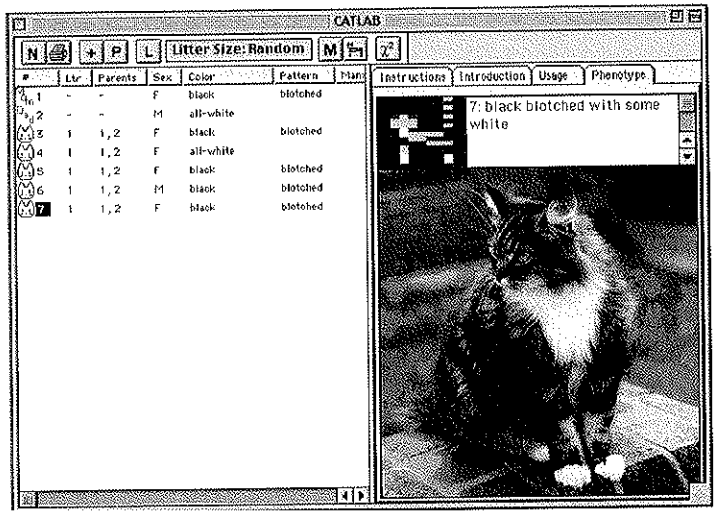
Time is generally *not* included as a variable in iterative simulations. That is, whether the real phenomenon occurs very quickly or very slowly, in iterative simulations the learner manipulates parameters, runs the simulation, and sees immediate results. Some simulation theorists refer to this as a *static* (in contrast to a *dynamic*) time frame because the user is not manipulating phenomena as time continually passes (van Joolingen & de Jong, 1991). Rather, time "stops" between runs and nothing happens while the user decides on parameter values for the next run. The ability of simulations to manipulate time (speed it up, slow it down, or freeze it) is one of their great educational benefits. Some actions happen too fast to see, such as the movement of electrons in a wire.

the dynamics of a country's economy over a decade. It is much easier for learners to conceptualize what is occurring when it is presented in a time frame that *highlights the changes*. Once the learner understands the process, the true rate of occurrence can be introduced or explained, together with its ramifications.

Some researchers refer to this type of simulation as *scientific discovery learning* simulation (de Jong & van Joolingen, 1998) because the learner is essentially engaging in simulated scientific research, applying the scientific method and performing repeated experiments to arrive at an understanding of the underlying model of a scientific phenomenon. The intent of these simulations is not to *tell* the learner the underlying model but to have the learner figure it out by doing research. Such simulations often have a double objective, learning the specific content (such as physics or economics) and learning about scientific methods (such as hypothesis formation and testing). Because the scientific discovery learning approach is generally used with iterative simulations, they may be considered "black box" simulations (Alessi, 2000a; Wenger, 1987) because the underlying model is hidden and the learner's goal is to discover it independently through research.

Iterative simulations are often used (though not exclusively) for teaching about processes that are not directly or easily visible, such as economics (the laws of supply and demand) and ecology (changes in populations over long periods of time). In most cases, we see only the numeric outcomes, such as the prices of goods or number of people in various countries. However, iterative simulations are also used for some more visible phenomena, such as in physics or manipulating parameters in a mechanics laboratory.

Catlab (Kinnear, 1998), shown in Figure 7.4, is an example of an iterative simulation in biology. The learner chooses the initial physical characteristics of a female and a male cat, such as fur color and pattern. The cats mate and have kittens, a process that is speeded up so the kittens arrive in a few seconds rather than after the normal nine weeks. The learner then has a litter of kittens with characteristics derived from its parents, according to the laws of genetics. That is, the kittens are not identical to the parents, but share some of their characteristics. The process is then repeated, with the learner choosing to mate two new cats from any of those available. These include the original parents and all generations of their offspring, which are assumed to mature immediately. The purpose of the simulation is to become familiar with genetic research and the laws of genetics. This is accomplished by generating hypotheses about how physical characteristics of cats are inherited, and by testing each hypothesis through observing the results of each mating. Many generations of kittens have to be produced before you have enough information to understand how the genetic laws of inheritance operate. In the simulated world, this can be accomplished in a matter of minutes.



#	Ltr	Parents	Sex	Color	Pattern
1	-	-	F	black	blotched
2	-	-	M	all-white	
3	1	1, 2	F	black	blotched
4	1	1, 2	F	all-white	
5	1	1, 2	F	black	blotched
6	1	1, 2	M	black	blotched
7	1	1, 2	F	black	blotched

7: black blotched with some white

FIGURE 7.4

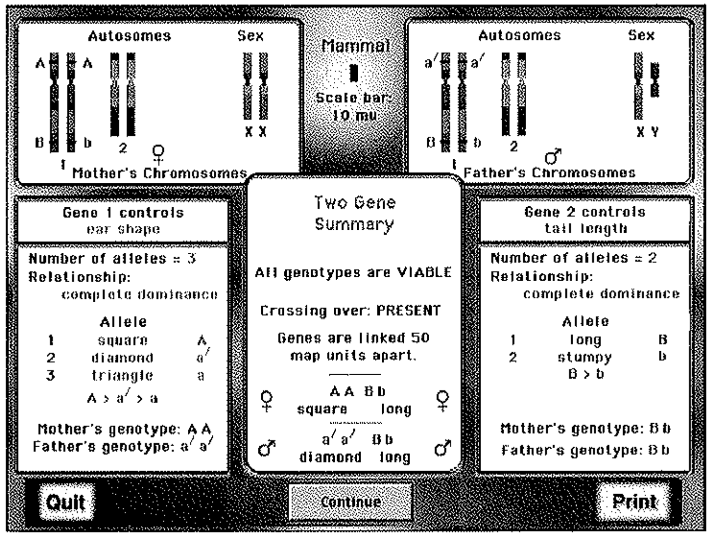
***Catlab*, an Example of an Iterative Simulation**

Courtesy of EME Corporation, Stuart, Florida.

Kangasaurus: Transmission Genetics (Kinnear, 1997), illustrated in Figure 7.5, is an iterative simulation closely related to *Catlab.* Whereas in *Catlab* the learner is restricted to the visible characteristics of cats, *Kangasaurus* allows the learner to see and manipulate the underlying genetic variables and characteristics that we cannot see with the naked eye, such as chromosomes, alleles, and linkages. It not only manipulates time to enhance learning, but makes visible the invisible. Both these types of simulation are beneficial in the biological sciences, in which scientists deal with visible macroscopic objects as well as the underlying invisible microscopic objects, and in which speeding up time makes learning more efficient and effective.

Iterative simulations are particularly useful in learning ecology and population dynamics. *Community Dynamics* (Lopez, 1998) is illustrated in Figure 7.6. Learners can manipulate the birth rates and death rates of predators and their prey, among other things and observe how their populations rise and fall. Similarly, *Population Concepts* (Lopez, 1994), shown in Figures 7.7 and 7.8, allows the learner to vary birth rates, the carrying capacity of the environment, and other variables, and to observe changes in population over time.

Future Lab: Gravity for Physical Science (Simulations Plus, 1998b), an example of an iterative simulation in the physical sciences, is illustrated in Figure 7.9. The learner can perform experiments relating to gravitational forces and mechanics. In the illustration, the learner is dropping balls of different mass and recording the time that they pass



Gene 1 controls
ear shape

Number of alleles = 3
Relationship:
complete dominance

	Allele	
1	square	A
2	diamond	a′
3	triangle	a

A > a′ > a

Mother's genotype: A A
Father's genotype: a′ a′

Two Gene
Summary

All genotypes are VIABLE

Crossing over: PRESENT

Genes are linked 50
map units apart.

♀ A A B b square long ♀

♂ a′ a′ B b diamond long ♂

Gene 2 controls
tail length

Number of alleles = 2
Relationship:
complete dominance

	Allele	
1	long	B
2	stumpy	b

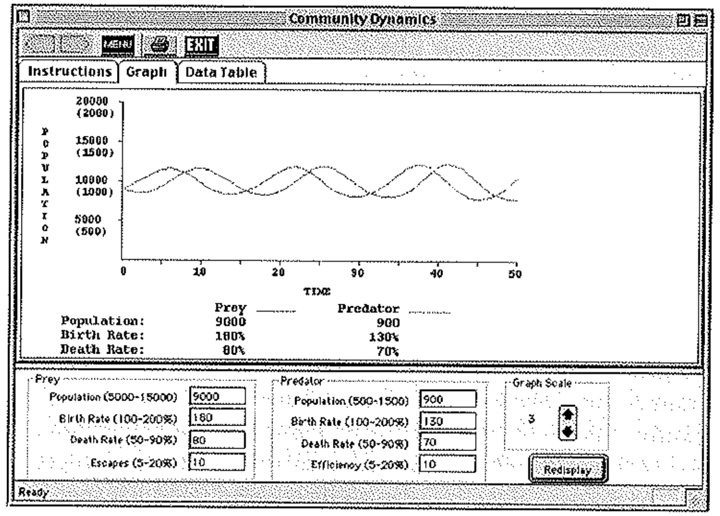
B > b

Mother's genotype: B b
Father's genotype: B b

FIGURE 7.5

***Kangasaurus*, an Iterative Simulation**

Courtesy of EME Corporation, Stuart, Florida.



	Prey	Predator
Population:	9000	900
Birth Rate:	180%	130%
Death Rate:	80%	70%

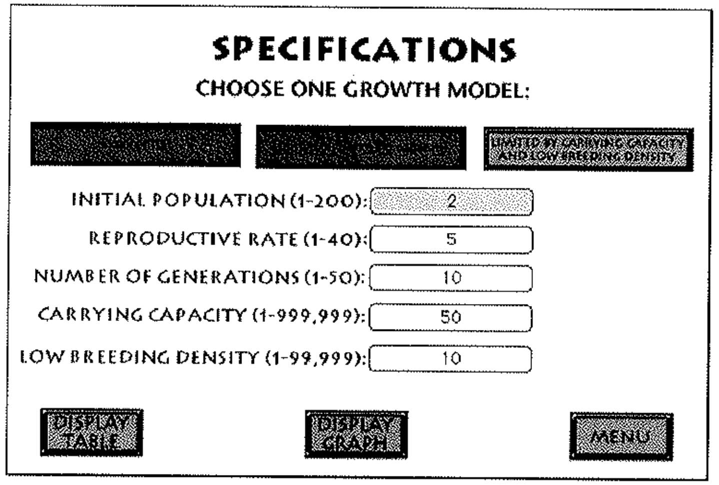
Prey
Population (5000-15000): 9000
Birth Rate (100-200%): 180
Death Rate (50-90%): 80
Escapes (5-20%): 10

Predator
Population (500-1500): 900
Birth Rate (100-200%): 130
Death Rate (50-90%): 70
Efficiency (5-20%): 10

FIGURE 7.6

***Community Dynamics,* an Iterative Simulation**

Courtesy of EME Corporation, Stuart, Florida.



INITIAL POPULATION (1-200): 2
REPRODUCTIVE RATE (1-40): 5
NUMBER OF GENERATIONS (1-50): 10
CARRYING CAPACITY (1-999,999): 50
LOW BREEDING DENSITY (1-99,999): 10

FIGURE 7.7

Setting Parameters in *Population Concepts,* an Iterative Simulation

Courtesy of EME Corporation, Stuart, Florida.

various points. The measurement instruments show the times, allowing the learner to calculate velocity. The learner can investigate, for example, the velocity and acceleration of different masses, with and without air and on different planets. The benefit of doing these experiments with a simulation is that the user can complete many more trials with less effort than could be done in a laboratory with real objects. The learner can easily compare the relative velocities under different conditions. In a real laboratory, the learner can only deal with a limited range of height and velocity, cannot manipulate friction or other parameters, such as gravity, and cannot easily measure a falling object. *Future Lab: Gravity for Physical Science* clearly has the characteristics attributed to the category of physical simulation, but because of its method of interaction and treatment of the time frame, it is classified as an iterative simulation.

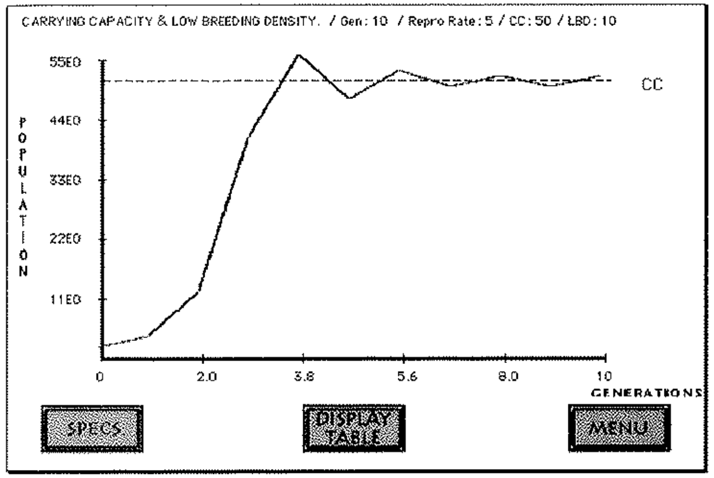

CARRYING CAPACITY & LOW BREEDING DENSITY. / Gen: 10 / Repro Rate: 5 / CC: 50 / LBD: 10

FIGURE 7.8

Observing Results in *Population Concepts,* an Iterative Simulation

Courtesy of EME Corporation, Stuart, Florida.

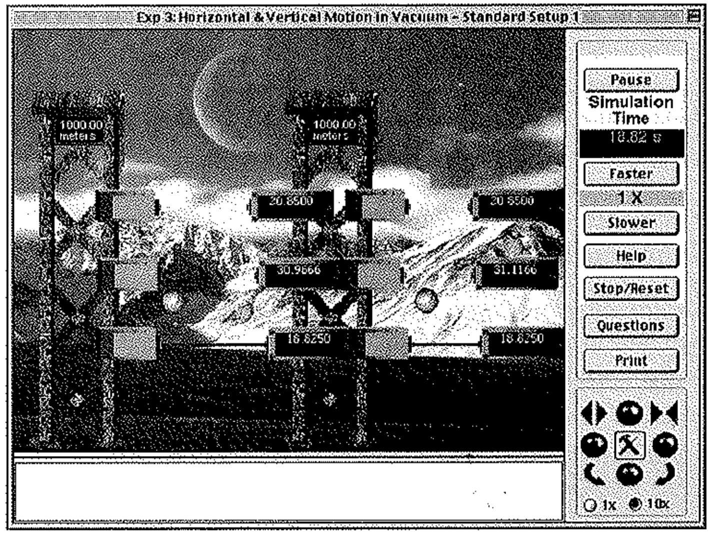


FIGURE 7.9

An Iterative Simulation in Physics *Future Lab: Gravity for Physical Science.*

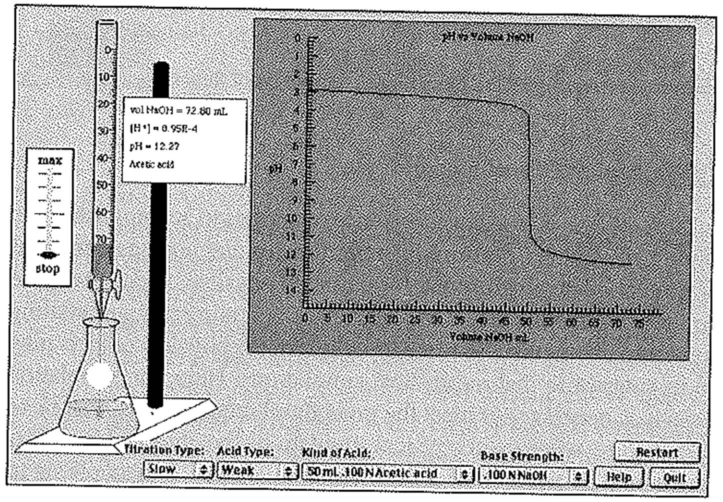
Courtesy of Simulations Plus.

Procedural Simulations

The purpose of procedural simulations is to teach a sequence of actions to accomplish some goal. Examples include flying an airplane, performing a titration, or diagnosing equipment malfunctions. Procedural simulations typically contain simulated physical objects, because the learner's performance must imitate the actual procedures of operating or manipulating them. However, it is important to distinguish between the role that the physical objects play in this type of simulation in contrast to that in physical simulations. Here, the simulation of the various physical objects is necessary to meet the procedural requirements, that is, to allow engagement in the *procedure,* whereas in physical simulations the objects *themselves* are the focus of the instruction. The purpose of a titration simulation, for example, is to teach science students how to obtain measurements for calculating the strength of acids, not to show what the apparatus looks like, although it may serve this function too. As stated earlier, the primary objective of a procedural simulation is to teach the learner *how to do something,* whereas a physical simulation is designed to teach *about* something or how something works.

An important type of procedural simulation is the laboratory simulation (McAteer et al., 1996), which might more appropriately be called the *prelaboratory simulation.* Such simulations are not generally intended to replace the learning of a real laboratory activity, but to introduce it and prepare the learner for it. A good example is *Burette* (EME, 1999), which teaches the procedures of performing a chemistry titration. *Burette* is shown in Figure 7.10. Sometimes laboratory simulations *are* intended to replace (or reduce the frequency of) actual laboratory experiments. *BioLab Frog* (Pierian Spring Software, 1997), shown in Figure 7.11, allows the learner to dissect a frog and perform other experiments without hurting live frogs. Experiments with animals raise ethical concerns as well as being expensive, so many people welcome such a simulated environment. Although some occupations may require learning with real animals (for example, veterinarians), the majority of learners can benefit just as much from simulations. (Surveys among frogs indicate that *they* overwhelmingly prefer the use of simulations.) More importantly, procedural simulations are increasingly being used in the health education fields (including medicine, dentistry, nursing, and surgery) to teach critical procedures without pain or risk to human patients. Dependence on the use of cadavers for teaching anatomy and physiology is being decreased through the use of anatomy simulations such as *A.D.A.M. Interactive Anatomy* (A.D.A.M. Software, 1997).

Another common type of procedural simulation in medicine is the *diagnosis* simulation (Johnson & Norton, 1992). The learner is presented with a problem to solve, such as a patient with particular symptoms, and must follow a set of procedures to determine the solution, in this case determining the illness. Diagnosis simulations are also common in military training to teach how to diagnose malfunctions in mechanical and electronic equipment and how to correct malfunctions. A large portion of military training involves using and maintaining sophisticated equipment, much of which is costly and dangerous. Procedural simulations have become an indispensable part of such training.



vol NaOH = 72.80 mL
[H+] = 0.95E-4
pH = 12.27
Acetic acid

Titration Type: Slow
Acid Type: Weak
Kind of Acid: 50mL .100N Acetic acid
Base Strength: .100N NaOH

FIGURE 7.10

***Burette,* a Procedural Simulation**

Courtesy of EME Corporation, Stuart, Florida.

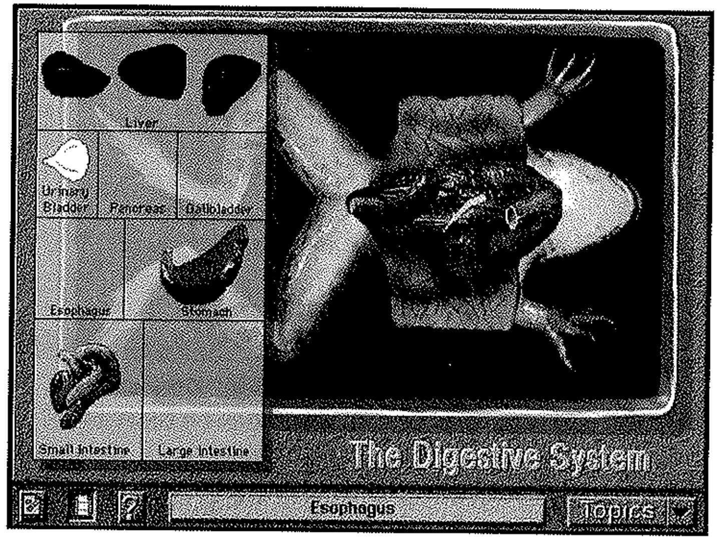


FIGURE 7.11

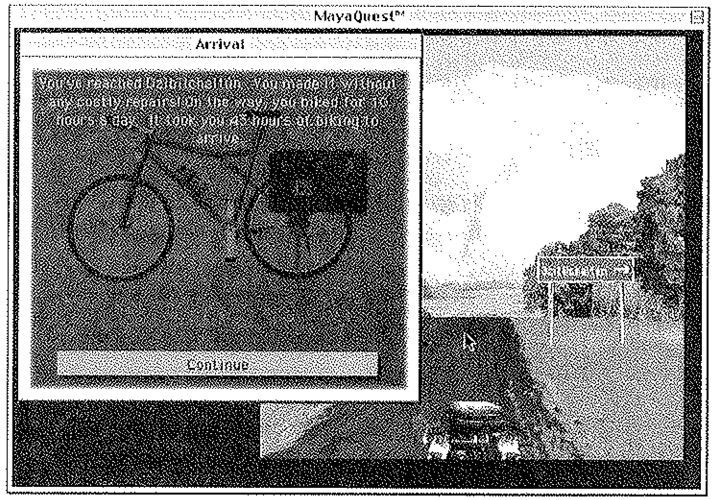
***BioLab Frog,* a Procedural Simulation**

Courtesy of Pierian Spring Software. Copyright © 2000 Pierian Spring Software.

One of the most popular home computer simulations is the flight simulator, such as *Microsoft Flight Simulator* (Microsoft, 1989). Although these microcomputer simulations are primarily for home entertainment, the military and commercial aviation industries depend on sophisticated aircraft simulators to train and test real pilots. Simulators for large commercial jet aircraft may cost tens of millions of dollars. However, the real aircraft can cost a hundred million dollars or more. Not only are the simulators much less expensive than real aircraft, but they do not require expensive fuel to "fly"; they do not endanger pilots; and they permit exhaustive practice in rare emergency procedures, such as what to do when the engines fail. Just as modern airline companies and the military have become dependent on simulations for training, so have other industries and branches of government, such as electrical utility companies (using simulators to train operation of both conventional and nuclear power plants) and NASA, which uses simulations to train all aspects of an astronaut's job.

Another type of procedural simulation is the trip simulation. *Africa Trail* (MECC, 1995a), *Amazon Trail* (Softkey Multimedia, 1996), and *MayaQuest* (MECC, 1995b) allow the user to plan and engage in trips in Africa, Central America, and South America. In the case of both *Africa Trail* and *MayaQuest* (the latter shown in Figure 7.12), the trip is taken by bicycle, and the learner must deal with the details of long-distance bicycle trips. In addition to simulating the procedures and activities involved in such trips, these programs are also entertaining vehicles for learning about the geography, culture, and history of the regions.

In all procedural simulations, whenever the user acts, the computer program reacts, providing information or feedback about the effects the action would have in the real world. Based on this new information, the user takes successive actions and each time obtains more information. As an example, consider



You've reached Dzibilchaltun. You made it without any costly repairs on the way. you biked for 10 hours a day. It took you 45 hours of biking to arrive.

FIGURE 7.12

Traveling by Bicycle in *MayaQuest,* a Procedural Simulation

Courtesy of MECC.

learner is told that a patient has been admitted to the hospital with certain symptoms, such as a high temperature, skin coloration, and a feeling of general weakness. The learner must take some action, such as choosing from available medical tests: blood test, tissue samples, throat cultures, and so on. The simulation then provides the test results: for example, the blood and tissue are normal, but the throat culture is not. Based on these results, the learner selects further tests and obtains more results. This procedure continues until the learner feels confident enough to enter a diagnosis. The program then provides feedback about the accuracy of that diagnosis. In some cases, the patient may be so ill that the learner must perform some medical procedures before receiving the results of tests.

Procedural simulations vary greatly in terms of what they consider to be correct or good sequences. Some simulations, such as a chemistry titration, have one preferred sequence of steps that the user should learn to perform. Others, such as a flight simulator, have many ways of reaching the same destination, though not all are equally efficient. Still others, such as *Africa Trail,* are more exploratory with many equally valuable paths. However, even in simulations with one or a few "best" sequences, learners may be encouraged to try out poor or even disastrous paths to learn about their consequences and what to watch out for. Therein lies another advantage of simulations, being able to explore questionable alternatives safely.

Situational Simulations

Situational simulations deal with the behaviors and attitudes of people or organizations in different situations, rather than with skilled performance. They may be considered a special type of procedural simulation, but it is useful to distinguish them for several reasons.

In procedural simulations, learners are encouraged to explore alternatives and see their effects even when there are preferred procedural paths. This is even more true concerning interactions with other people and organizations. The behavior of people and organizations is not as predictable as that of machines and physical (that is, not living) objects. To interact successfully with them requires an understanding of this unpredictability. Simulations that teach such interactions must exhibit some degree of probabilistic (or even random) behavior. For example, in a parenting simulation, scolding a child might sometimes elicit apologetic behavior from the child and other times aggressive behavior. In a business simulation, lowering your product price far below that of competitors might result in more customers, or might decrease customers due to a perception that your product must be inferior. It is even more important in situational simulations for learners to explore alternative choices and to compare the same choices at different times.

Most situational simulations incorporate role playing. The learner is not outside the simulation watching it occur, but is one of the participants or objects within the simulation. Like traditional role playing, some situational simulations are *multiuser* simulations or games, meaning that several learners participate in the simulation simultaneously, either taking turns at a single computer or working simultaneously on networked computers. When this is the case, the predictability of outcomes in the simulation is even more variable, and learning is greatly enhanced by engaging in the simulation several times and discussing it with the other participants.

Situational simulations have been used for training counselors, teachers (typically dealing with classroom behavior control), and lawyers. However, the most popular field for situational simulations is in business education, including marketing, contract negotiation, employee relations, and interaction with other businesses (Faria, 1998; Keys, 1997). Most of these programs are referred to as simulation games or just as business games because, as the next chapter explains further, they incorporate the features typical of games, including rules, role playing, winning versus losing, teamwork, and, most importantly, competition.

Capitalism (Interactive Magic, 1996), which is shown in Figure 7.13, and *Capitalism Plus* (Interactive Magic, 1997) allow participants to create companies, manufacture

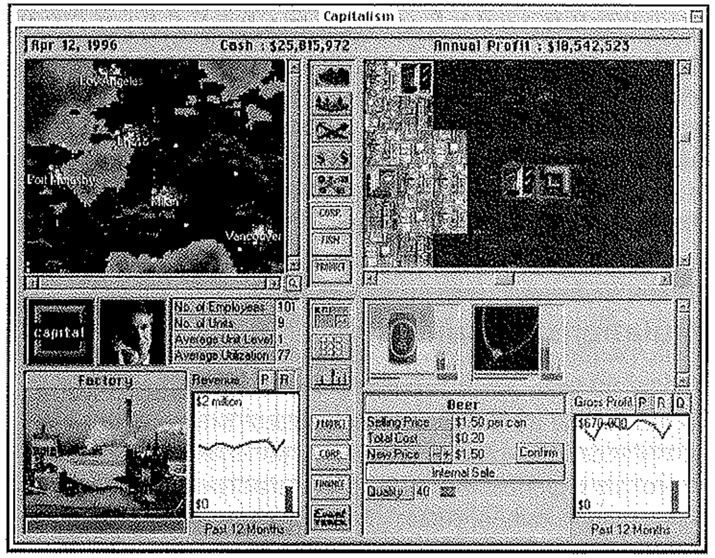


FIGURE 7.13

Setting Up a Business in *Capitalism,* a Situational Simulation

Courtesy of Trevor Chan and Enlight Software [www.enlight.com].

and market products, and compete in markets. Single or multiple participants are possible. A wide and realistic variety of activities are included, such as choosing factory sites, researching product design, dealing with suppliers, advertising, selling stock, and the like. This is but one of many simulations relating to business management. A substantial number of the articles in the journal *Simulation and Gaming* deal with business simulation games.

Another example of a situational simulation is *The Interactive Courtroom* series (Practising Law Institute, 1999), which includes simulations of interviewing, examining and cross-examining, being the judge, making motions, negotiating, and other courtroom skills. This series of lessons includes not only tutorial, practice, and assessment components, but also frequent situational simulations as well. For example, as one of the lawyers in a simulated trial sequence, you observe video of an opposing attorney speaking and may choose to object to the attorney's questions or make motions on the admissibility of evidence. Depending on the nature and timing of your objections or motions, they may be sustained or overruled by the trial judge.

Many adventure games have the characteristics of situational simulations. Once again, an adventure game can be considered a simulation game *if* one of the learning goals is to learn an underlying model. Take for example an adventure game in which you must survive in the wilderness. It can be considered a simulation game if its educational objective is to learn survival skills and techniques. If the same adventure game is used only as a motivational vehicle to reward reading or math problem solving, it is considered an educational game rather than a simulation.

Situational simulations are the least common type of educational simulation, perhaps because they are more difficult and expensive to develop, given the great complexity of human and organizational behavior. Another reason may be that learners and educators are less convinced of the effectiveness of simulations for teaching "soft" skills, such as interpersonal behavior, believing instead that on-the-job experience is better. This is an area in which research is needed.

Advantages of Simulations

The discussion above concerning types of simulations illustrated the wide variety of simulation techniques and some of their advantages. These advantages can be examined in greater detail if we divide our discussion into two parts—the advantages of simulations compared to learning in the *real world,* and the advantages of simulations compared to *other multimedia methodologies.*

Advantages of Simulations Compared to Reality

Learning with simulations has several advantages as compared to using the real world as a learning environment. We alluded to some of these in the previous section. Simulations can enhance safety, provide experiences not readily available in reality, modify time frames, make rare events more common, control the complexity of the learning situation for instructional benefit, and save money.

Many educators consider safety to be one of simulation's most important advantages. A good example is teaching someone how to operate a nuclear power plant. The power industry makes extensive use of simulation for operator training. For obvious safety reasons, it is not appropriate to use a real nuclear power plant as the learning environment in which to teach novices about the complexities of operation. If the trainee were to make a mistake, the results could be catastrophic. Similarly, nobody would consider teaching a future Boeing 747 pilot how to deal with multiple engine failure on takeoff by actually cutting the power at lift-off in a real aircraft.

Sometimes simulation is the *only* way of providing certain types of instruction. In a history course, for example, it is impossible for learners to actually witness events in the past. However, simulation can create an impression of what happened and provide role playing of historical figures. Similarly, in an economics course, simulation may be the best way for learners to recreate and analyze the events of the Great Depression.

Another attribute of simulations is that one can control aspects of reality that ordinarily make learning difficult. The best example is manipulating time for instructional benefit. In some circumstances, accelerating the passage of time is useful. For example, in simulations of genetics and inheritance, successive generations can arrive in seconds rather than hours or months. Similarly, in the study of the movement of glaciers, simulations can compress years or even centuries into minutes. Sometimes it is necessary to slow down time. In studying the movement of molecules, for example, manipulating time helps the learner see the movement, which in reality is much too rapid to observe.

It is often important to learn how to deal with rare events, but doing so in reality is obviously difficult precisely *because* of their rarity. In medicine, certain diseases occur so rarely that a medical student might never encounter them during medical school. Similarly, airplane engine failure is rare enough that it will not likely occur during a pilot's training. Nevertheless, physicians need to be able to diagnose and treat the more rare diseases, and pilots need to be able to react correctly to engine failure. In simulations, these events can occur as frequently as necessary to ensure that learners can deal with them.

Because simulations simplify reality, they can be more conducive to learning than some real environments. Real-world situations are inevitably filled with distractions. Consider again the task of learning how to fly an airplane. The cockpit of a modern airplane is one of the worst learning environments possible. Not only are many instruments facing the novice pilot, but messages are being relayed between the air traffic controllers and all planes in the vicinity, requiring careful attention to what is being said. The novice pilot is apprehensive about being up in the air and is also concerned about other aircraft nearby. All this creates a situation in which most attention is being concentrated on aspects actually irrelevant to the immediate task at hand, which is learning to control the plane. With attention thus divided, it is not surprising that it takes a long time to learn how to fly when the actual plane is used as the vehicle for learning.

Thus, the aviation industry and the military make extensive use of aircraft simulators. The most sophisticated of these, used for jet aircraft training, consist of a cockpit that replicates perfectly the aircraft it is simulating, mounted on a hydraulic motion platform, with computer monitors replacing the windows. The computer monitors can reproduce scenes outside the aircraft to whatever degree of realism is desired, and the hydraulic motion base can recreate most of the kinesthetic aspects of flying the aircraft. A realistic impression of flying an aircraft can be created even though in reality it never leaves the ground. Most of the extraneous, intrusive factors, such as air-traffic conversation, noise, and fear, do not exist in the simulation or at least can be controlled by the instructor. Thus, the student pilot is able to pay greater attention to the goals and critical cues of a particular activity. This makes simulator time at the beginning of a person's training considerably more productive than the same time spent in a real airplane.

As an example, Trollip (1979) used a computer-based simulation to teach pilots how to fly holding patterns. Because this task is primarily a cognitive one rather than one of controlling the airplane, using a simulation results in several benefits. The instruction uses time more efficiently because the simulated airplane can be repositioned for each sequence of instruction, whereas a real airplane would have to fly to the appropriate starting position each time. In this simulation, learners flew holding patterns using simulated instruments in a variety of conditions. At the end of each pattern, a simulated instructor analyzed the pattern flown, compared it to the ideal pattern for the given set of conditions, and provided both informational feedback ("Your inbound leg was more than ten degrees to the left of where it should have been") and prescriptive feedback ("You must adjust the outbound heading more for the crosswind").

A different type of complexity is the number of variables in a phenomenon. For example, *Catlab* (Kinnear, 1998) deals with just a few of the inheritable characteristics of cats. Real cats have many more characteristics than are simulated in the program. Similarly, some physical and social science phenomena have hundreds of relevant variables and cause–effect relationships. Simulations dealing with them include just the more important variables, those having the greatest effect on outcomes. This simplification of reality is often beneficial pedagogically because learners tend to be confused or overwhelmed by a large number of variables to control.

Simulations are also more convenient than their real-world counterparts. For example, they cost less, are available at any time, and are repeatable. A simulation of flying an airplane is certainly less expensive than actually flying an airplane and can be used anytime, day or night, irrespective of weather conditions. Furthermore, when an instructor can be simulated, a learner does not have to coordinate with a real instructor and can even learn at home. A simulation of diagnosing a particular disease in a patient can be done at any convenient time, whereas in reality the learner may have to wait for a patient with the relevant disease to enter the hospital. A new teacher's first year of teaching occurs only once in real life, but in a simulation the events of first-year teaching can be repeated over and over, hopefully improving the real experience when it occurs. Similarly, one can repeat the treatment of a sick patient until the appropriate tests and treatments are learned. In the real world, finding patients with identical symptoms is close to impossible.

Simulations are also more controllable than reality. As mentioned earlier, simulations are not only *imitations* of reality, but also *simplifications* of it. This is inevitable because reality is impossible to imitate in all its detail. Simplification is also instructionally advantageous. A person learns faster when details are eliminated at the beginning of instruction. For example, in an automotive diagnosis simulation, unimportant differences between engines, such as age, quality of spark plugs, and months since the last tune-up, can be ignored or eliminated. Instruction may then focus on the particular problems to be diagnosed. When real engines are used for training, learners may be distracted or misled by the minor, but irrelevant, problems all engines have.

In general, simulations of all types may facilitate initial learning by simplifying the phenomena. As a learner becomes increasingly competent in dealing with the simplified case, a simulation may then add detail to bring the learner closer to reality.

Advantages of Simulations Compared to Other Media and Methodologies

Simulations typically have four main advantages over more conventional media and methodologies such as books, lectures, or tutorials. They tend to be more motivating. They enhance transfer of learning. They are usually more efficient. Finally, they are one of the most flexible methodologies, applicable to all the phases of instruction and adaptable to different educational philosophies.

Motivation That simulations enhance motivation is well known and not surprising. Learners are expected to be more motivated by active participation in a situation than by passive observation. It is more interesting to fly a simulated airplane, for example, than to read about flying it. It is more exciting to try to diagnose and treat a simulated patient than it is to attend a lecture about it. The theories of motivation design we have discussed (Keller & Suzuki, 1988; Malone & Lepper, 1987) suggest several motivational elements that are found in most simulations. Malone's elements include *challenge* and *fantasy*. Realistic fantasy (imagining oneself in an interesting activity) is a part of most simulations and is a function of the simulation storyline or scenario. Challenge is easily maintained in simulations that increase in difficulty as the learner progresses. One of Keller's motivational elements is *relevance*. Most learners consider simulations more relevant to their learning than lectures, books, or other more passive methods, because they are engaging in the activity rather than just reading or hearing about it.

Last, many simulations include gaming techniques. As indicated earlier, these are particularly popular in business education, where the competition inherent in games is also a factor in the real world of business, including competition between firms, between management and labor, or between salespeople. Such simulation games have the potential to be more intrinsically motivating (Lepper & Chabay, 1985) than other instructional strategies.

Transfer of Learning Transfer of learning refers to whether skills or knowledge learned in one situation are successfully applied in other situations (Clark & Voogel, 1985). Simulations have good transfer of learning if what was learned during the simulation results in improved performance in the real situation. It is easy to understand why a simulation of growing a rose garden, for example, in which you manipulate soil acidity, the exposure to sunlight, and the amount of watering, would result in better transfer than would reading a gardening book. The simulation gives you practice in growing roses and the opportunity to try out different combinations of conditions and care. The book, however, only provides information and hints on how to do it. We would expect learners who use the simulation to be better prepared.

As discussed in Chapter 2, the term *transfer of learning* is often used in reference to quite different ideas. The term *near transfer* refers to applying what is learned to very similar circumstances. The term *far transfer* refers to applying what is learned to somewhat different circumstances, or *generalization* of what is learned. Simulations can be designed to optimize either type of transfer. Near transfer, which is generally more relevant to procedural (how to do it) learning, tends to be facilitated by making the training simulation as similar to the real-world work situation as possible. This is known as the theory of identical elements (Osgood, 1949). Far transfer, which is generally more relevant to declarative learning as in physical and iterative simulations, tends to be facilitated by introducing more variety into the simulation environment, including a variety of visual information, auditory stimuli, situations, and learner activities (Clark & Voogel, 1985).

Efficiency The idea of transfer of learning can be taken a step further. Not only can one measure how effectively knowledge, skills, or information transfer from one situation to another, but one can also measure how efficient the initial learning experience is with respect to the transfer. This is best illustrated with a hypothetical example.

Assume that you have two different classes for one chemistry course. To one class you give a series of interesting and informative lectures dealing with a specific laboratory procedure. To the other you give a computer program that provides the same information and includes a simulation of the laboratory. On completing their respective forms of instruction, each class of chemistry students performs the procedure in a real laboratory. Your observation of the two classes convinces you that there is no difference in performance and that both perform well. On the basis of this information you might conclude that both instructional methods have the same transfer of learning. However, if the lecture series took ten hours, and the average time to complete the simulation required only five hours, you might conclude that the simulation was more time efficient. That is, more transfer occurred per unit of learning time with the simulation than with the lectures. Although simulations don't guarantee time efficiency, there is evidence that well-designed simulations do foster it. (For more information on transfer of learning and transfer efficiency see Beaudin, 1987; Broad & Newstrom, 1992; Cormier & Hagman, 1987; Detterman & Sternberg, 1993; Foxon, 1994, 1993; Parry, 1990; and Yellon, 1992.)

Flexibility The last advantage of simulations is their flexibility. Simulations can satisfy more than one of the four phases of instruction. In fact, they usually satisfy at least two: either initially presenting material and guiding the learner through it (Phases 1 and 2) or guiding the learner through previously learned material and providing practice with it (Phases 2 and 3). Simulations are also being applied increasingly to the assessment of learning (Alessi & Johnson, 1992; Lesgold, Eggan, Katz, & Govinda, 1992; O'Neil, Allred, & Dennis, 1997; O'Neil, Chung, & Brown, 1997), although assessment is usually not combined with other phases. It is rare to find simulations that provide three or all four phases of instruction in the same lesson. However, the applicability of simulations to all phases certainly stands in contrast to most other media and methodologies.

When simulations do provide initial instruction, they frequently do so by the scientific discovery learning or experimentation approach (de Jong & van Joolingen, 1998). *Catlab* and most other iterative simulations are examples of this. Not all simulations teach in this way, however. Some provide extensive help or tutorial components that learners may access at any time and according to their own choice. A simulation about road signs and driving laws might introduce the signs and rules, guide the learner in their use, and provide practice by letting the learner use the simulation over and over until familiar with the laws. Many simulations have this characteristic. If used once, the simulation presents information and guides the learner in its acquisition. If used repeatedly, it takes on the characteristics of a drill. Some simulations *are* in fact drills, requiring the learner to continue until proficiency is demonstrated.

Many simulations combine instructional strategies. *Microsoft Flight Simulator,* for example, allows the user to fly the airplane alone or with a simulated instructor. With the instructor feature turned on, presentation and guidance is provided. Without it, practice is emphasized. Laboratory simulations, such as *Burette,* generally assume the learner has had some introduction to laboratory procedures and provide guidance and practice, usually before performing the real experiment.

Finally, simulations may be used as tests. Flying a simulated airplane may be the test that determines if the learner is ready to fly real planes. If the learner "crashes" in a flight simulation, more practice is probably needed in the simulated environment. If the simulated flight is successful, the learner is perhaps ready to fly in a real airplane—with an instructor. Another benefit of using simulations as tests is that they have greater face validity than the alternatives. For example, a paper-and-pencil test with multiple-choice items is unlikely to provide the same information about a learner's ability to perform a chemistry experiment as requiring the learner to perform the experiment via a simulation. This is a controversial area however. Psychometricians argue that because a simulation is different for each user, simulation-based tests may suffer from lower reliability, which threatens their validity as well. Despite these difficulties, simulation-based exams are being used in a variety of fields to assess professional skills, such as aviation, medicine, and dentistry (Alessi & Johnson, 1992; O'Neil, Allred, & Dennis, 1997; O'Neil, Chung, & Brown, 1997).

The other aspect of flexibility and the final advantage of simulations is the applicability of simulation to different educational philosophies. Simulation is one of the few methodologies embraced equally by behavioral versus cognitive psychologists, and by instructivist versus constructivist educators. Simulations can be designed in accordance with any of these approaches. We can design simulations that emphasize behavioral objectives; thinking and problem solving; direct instruction; discovery or experiential learning; or individualized, collaborative, or competitive learning. We can also combine these techniques in the same program. A simulation may provide extensive learner control or may be directive and program controlled. Simulations provide designers with far more options and decisions than other methodologies, as we shall see in the next section on factors important to instructional simulations.

Factors in Simulations

Introduction—A Theory of Learning from Simulation

We begin our discussion of factors in simulation by briefly describing a basic theory of learning from simulation. That theory (Alessi, 2000b), which is diagrammed in Figure 7.14, maintains that three considerations combine to determine how learners encode,

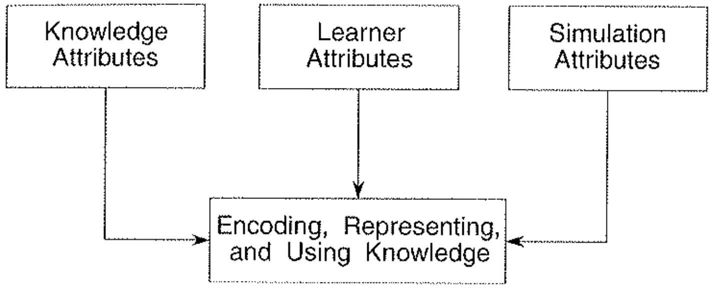


FIGURE 7.14
Knowledge, Learner, and Simulation Attributes Combine to Affect Learning

represent, and use knowledge: the attributes of the knowledge to be learned, the attributes of the learners, and the attributes of the simulation environment.

Knowledge attributes include the type of knowledge in a simulation, its organization, its complexity, and its precision. The *type* of knowledge primarily means whether it is declarative or procedural (Anderson, 1980), corresponding to the two broad categories of simulation described earlier, simulations *about* something versus simulations on *how to do* something. The *organization* of the knowledge refers to whether it is sequential, hierarchical, weblike, cyclical, or perhaps something else. *Complexity* is independent of organization. The knowledge needed for diagnosing and fixing problems in a bicycle and a diesel engine may have the same organization (some parts hierarchical and some sequential), but the knowledge relevant to the diesel engine is clearly much more complex, because the engine has more parts, more potential problems, more diagnostic procedures, and more ways to fix it. *Precision* refers to the accuracy of knowledge and how well predictions can be made from it. Some natural phenomena are well understood, and events can be predicted and effects controlled precisely, based on our knowledge of them. The mechanics of a bicycle are precise. Other phenomena are poorly understood, so prediction and control are more difficult. Our knowledge of weather is not precise, and our knowledge of human psychology is imprecise at this time. Each of these attributes has implications for the manner and ease with which people learn knowledge, and thus affects how simulation attributes are designed.

Learner attributes (of the people expected to use the simulation) are many and include age, gender, prerequisite knowledge or skills, prior knowledge or ability in the subject area, general cognitive abilities such as memory and problem solving, metacognitive abilities, interest in the subject area, learning styles and preferences, and motivation to learn. Once again, all these affect the probability of successful learning, and simulations should be designed accordingly.

Attributes of the simulation are the realizations of the factors relevant to simulation design. That is, each choice a designer makes about a factor (for example, the *scenario* factor) results in a particular attribute for the simulation (for example, having the scenario of an industrial labor strike, which requires negotiation between representatives of labor and management). The designer has little control over the attributes of knowledge or the attributes of potential learners, so simulation design is largely a matter of selecting the simulation attributes in accordance with the uncontrollable attributes, and in such a way as to best facilitate learning.

All these attributes combine to affect learning and how the learners encode knowl-

ory (encoding, representing, and using knowledge) deals with the learning issues, discussed in Chapter 2. The two issues most relevant are *mental models* (Frederiksen, White, & Gutwill, 1999; Gentner & Stevens, 1983) and *transfer.* Any simulation incorporates a model of some phenomenon or procedure, and the primary objective is for the learner to internalize that model—to develop their own mental model. Therefore, how such internal models form and work is crucial to our understanding of learning from simulations. Second, one of the main reasons we use simulations is to enhance transfer of learning. An understanding of what facilitates transfer (both near and far) is also crucial to our understanding.

Fidelity

For the remainder of this section, we describe the simulation factors and how they should be related to the knowledge attributes, learner attributes, and desired learning outcomes. The factor of *fidelity* is an overarching issue that affects all aspects of a simulation, such as the underlying model, presentations, and interactions. Fidelity refers to how closely a simulation imitates reality. An aircraft simulator with a sophisticated visual and motion system provides a high-fidelity simulation of flying. A computer-based simulation, such as *Microsoft Flight Simulator* (Microsoft, 1989) has much lower fidelity. Fidelity affects both *initial learning* (the learner's performance during the simulation) and *transfer of learning* (how well one applies new knowledge or skills to new situations).

Historically, people believed that increasing fidelity in an instructional setting necessarily led to better transfer (Hays, 1980). However, research has demonstrated that the relationship between fidelity and transfer is more complex and depends on, among other things, the instructional level of the learner (e.g., Andrews, Carroll, & Bell, 1995; Burki-Cohen, Soja, & Longridge, 1998). Alessi (1988) suggested this complexity exists because transfer of learning depends not only on fidelity but also on *initial learning* (the level of learning at the time of instruction), which in turn is also affected by fidelity, as indicated by Figure 7.15.

The relationship between fidelity and initial learning is complex, as illustrated in Figure 7.16. For a novice learner, low-fidelity instruction does yield learning, but some increase in fidelity might result in better learning. For example, a student pilot would learn *something* from reading a text about flying an airplane, but might learn more from watching a film with narration. The same person might learn less, however, from a very high-fidelity experience, such as in a mechanical simulator. Putting the novice in a real airplane, the highest possible level of fidelity, may be so confusing and stressful as to result in no learning at all. On the other hand, an experienced user initially learns more from higher fidelity, such as in a mechanical simulator. In a real airplane, that person may learn less than in a simulator, but more than the novice. For an expert, such as an

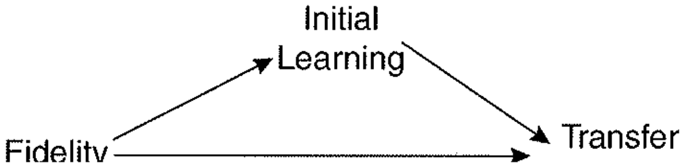


FIGURE 7.15
Transfer of Learning
Transfer is affected directly by fidelity and by initial learning, which is also

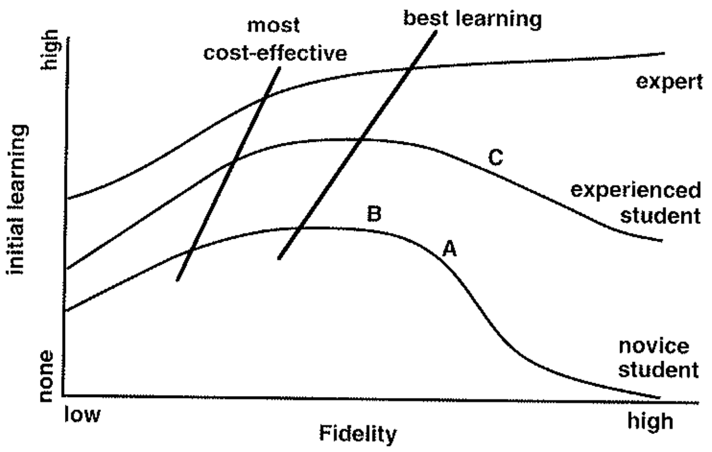


FIGURE 7.16

Hypothesized Relationship of Fidelity and Learning

By permission of the *Journal of Computer-Based Instruction* and the Association for the Development of Computer-Based Instructional Systems.

experienced pilot learning to fly a new airplane, a high-fidelity simulator might be very effective, and the actual airplane even more effective.

In Figure 7.16, the line labeled *best learning* is plotted through (or near) the maximum point of each curve, the point of best learning for different levels of learners. For increasingly sophisticated learners, it reflects increasingly high-fidelity instruction. However, the line does not represent the best level of fidelity in terms of other training factors. The line labeled *most cost-effective* intersects the curves where they begin to exhibit diminishing returns. Beyond that point, great increases in fidelity and expense are required for small increases in learning. An efficient curriculum would train students of increasing knowledge levels at correspondingly higher points on the *most cost-effective* line. Methods of measuring and deciding on the most cost-effective combinations of media and instructional fidelity are discussed in Roscoe (1971; 1972), Povenmire and Roscoe (1973), and Carter and Trollip (1980).

Figure 7.16 describes the effect of fidelity on learning at the time of instruction. As Figure 7.15 indicated, transfer of learning depends directly on fidelity and on initial knowledge obtained at the time of instruction. Initial learning, however, is not only affected by *actual* fidelity, but also by the *perceived* similarity of the instructional situation to the performance environment. This perception often affects the learner's level of motivation. These additional factors, which are not independent, are illustrated in Figure 7.17. Actual similarity (fidelity) affects perceived similarity, which affects motivation. For that reason, it is possible that point *A* on Figure 7.16 will yield less initial learning for the novice learner than *B*, but may still yield higher transfer. However, point *C*, which represents higher fidelity and higher initial learning, will probably yield better transfer for a more advanced learner than either points *A* or *B*.

It would appear we are faced with a dilemma in simulation design. Increasing fidelity, which theoretically should increase transfer, may inhibit initial learning, which in turn would inhibit transfer. On the other hand, decreasing fidelity may increase initial learning, but what is learned may not transfer to the application situation if too dissimilar.

One solution to this dilemma may lie in using a level of fidelity based on the learner's current instructional level. As a learner progresses, the appropriate level of fidelity increases. For a new learner, initial learning is emphasized (with lower fidelity), and

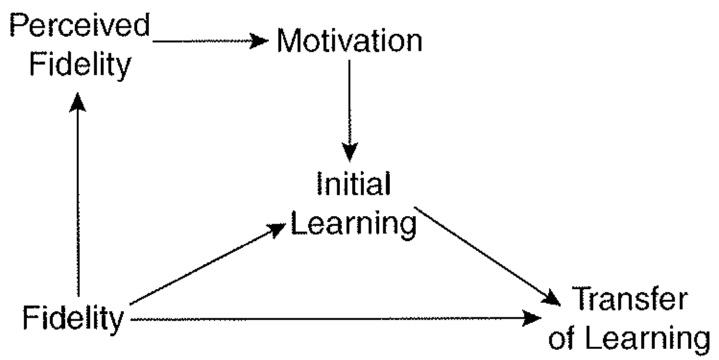


FIGURE 7.17

Transfer Is Also Affected Indirectly by Perceived Fidelity and Motivation

for an advanced learner, transfer is emphasized (with higher fidelity). This can be accomplished by choosing points along the *best learning* line in Figure 7.16 or, in consideration of economic and time limitations, along the *most cost-effective* line.

This solution has been suggested for instruction generally in Bruner's spiral curriculum (Bruner, 1966) and Reigeluth's elaboration model of instruction (Reigeluth, 1979). For simulations, this approach is called *dynamic fidelity.* It has been suggested under different names by other simulation researchers and developers (for example, de Jong et al., 1999; Goodyear, Njoo, Hijne, & van Berkum, 1991; Reigeluth & Schwartz, 1989; Swaak, van Joolingen, & de Jong, 1998). There are many ways to implement dynamic fidelity. It may be increased (or decreased) continuously based on the learner's performance. It may be increased from one lesson segment to the next in a more discrete fashion. It may be increased by the learner's own choice or the instructor's choice. Combinations and other approaches are also possible.

The discussion above may imply that fidelity is a single entity. In fact, simulation designers speak about fidelity *of* the presentations and fidelity *of* the model and fidelity *of* many other components of a simulation. A simulation may have realistic presentations but rather simple interactions. Similarly, it may have a fixed level of fidelity for the underlying model but dynamically increasing fidelity of presentations and interactions. After discussing the other simulation factors in the remainder of this section, we will return to the issue of fidelity and suggest a model for integrating the many decisions a designer must make about it.

Delivery Mode

One of the most visible of the factors in simulation is the delivery mode. Some of the many modes of delivering a simulation include:

- An individual microcomputer with no special peripherals
- Networked microcomputers with no special peripherals
- Individual microcomputer-based virtual reality environment with head-mounted viewer and sensory gloves
- Networked microcomputers with virtual reality equipment
- Supercomputers with virtual reality environment
- Large-scale physical simulators (such as for a jet aircraft) controlled by super-

The choice of one of the above modes *should* be made in appreciation of the instructional goals, the learner characteristics, the level of fidelity required, the importance of performance, learner motivation, and economics. In many cases, however, the delivery mode is chosen first. For example, a company already owns a large simulator and wishes to use it as much as possible. In that case, the choice of delivery mode impacts many decisions concerning presentations, scenario, feedback, and the like.

Delivery mode can have a huge effect on cost. A large-scale jet aircraft simulator costs tens of millions of dollars. The same is true for many military equipment simulators, power plant simulators, NASA spacecraft simulators, and medical simulation environments. What can justify such cost? Perhaps the most critical consideration is *importance of performance,* which can best be appreciated by assessing the *consequences of making errors.* Millions of dollars can be justified for the use of aircraft simulators because the consequences of pilot error in a commercial jet airliner can be catastrophic, affecting hundreds of passengers and very expensive equipment and property. The same is true for the operation of a nuclear power plant. The consequences of errors can be deadly and incredibly expensive, so training must be completed to a degree of mastery (near perfection) that justifies very expensive simulators.

Instructional Strategy

Related to delivery mode, and sometimes a function of it, is the simulation's instructional strategy. Some of the more common instructional strategies for simulation follow. These are not all independent. A particular lesson may combine different strategies, such as microworld and laboratory simulation, or virtual reality and operator-in-the-loop simulation.

Microworlds Many physical simulations are delivered as microworlds. The learner is given a collection of objects that can be assembled, manipulated, turned on and off, measured, and so on. Sometimes a microworld follows the rules of the real world, and sometimes it allows the user to ignore them. *Future Lab: Gravity for Physical Science* is a small microworld, in which some aspects are true to the laws of physics (earth's gravitational pull is fixed and cannot be manipulated), whereas other aspects may be overridden (for example, friction can be turned on and off).

The term *microworld* has been used for many types of software. Some, but not all such microworlds, are simulations. Microworlds that have an underlying model to be learned, such as *Future Lab: Gravity for Physical Science* or White's physics and chemistry microworlds (White, 1993), are simulations. In contrast, programming microworlds such as *Logo* are not (Papert, 1980). *SimCity Classic* and related programs (*SimFarm, SimEarth*) may be considered microworlds.

Scientific Discovery Learning These are usually iterative simulations, such as *Catlab* and *Population Dynamics,* in which learners perform scientific experiments to ascertain the laws of nature for themselves.

Virtual Reality Virtual reality environments (Milheim, 1995; Psotka, 1995) replicate a real-life environment visually, functionally (the actions you can perform), aurally, and sometimes kinesthetically (what you can feel). They may require special equipment, such as a head-mounted computer monitor and gloves that can sense hand movements and provide tactile feedback to the hands. Virtual reality is generally used for physical or procedural simulations. Newer microcomputer-based flight simulators may be considered low-end virtual reality programs. Many can be operated with realistic aircraft controls connected to the computer.

Laboratory Simulations These are usually physical or procedural simulations, such as *Burette* and *BioLab Frog,* that allow the learner to engage in procedures, observations, and computations of laboratory research. Often, using these is followed by doing real laboratory activity. Some laboratory simulations allow users to follow any paths they want (including completely nonsensical ones), but most have constraints, which guide the learner down correct pathways.

Role Playing This strategy is generally used in situational simulations, such as *Capitalism, The Interactive Courtroom,* and *Choices Choices—Taking Responsibility* (Tom Snyder Productions, 1997). Role playing is a popular approach for training in business, salesmanship, counseling, parenting, and classroom teaching (Holsbrink-Engels, 1997).

Operator-in-the-Loop This is a term used by engineers for large-scale physical simulators of aircraft, military equipment (such as tanks, submarines, and radar systems), automobiles, and power plants. *Operator-in-the-loop* refers to the fact that a physical device (such as an aircraft simulator) is running while a live operator (the learner) interacts with it in real time.

Case-Based Scenarios Sometimes called *goal-based scenarios* (Collins, 1994), case-based scenarios (Schank & Cleary, 1995) refer to an approach closely related to and sometimes incorporating role playing. The learner is placed in a hypothetical work or problem situation and must carry out the job or find and implement solutions. These are popular in business. For example, training accountants may have the learner playing the role of auditor who must audit a small business. The auditor must obtain and inspect the company books, interview people, perform audit tests, and write an audit report.

Simulation Gaming This strategy is sometimes just called *gaming.* Often gaming is competitive, as in business games like *Capitalism* (Interactive Magic, 1996). This is especially true for multiple-user simulation games. Other simulation games are less competitive, such as *MayaQuest* (MECC, 1995b) and *Africa Trail* (MECC, 1995a).

The Underlying Model and Its Components

The underlying model of the simulation is the *representation* of the system or phenomenon being simulated. The computer program depicts the physical entity, the procedure, the situation in which the learner is a part, or the process that the program mimics. Computer models that underly simulations are primarily of three types: *continuous, discrete,* and *logical.*

Continuous simulation models are those that represent phenomena having an infinite number of states. Most phenomena in the physical sciences and many in the social sciences are of this sort. The model underlying the motion of a falling object, the growth of animal populations, and the cycles of an economy all are based on continuous simulation methods.

The mathematics used to represent such systems is calculus, and the solutions that enable one to program the model are based on numerical integration. The program for a continuous simulation includes the initial conditions for relevant variables, their rates of change, and the time period over which they are examined.

If you are building a program to simulate plant growth, for example, variables might include sunlight intensity, air temperature, water and mineral availability, chlorophyll content of the plant, and plant tissue mass. The initial values of each of these would be necessary along with formulae indicating how each changes over time, as well as the values of other variables and constants. Given this information, a continuous simulation calculates the change in these variables for each time increment (for example, every hour).

The mathematics underlying this type of simulation is complex, but software exists to aid the designer in development. For Macintosh and Windows computers, *STELLA* (High Performance Systems, 1987) and *PowerSim* (PowerSim, 1999) allow the developer to enter initial conditions, rates of change, and time parameters; and the software generates the necessary equations.

Discrete simulations are less common in education. They represent phenomena in which quantities vary by discrete amounts. Whereas the variables of continuous simulations are real numbers, those of discrete simulations are integers. Common examples of discrete simulations are queuing simulations, which represent objects waiting in line. Systems such as automobile and air traffic, check-out lines in a grocery store, and production on an assembly line are examples amenable to discrete simulation.

The mathematics of discrete simulations is probability, statistics, and queuing theory. Discrete phenomena are characterized by objects arriving for some kind of service (cars, airplanes, or people), waiting in line for service, being served, and finally leaving the system. The simulation depends on knowing the distribution representing arrival of objects, and the patterns and time required to serve them. Although discrete simulations are easier to program than continuous simulations, their development is also facilitated by simulation systems and languages. Both *STELLA* and *PowerSim*, for example, include commands for creating discrete simulation models.

Logical simulations are very common among educational simulations, though uncommon in other uses of simulation. Logical models are represented by sets of *if–then* rules in a computer program. Systems represented by logical models include the operation of machines, decisions in running a business, and many social interactions. For example, a machine starts to operate *if* the power switch is depressed. A camera takes a picture *if* there is film, and it is properly advanced, and you press a button. Sales take place *if* it is a work day, there is inventory to sell, and customers are willing to pay the required price.

Continuous and discrete simulation methods have long been used by scientists and engineers for research and development. They often used these simulations to understand how various physical phenomena work and to design systems based on them. Logical simulation methods are of greater interest to educators than scientists and engineers. However, many simulations, especially educational ones, use a combination of these methods. Most simulations in science education, for example, include a combination of

the physical system itself, and the logical parts of the model represent the ways in which users interact with the physical system. This is also true in most procedural simulations. In a program such as *Microsoft Flight Simulator*, the model of aircraft lift and movement is a continuous one based on principles of fluid dynamics. On the other hand, all the pilot actions, such as turning the yoke or setting the navigation instruments, are represented by logical models.

The underlying model must include a number of components that determine both the nature of the simulation and the nature of the learner's interactions with it. These components are

- Objects
- Precision
- Type of reality
- Sequence
- Number of solutions
- Time frame
- Role of the learner

Objects The objects of the simulation are any physical entities, pictured or described. The objects in *SimCity Classic* (Figure 7.1) include the buildings, streets, parks, cars, and funds available. The objects in *Future Lab: Circuits for Physical Science* (Figure 7.3) include the meters, switches, bulbs, wires, as well as the electrical quantities voltage and amperage. Other examples include airplanes, parts of a chemical apparatus, telephones, spaceships, hospital patients, automobile engines, unknown substances, a job application, road signs, school principals, animals, corporations, countries, and so on. Some simulations may deal with a single object, such as a piano, whereas others may deal with many objects, such as the teachers and students in an elementary school.

Having a larger number of objects does not necessarily make the simulation more complicated, either to program or to use. Rather, it is usually the presence of *people* among the objects that increases complexity. The rules governing the behavior of people are far less understood than those governing the behavior of airplanes, pianos, and animals.

Precision Precision refers to how well we understand the processes being simulated. The precision of the real phenomenon is closely correlated with the presence or absence of people as objects. The most precise subjects are those involving strict mathematical, physical, or chemical laws. It is well known what happens when a distillation apparatus is heated or when a 5-kilogram weight is dropped from three meters.

However, even phenomena that follow physical or chemical laws may have elements of probability or chance. That is, some of the factors that influence reality are either unknown or impossible to determine. An automobile engine, for example, follows physical and chemical laws completely, but deciding why an engine runs poorly is still a difficult matter because so many physical and chemical influences can affect the many parts of the engine. Many unknown influences, such as the care the engine has received in the past or how fast the owner normally drives, also affect the engine. Thus, the operation of an engine is based on chance or probabilistic considerations, as well as scientific or mechanical ones. The more chance is involved, the less precise the model is, and the harder it is

The extreme case is when people are involved. Very little is really understood about individual human behavior, which makes predicting it almost impossible. Simulations that include humans as objects, therefore, incorporate a great deal of chance and consequently are the least precise and most difficult to program. This is the case for almost all situational simulations.

Remember that simulations require a description or prediction of the behavior of the various objects. When trying to determine how difficult a simulation will be to program, think about how *predictable* the various objects are. Pianos, for example, are very predictable: pressing a particular key always results in the same sound. Automobile engines are less predictable. When you turn the ignition key on a cold morning, they do not always start. People are very unpredictable, and the degree of predictability varies from one individual to another. Thus, simulating a piano is easier than simulating an automobile engine, which in turn is easier than simulating a person.

Type of Reality The type of reality of a simulation refers to whether the phenomenon depicted is one that occurs in the real world. There are three levels of reality. Some phenomena *do* occur as simulated, which includes most simulations described so far. Some phenomena do occur, but *not exactly* as simulated, such as the learner taking the role of an animal in a predator–prey simulation, or turning off friction in a physics laboratory, or doubling the birth rate of a country in one year. Some phenomena are *imaginary,* which do not occur at all, such as castles with dragons or battles between spaceships. Realistic subjects are neither better nor worse than imaginary ones. They simply have different purposes and advantages.

Sequence Sequence refers to whether the events occur in a linear, cyclic, or more complex fashion. The events of a titration are essentially linear. Basically a titration should be performed in one way. The events of driving and obeying road signs are cyclic. We periodically approach a road sign and engage in the appropriate behavior, such as slowing, stopping, and looking. The same scenario occurs repeatedly as we drive.

Many phenomena are complex, which means that the order of events is not strictly definable or that many different orders may be possible, some perhaps preferable to others. There are many ways to fill out a job application, land an airplane, diagnose problems in an automobile engine, or run a business. Many unpredictable events may occur in one's first year of teaching or in treating a hospital patient. In general, the inclusion of unpredictable events makes the sequence of a simulation more complex. Although complexity is a function of reality, the underlying model of the simulation is usually simplified to make it easier to design and program, and to facilitate learning.

Number of Solutions Reality varies a great deal with respect to the number of solutions available. Sometimes there is no solution because there is no such thing as right or wrong. An example of this is how mating different cats (in *Catlab*) affects the characteristics of the offspring. Another is measuring the time it takes for objects to reach the ground when dropped from different heights.

Other subjects, particularly procedural ones, have one preferred sequence of events. Examples include performing a titration, playing a particular song on a piano, or properly obeying a series of road signs encountered on a road. Finally, some subjects have your first year of teaching, or running a business. In most simulations, the number of solutions possible in the real world is reduced in the simulation, both to simplify programming and to facilitate learning.

Time Frame The time frame is the period of time over which a phenomenon normally takes place. An event in optical physics, such as light moving through a lens, occurs in a billionth of a second. A titration may take from ten minutes to an hour. Diagnosing a rare disease takes days or weeks. Breeding and raising cats takes months. Doubling the population of a country takes decades. The formation of mountains and rivers takes a million years. Models can be built to simulate all these things, but the more extreme the time frame of the real phenomenon (microseconds or millennia), the less realistic the model can necessarily be on this dimension. Nevertheless, it is precisely those events that occur extremely fast or very slowly that simulations excel at teaching. Learners cannot in reality observe the motion of light through a lens or the growth of a mountain, but they can do so through simulation.

The internal model may deal with the time frame in three ways. It may eliminate time, modify time, or maintain real time. Elimination of time is accomplished in what are called *static* simulations (van Joolingen & de Jong, 1991). Most iterative simulations eliminate time. You set parameters, click the start button, and get results. There is no natural passage of time. When time does pass (modified or not) during a simulation, it is called a *dynamic* simulation (van Joolingen & de Jong, 1991). Modification of time is probably the most common, and occurs frequently in physical, procedural, and situational simulations.

Time may be speeded up or slowed down. This may be done consistently (for example, ten seconds in the simulation is always equivalent to one hour in the real world) or inconsistently (for example, sometimes ten seconds represents an hour and sometimes ten seconds represents one minute to accentuate a critical part of a procedure). Last, a model may maintain real time. This is not very common and applies mostly to procedural simulations for procedures that do not take very long, like performing a chemistry titration. An interesting example is *Microsoft Flight Simulator.* This program allows users to select whether they want to fly in real time or modified time and to change their mind whenever desired. This is very useful for a skill such as flying. During take-off and landings, which take just a few minutes, you generally want to do everything in real time (which is the program's default). But as you fly for several hours from Chicago to Saint Louis over the flat Illinois corn fields, most users like to speed up the trip.

Modification of time is a very important example of lowering a simulation's fidelity for instructional advantage. Most of what we want to learn does not occur in convenient time frames. Rather, they occur too quickly or too slowly. Modifying the time frame can decrease boredom, improve time efficiency, accentuate critical events, and clarify the big picture.

Role of the Learner The role of the learner refers to whether the person using the simulation is considered one of the objects in the model or is external to it. Being a part of the model does not necessarily mean the learner is a *person* in it; the part may be an animal or physical object. Usually, however, people are people. In most situational simulations and in simulations using game, role playing, or case-based scenario aplearner generally manipulates and observes objects from outside. Procedural simulations can go either way, depending on whether the designer wants to create a sense of involvement. Operator-in-the-loop simulations (such as flight simulators) generally create a high level of involvement (you are in the plane and your life depends on landing it), whereas diagnostic simulations (diagnosing a patient's illness or a car engine's failure) generally do not.

Another aspect of the learner's role is whether the learner is primarily an *actor* or a *reactor*. By *actor* we mean the learner engages in actions to which other objects react. In contrast, objects may be the primary actors to which the learner reacts. Sometimes neither take the primary role, but act and react in equal ways. In a titration simulation, the learner is the primary actor who controls the apparatus and the experiment. This is the case for most iterative simulations. In a driving simulation, however, the learner must react correctly when a particular road sign comes into view. The learner has no control over the signs, and thus is the reactor. The learner is also the reactor when filling out a job application.

The most challenging simulations to design are those in which both the learner and the model act and react because this makes the underlying model more complex. In a medical simulation, the patient shows symptoms, the physician performs tests, the test results come back. The physician prescribes a treatment, new symptoms begin to develop in the patient, and so on. Each change in symptoms causes a reaction in the physician (the user of the simulation), and each choice or decision by the physician, such as giving medication, causes reactions in the patient and other aspects of the simulation. Situational simulations usually include equal action and reaction by both the model and the user of the simulation. This is seen in *The Interactive Courtroom* (Practising Law Institute, 1999), in which your decision, as a defense attorney, to object to questions by the prosecuting attorney, elicits a reaction by the trial judge to either sustain or overrule your objection.

Providing Objectives

Like all instructional software, simulations generally have an introductory section that includes a title page, description of the objectives, directions, and so on. Because learners are generally less familiar with simulation methodology than they are with tutorials, drills, and tests, greater emphasis should be placed on explaining the *purpose* of a simulation. We often observe that when using simulations, learners ask, "What am I supposed to be getting out of this." Without giving away any surprises, it is useful for motivational reasons to both clarify a simulation's educational purpose and to give some idea of the activity to come. To inform learners that a lesson is about the Civil War is not likely to excite them. For many it may conjure up memories of history books filled with dry facts about the civil war and dates to be remembered. If the lesson introduction states, in contrast, that "You will play the role of advisor to General Grant. You will help make decisions about purchasing weapons, food, medical supplies, and about strategy, which will affect the outcome of the war," the learner is likely to be more interested.

One needs to exercise some caution, however, because an introduction like the one above is so different from most learners' educational experiences that they might still wonder what the lesson is all about. Consequently, a lesson should not only state what will *happen* in the simulation, but should also make clear the *purpose* of the activity. The

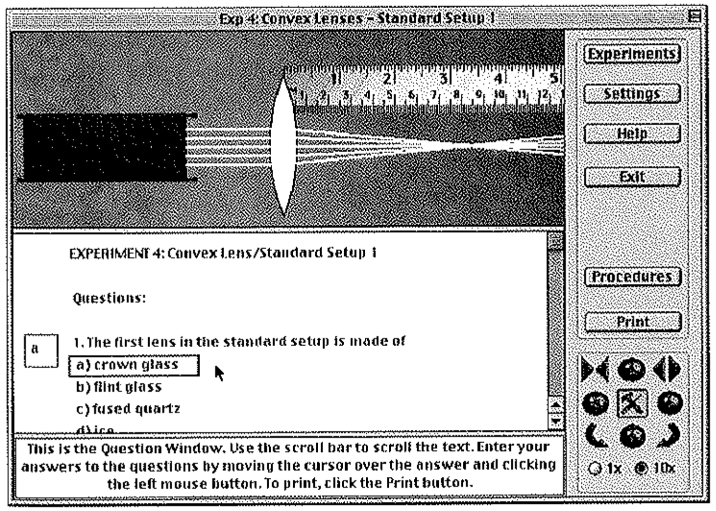
above introduction might continue by explaining, "This simulation will acquaint you with the social, political, and economic conditions of the middle nineteenth century, and how they influenced the outcome of the civil war." In their theory of motivation, Keller and Suzuki (1988) indicate that *relevance* of the lesson to the learner's needs should be clear. More than in other methodologies, providing goals or objectives for a simulation can make the relevance clear.

Directions

Another critical part of the introductory section of a simulation is the directions. We discussed directions in Chapter 3, but we must emphasize here that clear and complete directions are more important in simulations than in most methodologies because learners engage in activities that are more complex and varied. For example, there is greater use of devices other than the keyset and the mouse for interacting and inputting information. When devices such as a joystick or aircraft yoke are used, their operation must be explained and perhaps practice provided to master their use. If using the device is crucial to the successful operation of the simulation, you may even want to ensure the skill level by requiring proficiency before the learner starts the main simulation. Because of the complexity of directions in simulations, such as explaining how to operate devices on the display, it is usually better to give directions *when they become relevant,* rather than at the beginning of the program. Figure 7.18 shows directions (at the bottom of the illustration) explaining how to manipulate and respond to the interaction currently on the screen.

The Opening

After the title page, objectives, and directions, a simulation sometimes establishes the *scenario* for the lesson. This is often accomplished with what has been called an *opening scene.* This generally describes the context of the simulation, paying particular attention



EXPERIMENT 4: Convex Lens/Standard Setup 1

Questions:

a

1. The first lens in the standard setup is made of

a) crown glass

b) flint glass

c) fused quartz

d) ice

This is the Question Window. Use the scroll bar to scroll the text. Enter your answers to the questions by moving the cursor over the answer and clicking the left mouse button. To print, click the Print button.

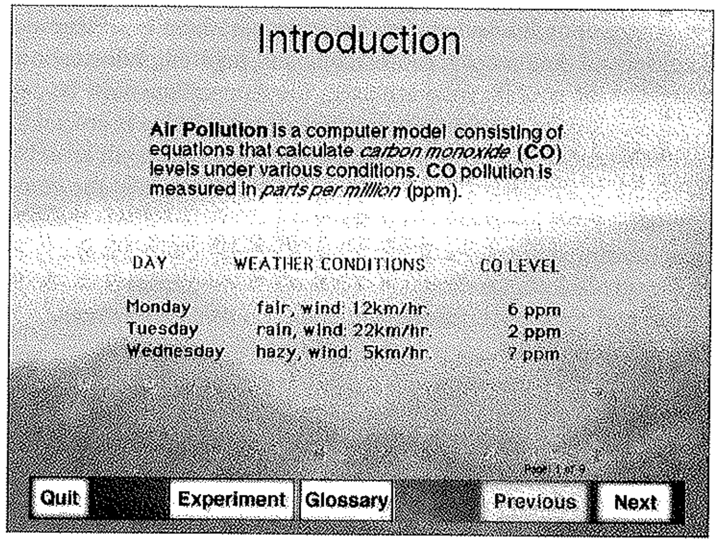
FIGURE 7.18

Directions in *Future Lab: Optics for Physical Science*

Courtesy of Simulations Plus.

to the physical entities the learner can manipulate, as well as the procedures the learner can engage in, the situations that the learner may encounter, or the processes to be studied. In *Air Pollution* (Chandler, 1995) the opening is textual (Figure 7.19) whereas in *MayaQuest* the opening is graphic with a woman in the top-left speaking to you and explaining your task (Figure 7.20).

The opening merely sets the stage. It does not attempt to describe all that the simulation can do. The narrator in *MayaQuest* describes your overall goal ("Save the earth from an incoming asteroid") and your initial goal ("Find some ancient burial site"), but many tasks are necessary to find the burial site, and additional tasks must be completed to reach the overall goal.



Introduction

Air Pollution is a computer model consisting of equations that calculate *carbon monoxide* (**CO**) levels under various conditions. **CO** pollution is measured in *parts per million* (ppm).

DAY	WEATHER CONDITIONS	CO LEVEL
Monday	fair, wind: 12km/hr.	6 ppm
Tuesday	rain, wind: 22km/hr.	2 ppm
Wednesday	hazy, wind: 5km/hr.	7 ppm

FIGURE 7.19

A Text Opening in *Air Pollution*

Courtesy of EME Corporation, Stuart, Florida.

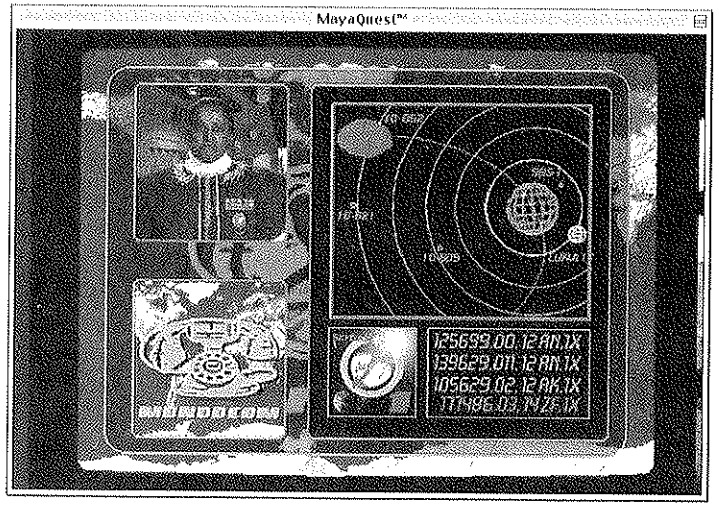


FIGURE 7.20

Graphic and Audio Opening in *MayaQuest*

Courtesy of MECC.

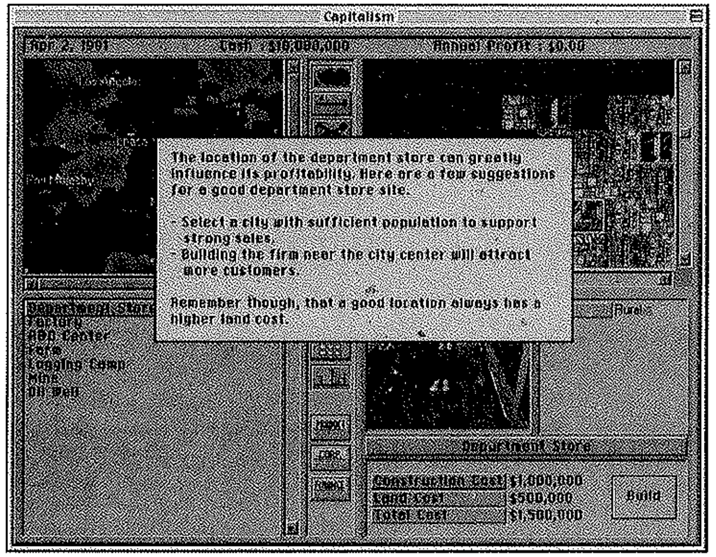
Instructional Supports

In addition to modeling activities and phenomena, instructional simulations almost always provide support for successful learning. Hints, corrective feedback, coaching, and providing assignments (de Jong et al., 1999) are the most common of instructional supports. Another is augmentation of reality (de Jong & van Joolingen, 1998), which refers to instructional support that emphasizes, clarifies, or makes the invisible visible. Examples of augmentation of reality are displaying the potential and kinetic energy of moving objects or highlighting components of a machine with color. An instructional support technique of fairly recent vintage is hyperlinks. Being able to click on a screen object to receive text or audio identifying its name and explaining its function has become a popular and effective type of support. Even more popular is the rollover technique, in which moving the cursor over an object gives its name, purpose, or directions for use.

The most common type of instructional support is text explanations embedded within the simulation. Figure 7.21 shows *Capitalism* (Interactive Magic, 1996) with a box in the center of the simulation giving advice about selecting a good site for building a department store. That type of instructional support usually decreases a simulation's fidelity. It is also an example of how lowering fidelity benefits initial learning.

Motivators

One of the advantages of simulation is motivation. Simulations tend to emphasize intrinsic motivation as recommended by Lepper and Chabay (1985). Learners find them interesting because they are participating in events rather than reading about them. But, as always, it is unwise to *assume* motivation is present. It is better to design for motivation enhancement and to use various motivational approaches to guarantee success with



The location of the department store can greatly influence its profitability. Here are a few suggestions for a good department store site.

- Select a city with sufficient population to support strong sales.
- Building the firm near the city center will attract more customers.

Remember though, that a good location always has a higher land cost.

Construction Cost	$1,000,000
Land Cost	$500,000
Total Cost	$1,500,000

FIGURE 7.21

Instructional Support in *Capitalism*

Courtesy of Trevor Chan and Enlight Software [www.enlight.com].

a wide variety of users. The approaches of Keller and Suzuki (1988) and Malone and Lepper (1987) are both useful.

Following are some examples. In a business simulation such as *Capitalism,* making money serves Keller's *satisfaction* component and is also a logical outcome of running a business. On the other hand, one must be wary of providing fake computer money as a reward for nonbusiness games, as it does not logically follow and may be perceived by learners as silly. Malone's *fantasy* component (Malone & Lepper, 1987) is a powerful technique that works better in simulations than in most other methodologies. Simulation games such as *SimCity Classic, Africa Trail,* and *MayaQuest* all use fantasy effectively. Sometimes the fantasy is a realistic (that is, not impossible) activity, even if not a likely one for the learner, such as pretending to be the president of the United States in *Oval Office—Challenge of the Presidency* (Meridian Creative Group, 1996).

Control is another component of Malone's theory of motivation, suggesting that learners are more motivated when they have a sense of being in control. This is not inherent in all simulations. Designers can and should design in such a way that learners feel in control. That does not mean they are able to control everything about a simulation. Providing a few good types of user control, for tasks users typically want to control, can be sufficient.

Last, Malone's *challenge* component suggests that learners must be *properly* challenged. The activities must not be either too easy or too difficult. This is a particularly relevant component for simulations and relates to the previously mentioned technique of dynamic fidelity. As learners progress through a simulation, it can gradually increase visual realism, add user actions, and *decrease* instructional supports (such as hints and coaching). Doing so increases the challenge to learners in keeping with their improving performance and may enhance transfer of learning as the simulation becomes more realistic.

Sequence

Earlier we discussed the sequence of the underlying model and of the real phenomenon. Here we refer to the sequence of the simulation as the learner encounters it, which is obviously related to both the sequence in the model and the real phenomena. Like the model's sequence, the overall simulation sequence may be linear, cyclic, or complex. Many procedural simulations are linear, such as performing a titration, whereas others, such as flying an airplane in a holding pattern, are cyclic. A cyclic sequence is inherent in almost all iterative simulations, such as the breeding of cats in *Catlab* (Kinnear, 1998).

Some procedural and most situational simulations have complex sequences with multiple paths and a variable number of steps. The paths and number of steps depend on the actions of the user, and so may change every time the user makes a decision or takes an action. Procedural simulations of the diagnosis variety and most situational simulations like *Capitalism* (Interactive Magic, 1996) illustrate complex sequence.

The sequence of the real phenomena is completely out of a designer's control, and sequence of the underlying model tends to be tied tightly to that of reality (although simplified). However, the simulation sequence as seen by the learner is very much up to the designer and greatly impacts the simulation's effectiveness. An overall sequence must be created that makes sense to learners, is easy to use, and is efficient.

Presentations

The factors of this section deal with how the simulation is presented to learners, what they see and hear, and how faithfully objects of the simulation are represented.

Mode Presentations may include text, pictures, voice, animations, video, or a combination. The best mode or modes are often dictated by the content. It is easier to depict road signs with pictures than describe them with words. Conversely, it is easier to describe the first year of teaching with a narrative than depict it with a drawing. Depicting a visual subject pictorially is usually more realistic, but not necessarily better from a pedagogical point of view. Sometimes it is just not possible to use anything other than text. Although expensive and time consuming to develop, situational and procedural simulations may benefit from video and audio presentation. Pictures and text tend to work better for physical and iterative simulations. Presentation mode affects motivation (people tend to like video), fidelity, and ease of use. Finally, some presentation modes are better for particular types of learners. For example, speech presentation is better for young children who do not yet read well.

Types Four major types of presentation are usually present to varying degrees in every simulation:

- Choices to be made
- Objects to be manipulated
- Events to react to
- Systems to investigate

Choices to be made may be either textual or pictorial, depending on the nature of the choice. Figure 7.22 from *Amazon Trail* shows a page on which the learner chooses what to do next. Figure 7.23, also from *Amazon Trail,* shows the page on which the user

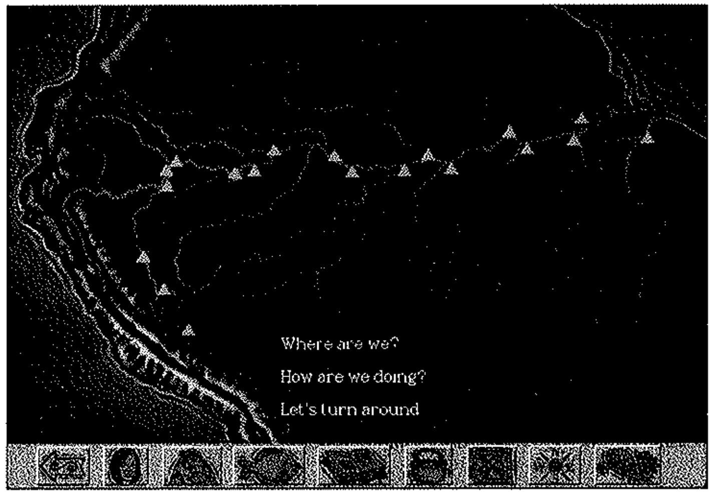


FIGURE 7.22

Text Choices in *Amazon Trail*

Courtesy of Softkey Multimedia.

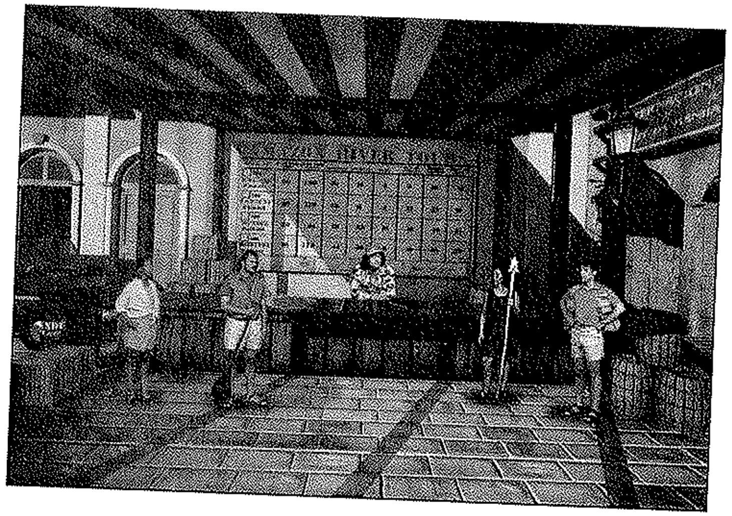
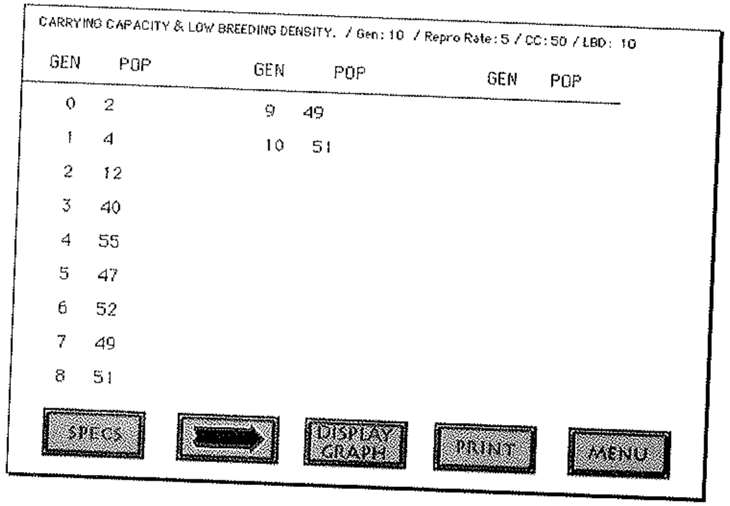


FIGURE 7.23

Graphic Choices in *Amazon Trail*

Courtesy of Softkey Multimedia.

chooses a guide by clicking on a person. Objects to be manipulated are usually pictorial, as in Figure 7.18 (shown previously) from *Future Lab: Optics for Physical Science* (Simulations Plus, 1998c). Events to which the learner must react can be of any mode. Thus, the learner may be told that a patient's vital signs have deteriorated, a pilot may see a change in the instruments, or a musician may hear a sequence of musical notes played by the computer. Systems to be investigated also typically use mixed modes. Figures 7.24 and 7.25 from *Population Concepts* (Lopez, 1994) describe the animals numerically and graphically whereas *Catlab* (Kinnear, 1998) draws pictures of each new generation of cats on the screen (Figure 7.4, shown previously).



CARRYING CAPACITY & LOW BREEDING DENSITY. / Gen: 10 / Repro Rate: 5 / CC: 50 / LBD: 10

GEN	POP	GEN	POP	GEN	POP
0	2	9	49		
1	4	10	51		
2	12				
3	40				
4	55				
5	47				
6	52				
7	49				
8	51				

FIGURE 7.24

Textual/Numeric Information in *Population Concepts*

Courtesy of EME Corporation, Stuart, Florida.

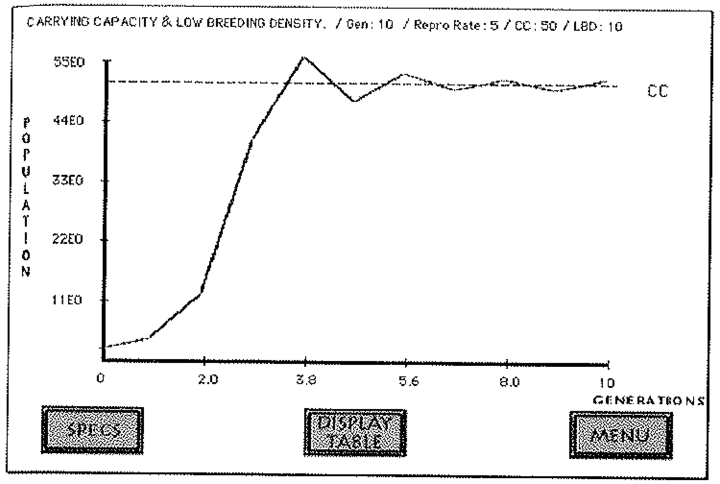


FIGURE 7.25

Graphic Information in *Population Concepts*

Courtesy of EME Corporation, Stuart, Florida.

Presentation Realism All simulations simplify reality. The result is a decrease in fidelity. It is important to remember that such a decrease in fidelity does not mean that the *effectiveness* of the simulation is decreased. To the contrary, it is usually beneficial to simplify.

The realism of presentations refers to the degree to which a particular component appears like its real counterpart (such as the fairly realistic depiction of the Amazon River as it appears from inside a boat, shown in Figure 7.26). To repeat, increased realism is *not* necessarily tied to increased effectiveness. In a program about weather, it may be important to represent a cumulonimbus cloud quite accurately, showing its anvil shape and giving details about growth rate, height, and intensity. However, in a program about cross-country flying, it may only be necessary to show clouds in the vicinity of the airplane. In a chemistry laboratory simulation, if the purpose is to introduce chemical apparatus, it may be important to include details such as volume markings, stoppers, and

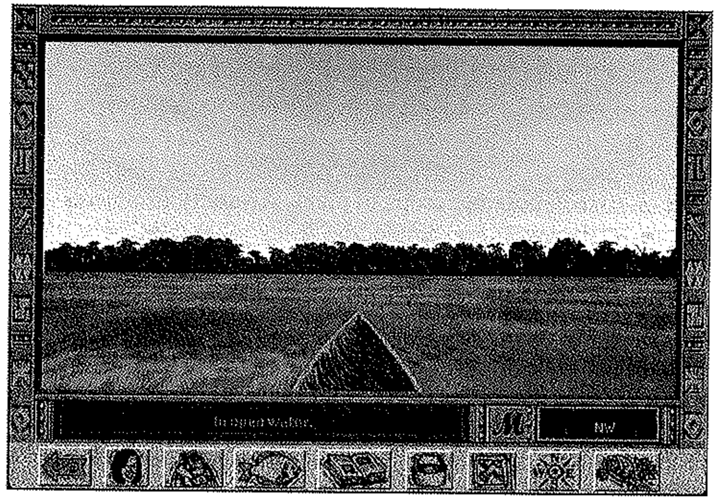


FIGURE 7.26

A Realistic Image in *Amazon Trail*

Courtesy of Softkey Multimedia.

the correct size of each object. However, if the simulation merely uses the apparatus as part of some experiment, simple silhouettes might suffice.

The *most* common error made in the design of simulations is believing that increased realism leads to improved learning. Particularly among novice designers, achieving high fidelity is almost a compulsion. The level of realism should be determined by instructional effectiveness.

Learner Actions

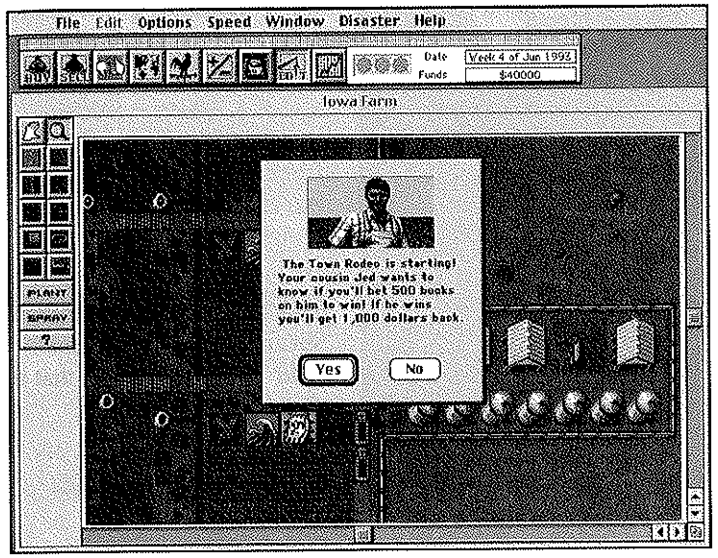
Simulations are very interactive in nature. That is what makes them both appealing and effective. Because simulations require a great deal of action on the part of learners, it is important to analyze learner actions carefully during simulation design.

Mode of Actions Simulations incorporate more variety of input modes than other methodologies. In addition to the keyset and mouse, they use devices such as steering yokes, joysticks, and microphones for speech input. Some of these are particularly useful in simulations that require the learner to manipulate objects on the screen. Designing input for disabled learners requires additional considerations to avoid creating obstacles for them. The best way to facilitate activity by all learners is to provide a variety of redundant methods for learner action.

The keyset is currently the best for inputting textual information, but as always, the age and typing ability of learners is a major design consideration. The mouse is good for selecting, drawing, and moving objects. Numeric inputs can be typed with the keyset, which is more accurate, or entered via numeric sliders using the mouse, which is easier and simultaneously shows allowable number ranges. If you use any devices other than the keyboard or mouse, be sure to determine whether they are available on the computers used by your potential learners. Finally, the use of *several* modes within a lesson enhances interest and stimulates more learning than the use of a single mode (Rigney & Lutz, 1976), as well as being advantageous for a wider variety of learners.

Types of Actions Earlier, four types of presentations were introduced and discussed: choices to make, objects to manipulate, events to react to, and systems to investigate. Each has its own associated user action. They are, respectively, making a choice, manipulating an object, reacting to an event, and collecting information. Figure 7.27 and 7.28 show several of these in the program *SimFarm* (Electronic Arts, 1996b). Figure 7.27 shows an event to be reacted to. Figure 7.28 shows the user, having clicked on the *plant* button, selecting from a list of sixteen possible crops to plant. A variety of objects also can be manipulated on the same display. For example, the *fence* tool (four tools above the *plant* button, on the left) allows you to build and change the farm's fence structure. The *bulldozer* tool (directly above the *plant* button on the right) allows you to remove buildings, trees, and other objects.

Increasingly, most of these actions are made with the mouse, which can be used to select among multiple-choice options, drag sliders, click on buttons, or arrange objects on the screen. In newer simulations, the keyboard is used primarily when words or sentences must be typed, which is less common in simulations than in other methodologies. Typing words and sentences is likely to be replaced with voice input in the near future.



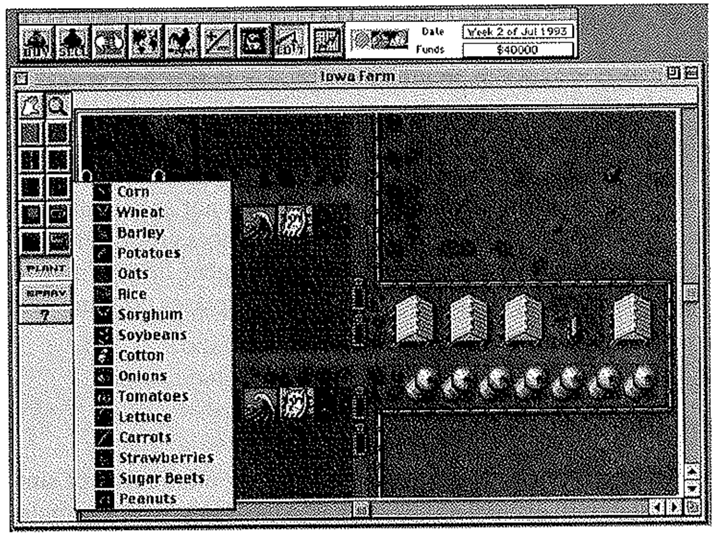
Date Week 4 of Jun 1993
Funds $40000

The Town Rodeo is starting! Your cousin Jed wants to know if you'll bet 500 bucks on him to win! If he wins you'll get 1,000 dollars back.

FIGURE 7.27

Reacting to an Event in *SimFarm*

Courtesy of Electronic Arts, Inc. Copyright © 1996 by Electronic Arts, Inc. SimFarm is a trademark or registered trademark of Electronic Arts, Inc. in the U.S. and/or other countries. All rights reserved.



Date Week 2 of Jul 1993
Funds $40000

FIGURE 7.28

Making a Choice A list of choices to select from is given in *SimFarm*. The tool bar also shows the fence tool and the bulldozer tool.

Courtesy of Electronic Arts, Inc. Copyright © 1996 by Electronic Arts, Inc. SimFarm is a trademark or registered trademark of Electronic Arts, Inc. in the U.S. and/or other countries. All rights reserved.

Realism of Actions Learner actions, like presentations, have varying degrees of realism. This means that the similarity between the learner's action and the action of a person in the real situation varies. Figure 7.29 from *Amazon Trail* (Softkey Multimedia, 1996) shows the user fishing for food by throwing spears into the water at the silhouettes of fish. The visual part of the activity is fairly realistic. However, the user throws the spear by clicking with the mouse, which is very different from the aim and coordination needed to hit fish with a spear in reality.

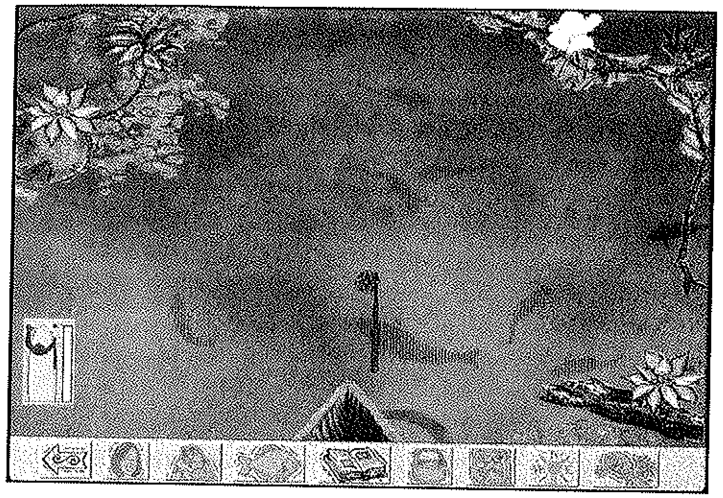


FIGURE 7.29

Spearing Fish in *Amazon Trail* A learner action that is visually realistic but not functionally realistic.

Courtesy of Softkey Multimedia.

In the context of user actions, fidelity may refer either to the *mode* of action (type, click, speech) or the *type* of action (select, generate, control). Typing in the desired rotation speed of a motor is of lower fidelity than dialing a simulated knob to achieve the same end. Multiple-choice questions are not common in the real world as a means for diagnosing why an automobile does not start in the morning. Touching the buttons of a simulated telephone is similar to the actual activity. Our real interactions with other people are usually predicated on speaking. Furthermore, the sentences in a conversation are generated rather than selected from a list. Because speech input is not sufficiently advanced, low fidelity typing or even lower fidelity selection of sentences are still the predominant techniques used for actions in situational simulations. Higher fidelity may be beneficial for motivation and for transfer of learning, but it also increases cost and may inhibit initial learning.

Learner Control of the Simulation

The amount of control a learner has in a simulation depends largely on the type of simulation. More control exists in iterative simulations, somewhat less in physical simulations, still less in procedural simulations, and the least in situational simulations. Types of learner control in a simulation include:

- Initial choices
- Sequence
- Obtaining directions
- Terminating the simulation
- Restarting within the simulation
- Restarting after termination
- Saving data

- Printing
- Changing the level of difficulty
- Changing the level of fidelity

Initial choices and returning to remake initial choices are essential in iterative simulations. They are the vehicle for determining process parameters and for rerunning the simulation with different parameters each time. Simulations with complex sequences, such as diagnosis simulations, often provide the user with control over sequence. Termination of a simulation (that is, the entire program), like any program, should be possible at any time. Restarting within the simulation means to begin the simulation sequence over, usually initializing all variables. Such an option is useful when the user takes an action that causes a failure, such as crashing an airplane into a mountain. Restarting after termination, in contrast, means choosing to use the simulation again after terminating the program. Simulations cannot always be restarted in the same way as tutorials and drills. In those methodologies, a marker may indicate where the user will return and continue working. Because of the holistic nature of simulations, the user must sometimes start again from the beginning. Procedural or situational simulations that are very long are more likely to incorporate markers for restarting than are physical or iterative simulations. In the latter types, users typically have considerable control over the simulation anyway, so restarting capabilities are not as critical.

The above are the most common and essential of learner controls in simulation. More optional are saving data, printing screens or data, changing the level of difficulty, and changing the level of fidelity. Saving and printing are conveniences. Designers should take advantage of them because they are easy to implement and because users like them (especially printing). As a result, they increase the *perception* of control, which, as stated earlier, improves motivation. Allowing users to change the level of difficulty is a potentially good way to maintain a proper level of challenge, which Malone and Lepper (1987) suggest is one method for increasing motivation. Similarly, allowing users to change the level of fidelity is a good way to implement dynamic fidelity, without requiring an accurate measure of the user's performance level.

Certain simulations have their own types of learner control. In simulation games, for example, players may choose names, tokens, teammates, and roles. In laboratory simulations, learners may choose which of several experiments to pursue. In adventure simulations like *Africa Trail,* learners have many choices, such as what country to visit next, what route to take, and what supplies to carry. Flight simulators allow you to choose among several types of aircraft, the airport you wish to depart, and your destination. A flight simulator might allow you to choose the weather (an example of decreasing both fidelity and difficulty) or might set it randomly (as happens in reality).

The distinction between learner actions and learner controls is often subtle, but taken together they are what simulations are all about. Designers should delineate and analyze all possible actions and controls, and then design them in consideration of the learner characteristics and learning goals. A key part of this is fidelity. For example, are the controls you provide the same as what we have available in the real world? We cannot control the weather, although some flight simulators permit that. Allowing low fidelity controls (and actions) can significantly enhance initial learning, though its effects on motivation and transfer may be the opposite.

System Reactions and Feedback

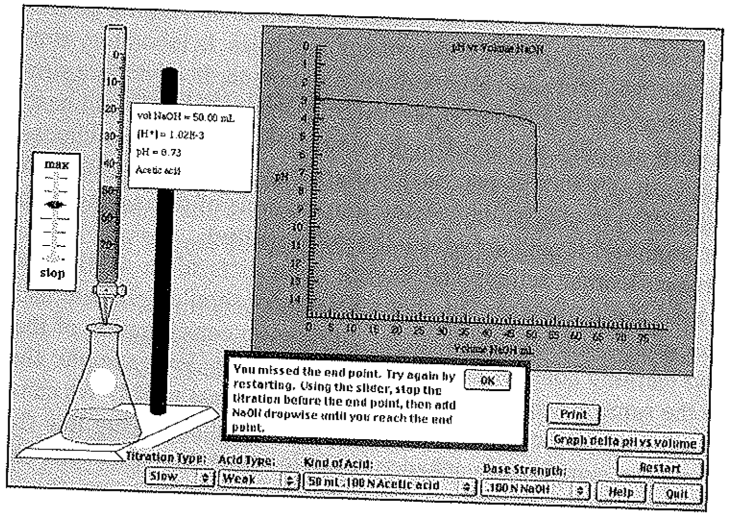
A simulation may react in many ways to a learner's action. If you add too much base during a titration, the liquid may change color, a voice may inform you that you added too much base, or corrective text may appear on the screen (Figure 7.30). If you are flying a simulated plane from Chicago to St. Louis and head northeast, a text message may appear, you might encounter unexpected weather, or nothing obvious may happen at all (unless you notice that you are over water rather than land). These examples point out the two main dimensions of system reactions and feedback. First, they may be natural (like the real world) or artificial. Second, they may be immediate or delayed.

Natural versus Artificial In the real world, if you fly an airplane into the clouds and become lost, you are given no verbal message to that effect nor informed about mountains in the vicinity. Natural feedback may come in the form of never reaching your destination or crashing into a mountain. In a simulation of such a flight, the same type of natural feedback can be provided with the simulated airplane crashing. However, artificial feedback can also be given in the form of a written or spoken message, such as the warning, "There are mountains up ahead" or "You have just crashed."

Imagine a simulation in which a mechanic is working on an engine that does not start. If the mechanic diagnoses the problem incorrectly and decides to replace the spark plugs instead of cleaning a blocked fuel line, which is the real cause of failure, natural feedback can be an engine that still does not start. Artificial feedback can be a message, such as, "The old spark plugs were fine. Try something else."

Natural feedback in a simulation is similar or identical to what occurs in reality. Artificial feedback may provide the same information, but in a way that does not occur naturally. Artificial feedback can also provide advance warning, which might not occur at all in the real world.

Figure 7.30 shows artificial feedback, a text message in the program *Burette* (EME, 1999). The message stops the learner from going any further with a failed experiment, explains what went wrong, and gives suggestions for how to succeed on the next attempt. Figure 7.31 shows an example of both natural and artificial feedback presented at the same time in *Future Lab: Gravity for Physical Science* (Simulations Plus, 1998b). Near the middle of the display you can see the 1-gram and 1-kilogram weights falling, a form of natural feedback. To their right you see two graphs showing the weights' change in velocity (differentially affected by air resistance) as a function of time. The latter is a form of artificial feedback, which the real world does not normally provide.



vol NaOH = 50.00 mL
[H⁺] = 1.02E-3
pH = 0.73
Acetic acid

You missed the end point. Try again by restarting. Using the slider, stop the titration before the end point, then add NaOH dropwise until you reach the end point.

Titration Type: Slow
Acid Type: Weak
Kind of Acid: 50 mL .100 N Acetic acid
Base Strength: .100 N NaOH

FIGURE 7.30

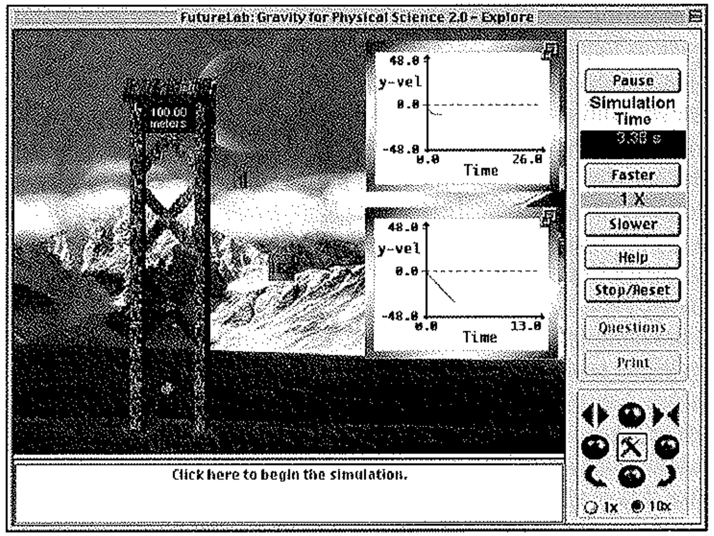
Textual Artificial Feedback in *Burette*

Courtesy of EME Corporation, Stuart, Florida.

Natural feedback is, by its nature, higher in fidelity. It usually indicates a problem without suggesting a solution. On the other hand, artificial feedback is lower in fidelity, tends to prevent errors, and facilitates immediate learning. Learners (especially beginners) often prefer artificial feedback, such as warnings to avoid a crash. Artificial feedback can be easier to interpret and be more positive in tone than natural feedback. These considerations must be weighed against the possibility that higher fidelity (in this case natural feedback) enhances motivation and transfer.

Immediate versus Delayed Feedback about any action may be given immediately or at some later stage. The examples discussed above also illustrate the alternatives of immediate versus delayed feedback. In the case of flying an airplane into the clouds, no feedback might be provided when first entering the clouds. Rather, the feedback is delayed until a catastrophe occurs. The alternative is to inform the learner the moment visibility is lost, thus preventing a crash. In the case of the mechanic fixing an engine, a message could be given as soon as the decision was made to change the spark plugs, rather than waiting for them to be replaced and attempting to start the engine.

Natural feedback in both these cases was delayed, even though the feedback occurred exactly at the time as it would have in reality. It is described as delayed because it occurred some time after the initial action that led to it, which is the way real-world



Click here to begin the simulation.

FIGURE 7.31

Natural and Artificial Feedback A combination of natural and artificial feedback is presented in *Future Lab: Gravity for Physical Science.*

Courtesy of Simulations Plus.

feedback often is. Thus, the learner may not discover the consequences of an action until a considerable time after the action. However, natural feedback is not always delayed. In the same flight in which the feedback is delayed about the decision to fly into the clouds, each movement of the joystick results in immediate changes in the flight instruments. Sometimes no feedback is given at all, which is often the case in reality. If you do not notice a stop sign while driving along a country road and drive through it, you may never know that you did so.

The issue of feedback timing (immediate or delayed) is one of the most interesting ones in simulation design because it poses a dilemma. Research has shown that immediate feedback is beneficial and preferred by learners. Natural feedback is presumed to enhance transfer of learning and is more fun and challenging. But natural feedback is usually delayed. The technique of dynamic fidelity again suggests a solution to the dilemma. Beginning learners can benefit from immediate feedback, even if unnatural. Advanced learners prefer and profit most from natural feedback, whether immediate or delayed.

Selecting Natural versus Artificial and Immediate versus Delayed Feedback

Regardless of how the real world works, simulation provides us the option of giving natural or artificial feedback, of giving immediate or delayed feedback, or of giving no feedback at all. The main reason for artificial feedback is to give more obvious and understandable feedback. A primary reason for giving immediate feedback, even if it is artificial, is to prevent errors and increase learning efficiency. The advantages of using natural feedback are that it has greater face validity, is usually more interesting, can be more challenging, and may enhance transfer of learning. Immediate verbal feedback about actions helps correct them before the learner becomes hopelessly lost and confused. Natural feedback is more like the real world, and sometimes better prepares the learner for performing in it.

In light of these considerations, we recommend using immediate corrective feedback, even it if is unnatural, when a learner first begins using a simulation or when the simulation's purpose is initial presentation and guidance. In contrast, when a more advanced learner uses the simulation, especially when it is used for practice or as a test, it should give natural feedback as much as possible, whether delayed or immediate. A good simulation can start the learner out with very helpful, immediate, and corrective feedback. As the learner progresses and improves performance, the amount of artificial feedback may be reduced and replaced with more natural feedback.

In general, learner actions fall into four major categories: desirable, neutral, negative, and critical. The feedback that is provided should be appropriate to the nature of the learner action and to the intention of the instruction. When a learner action is desirable (making progress toward a goal), immediate feedback is least necessary, and natural feedback is suitable in most cases. Neutral (or unnecessary) actions are those that have no effect on attaining a goal. For example, looking at your watch does not make your flight to London arrive any earlier. When such an action is taken in a simulation, immediate feedback is probably unnecessary. Negative actions cause the learner to move away from the goal or possibly prevent its attainment unless corrective action is taken. An example is deciding to fly an extra hundred miles when low on fuel. For beginning learners, future performance in a situation like this may be enhanced by using immediate feedback, even if it is artificial. Finally, an action may be critical, causing the goal to be permanently unreachable. Continuing to fly with the fuel gauge on empty is an example. In that circumstance, future performance probably is enhanced by giving the learner immediate artificial feedback, and showing the learner how to avoid the destructive consequences.

The examples above are meant only to illustrate the kinds of considerations involved in choosing feedback. They are not rules to be followed blindly. Artificial, natural, delayed, and immediate feedback can always be used, and may be used in various combinations. Each situation must be analyzed to determine which type of feedback best helps learners attain the goals of the simulation.

Completion of the Simulation

Completion of a simulation can mean many things. It may mean the learner has succeeded or failed in a particular run through the simulation. In iterative simulations this may simply mean the calculations and results are complete for a single set of choices. Individual runs of an iterative simulation are typically short and the learner repeats them many times. The learner may choose to begin again or not. In physical, procedural, and situational simulations, completion usually means the learner has followed either a successful path or one that has led to failure. In either case, it does not necessarily mean the learner terminates the simulation. The learner may choose (if the option is available) to do the simulation again immediately. If the learner does not choose to do so, the simulation is either temporarily or permanently terminated.

These distinctions may be unclear to learners. They might think "completing the simulation" is completing the entire program or lesson, whereas an iterative simulation may simply indicate a single run taking a few seconds. A distinction must be made between completing short simulation runs, as in iterative simulations, completing longer simulation sequences, such as flying from New York to London, and succeeding in the learning goals of an entire simulation program. As a designer, you must first be clear about these distinctions yourself, and then make them clear to your users.

A Taxonomy for Fidelity Analysis

This section began with a discussion of factors in simulation design, particularly fidelity, which is an overarching issue in simulation design. Fidelity is not a single factor for the entire simulation, but one that applies to many different components. Having discussed all the factors, we can now summarize the issue of fidelity in a more complete fashion.

Our summary is represented in Figure 7.32, which outlines a taxonomy for fidelity analysis. The rows correspond to the four types of simulations: physical, iterative, procedural, and situational. The columns represent four aspects of simulations to which fidelity is relevant. In the first column, which deals with the underlying model, fidelity considerations emphasize the *objects* inherent in the phenomenon and the *rules* underlying their behavior. In the second column, presentations, primary considerations are the visual and audible *stimuli* and the *time frame* in which events occur. In the third column, user actions, fidelity concerns the *number* and *type* of actions in which the learner may engage. In the fourth column, which deals with system feedback, considerations include whether there is *any* feedback, whether it is *immediate* or *delayed,* and whether it is *natural* or *artificial*. We now look at examples of fidelity analysis for each type of simulation.

FIGURE 7.32

Taxonomy of Simulation Fidelity Considerations

	UNDERLYING MODEL	PRESENTATIONS	USER ACTIONS	SYSTEM FEEDBACK
PHYSICAL	number of objects cause-effect relationships time frame	detail/realism of presentations visual versus textual presentations illusion of motion	user control versus natural progression of the phenomenon	mode of feedback immediacy of feedback whether there is any feedback at all exaggeration of feedback magnitude
ITERATIVE	number of variables in the math model accuracy of variables in the math model time increment for recalculation	what variables are: unknown known but not manipulated known and manipulated speeding or slowing the time frame	setting initial variables high level of user control between runs of the simulation	mode of feedback (text or pictorial) whether there is any feedback at all
PROCEDURAL	number of possible solution paths nature/complexity of solutions number of objects cause-effect relationships	mode of display (text, graphic, real) realism & completeness of images or descriptions	number of possible actions mode of actions (e.g., typing a word versus moving a joystick)	mode of feedback (text, pictorial, real) immediacy of feedback whether there is any feedback at all
SITUATIONAL	number of persons in the simulation probabilistic nature of human behavior behavior a function of multiple events level of precision of theory accuracy of theory chance events	mode of display (text, graphic, real) completeness of a scenario	number of possible actions flexibility of actions (e.g., multiple choice versus constructed response)	immediacy of feedback probabilistic feedback

By permission of the *Journal of Computer-Based Instruction* and the Association for the Development of Computer-Based Instructional Systems.

Fidelity in Physical Simulations In a physical simulation, such as teaching about gravitation and the orbits of satellites around the earth, a primary decision is what objects to include in the domain. For all learners, the satellite and the earth are necessary. At an advanced level of instruction and fidelity, the sun and moon can be included. Similarly, the mathematical equations governing the satellite and other bodies can be programmed with varying degrees of accuracy. Simplified equations can be used when the simulation is designed around fewer objects.

For presentations in the same simulation, the scale of the pictures, the realism of the earth and the satellite, the speed with which the satellite moves, and the extent that the images are labeled and explained all may vary. Beginners can learn faster given greater labeling and distortion of scale. In contrast, providing a high fidelity of speed at which the satellite moves would not benefit anyone—it would appear much too slow.

Fidelity of user actions in this example and many other physical simulations should not vary much even for different phases of instruction. Physical simulations usually give considerable user control to start, stop, and slow down, but do not allow changing the physical laws, which are the object of the lesson.

Feedback correcting input errors would be an instructional technique that decreases fidelity but would be appropriate for any level learner. However, artificial and immediate feedback for actions that lead to catastrophic results (the satellite crashing to earth) might help prevent such actions for beginning learners, whereas natural feedback that allows accidents to occur may be better for advanced learners. However, preventing such events is not as important in physical simulations as in procedural simulations, in which the instructional objective is the correct procedure. In general, fidelity of the *model* and of *presentations* is most critical for physical simulations.

Fidelity in Iterative Simulations An example of an iterative simulation is *Catlab* (Kinnear, 1998), described earlier. It is a genetics simulation in which the learner mates cats and investigates the laws of genetics by observing the characteristics of the parents passed on to the offspring. Considerable variation in the underlying model is possible, from dealing with just one genetic trait through a larger number of them.

Iterative simulations usually deal with multiple variables, and those variables that can be observed and changed is important to learning. Beginning users of *Catlab* have been observed to become confused and frustrated because they try to manipulate several variables at the same time. Additionally, users never see the cats' genotypes—the underlying genetic codes. They see only the phenotypes—the visible characteristics of the cats—and can manipulate those only for the original parents. The related simulation *Kangasaurus: Transmission genetics* (Kinnear, 1997) permits modification of genotypes (the internal genetic characteristics) but with a corresponding increase in simulation complexity.

When an iterative simulation represents rates of change, such as the change in a population over time, the internal model must include a *time increment* which is the amount of time that passes for each recalculation of all the system variables. The size of the time increment affects the quality of the model. In general, smaller time increments always produce more accurate results but cause a simulation to run more slowly. Fidelity of the *model* is most critical in iterative simulations. The designer has much more freedom to vary the fidelity of most other characteristics.

Fidelity in Procedural Simulations *Microsoft Flight Simulator* (Microsoft, 1989) is one of the most popular procedural simulations. It is a particularly instructive example for discussing fidelity. Variation of the underlying model is not as important as the previous types of simulations. The model should be faithful to reality and not affect learning much if the other aspects of fidelity are properly chosen.

The fidelity of the presentations, actions, and feedback are more important. The program confronts the learner with a bewildering array of instruments, views from various windows, and controls to manipulate. The beginner has considerable difficulty attending and reacting to the relevant visual information. The task of operating the simulated airplane is much easier if only a subset of the instruments and controls are present and the wide variety of visual stimuli from outside the airplane is reduced to just a few. But advantageous as this is for the beginner, the advanced learner must be facile with all the instruments, controls, and outside stimuli (such as other airplanes, thunderstorms, and tall buildings). Indeed, it is essential that a student pilot be able to do so when flying a real airplane.

Fidelity of feedback has great importance in procedural simulations because it affects whether incorrect actions are corrected in the future. In a flight simulation, a low-fidelity feedback warning of dangerous actions may be beneficial during initial instruction, but should be faded during practice and assessment.

Fidelity in Situational Simulations Counseling, teaching, and business simulations are typical of this category. They deal with individual human behavior, which is very complex and difficult to predict. That complexity and lack of predictability is difficult to model in a computer program, but it can be done to varying degrees. Various numbers of individuals may be included in such a simulation, just as may be degrees of variation in their behavior. The behavior of real people is not a function of immediately preceding events but of all their experience. A simulation's fidelity may be varied in terms of the degree to which individual behavior is based on multiple past events rather than just the preceding event. Because our knowledge of these real phenomena is imprecise, it is not possible to have high fidelity models, although we may create the *illusion* among users that their fidelity is high.

As with procedural simulations, a more critical issue is the number of *actions* the learner can make. A real teacher faced with misbehaving students can take a wide variety of actions. Classroom behavior simulations typically provide a limited number of actions in multiple-choice format. Users of such simulations often suggest actions that are not included among the programmed alternatives.

Because the emphasis is on learning what to do, feedback fidelity is again important. Beginning learners can benefit from artificial feedback correcting inappropriate actions or preventing unfortunate outcomes. Transfer to the real world requires that more advanced learners see the consequences of their behavior within social systems (see also Reigeluth & Schwartz, 1989).

Simulation Design and Development

This chapter concludes with some comments on overall procedures for designing and development simulations. Simulations are quite unlike other methodologies. The main difference is that simulations require an underlying model. Designers must learn about the real phenomenon (usually to a more sophisticated degree than they must learn content for a tutorial or drill), must create and refine a computer model to simulate it, and must then incorporate that model into an educational program. The following steps are suggested for simulation development:

- Learn and analyze the phenomenon.
- Make design decisions concerning the simulation factors.
- Create and refine the underlying model.
- Transfer the model into your authoring software.
- Develop the user interface in the authoring software.
- Develop instructional supports in the authoring software.

Learning and analyzing the phenomenon includes analyzing the knowledge to be

tions (Alessi, 2000b). That information influences both creation of the underlying model and the design of the lesson.

Making design decisions should be based on the characteristics of the knowledge, analysis of the learners and their characteristics, and the desired learning outcomes. The design process consists of making decisions about outcomes and each of the simulation factors. Some factor decisions impact other ones, so a four-level sequence of making successively finer design decisions should be used. The first decisions, which impact all following ones, include the relative importance of initial learning versus transfer, motivation of learners and techniques to enhance motivation, and techniques to support learning. The second level, which will follow fairly logically from the first, includes decisions about fidelity, learner control, delivery mode, and instructional strategy. The third level encompasses design details such as inputs and outputs, whether time is static or dynamic, and types of images users see. The fourth level includes many decisions about all the details of individual screens, texts, pictures, menus, buttons, sounds, movies, animations, user actions, feedback, and learning supports.

Creating a computer model is not easy in authoring systems (such as *Authorware* or *Director*) or in standard programming languages (such as *C* or *Java*). Modeling software such as *STELLA* (High Performance Systems, 1987) or *PowerSim* (PowerSim, 1999) are much better. They allow the developer to create a system diagram in which variables are represented by icons and the cause–effect relationships between variables are represented by arrows connecting them. The programs generate model equations and, when the model is run, can display either tables of numbers or graphs that describe system behavior over time and under various circumstances.

Figures 7.33 through 7.36 show examples from *STELLA,* which is based on the System Dynamics modeling approach created by Jay Forrester in the 1960s (Forrester, 1961; 1968; 1969; 1971). The rectangles in Figure 7.33 represent the primary variables in the system, in this case the amount of snow and ice in a glacier. The circles with little arrows on top of them represent the rates at which these variables increase or decrease. The plain circles represent other variables or constants that affect the system, such as the

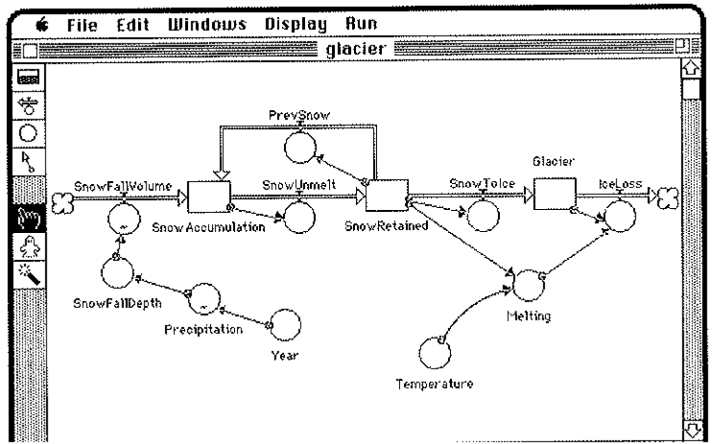
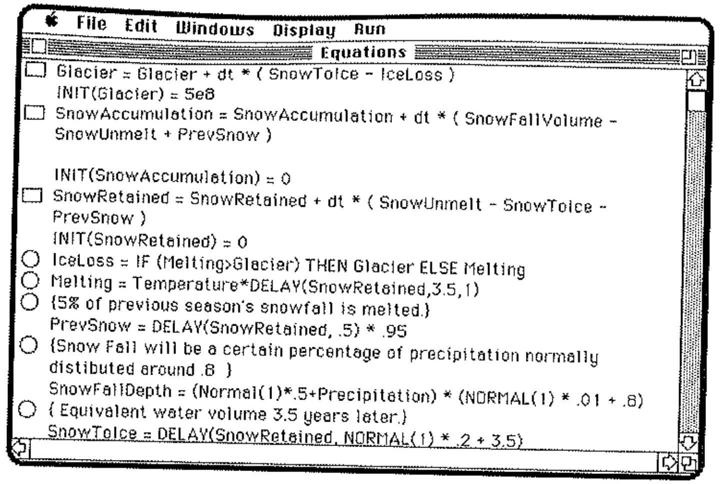


FIGURE 7.33

The Flow Diagram Describing a Simulation Model in *STELLA*

Courtesy of High Performance Systems, glacier model by James Quinn and Jay Cook.



```
☐ Glacier = Glacier + dt * ( SnowToIce - IceLoss )
  INIT(Glacier) = 5e8
☐ SnowAccumulation = SnowAccumulation + dt * ( SnowFallVolume -
  SnowUnmelt + PrevSnow )

  INIT(SnowAccumulation) = 0
☐ SnowRetained = SnowRetained + dt * ( SnowUnmelt - SnowToIce -
  PrevSnow )
  INIT(SnowRetained) = 0
○ IceLoss = IF (Melting>Glacier) THEN Glacier ELSE Melting
○ Melting = Temperature*DELAY(SnowRetained,3.5,1)
○ {5% of previous season's snowfall is melted.}
  PrevSnow = DELAY(SnowRetained, .5) * .95
○ {Snow Fall will be a certain percentage of precipitation normally
  distibuted around .8  }
  SnowFallDepth = (Normal(1)*.5+Precipitation) * (NORMAL(1) * .01 + .8)
○ { Equivalent water volume 3.5 years later.}
  SnowToIce = DELAY(SnowRetained, NORMAL(1) * .2 + 3.5)
```

FIGURE 7.34

The Equations Generated by the Flow Diagram in *STELLA*

Courtesy of High Performance Systems.

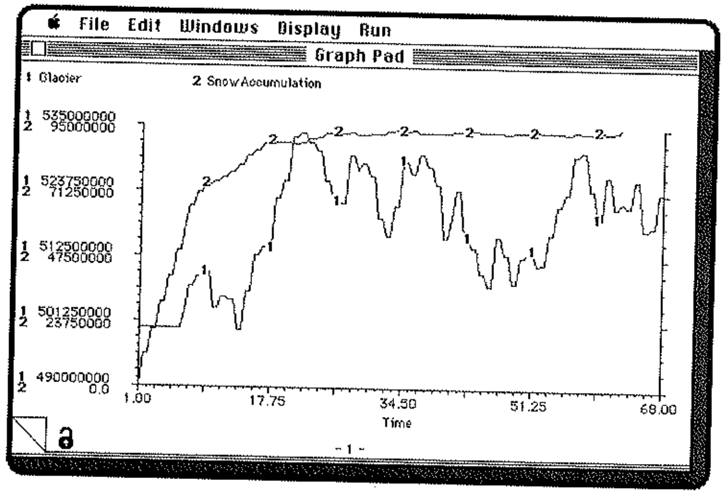


FIGURE 7.35

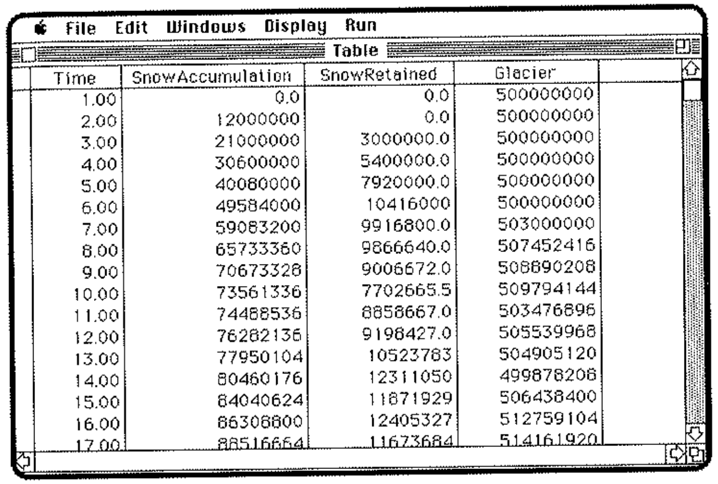
The Graph Generated by Running a *STELLA* Model

Courtesy of High Performance Systems.

amount of precipitation and air temperature. The plain arrows show cause–effect relationships, indicating which variables or constants affect which other ones.

The relationships, as the diagram shows, can be many. The modeling software automatically figures out the necessary equations and their solutions, shown in the *equations* window of Figure 7.34. When the simulation is run, the software progressively increments time and calculates changes in all variables for each time increment. The results are shown as a graph in Figure 7.35 and as a table of numbers in Figure 7.36. The developer can easily change the relationships of variables or their initial values and see resulting changes in equations and system behavior.

Modeling software has obvious utility for the simulation developer. But useful as it is for modeling, it is usually not good for creating user interfaces or instructional support features. Rather, the equations, graphs, or tables generated in *STELLA* or *PowerSim* may be copied into authoring software, such as *Authorware*. In that kind of authoring package, you can then develop an appropriate interface and add instructional support features. This process is explained more fully in Alessi (2000a).



Time	SnowAccumulation	SnowRetained	Glacier
1.00	0.0	0.0	500000000
2.00	12000000	0.0	500000000
3.00	21000000	3000000.0	500000000
4.00	30600000	5400000.0	500000000
5.00	40080000	7920000.0	500000000
6.00	49584000	10416000	500000000
7.00	59083200	9916800.0	503000000
8.00	65733360	9866640.0	507452416
9.00	70673328	9006672.0	508890208
10.00	73561336	7702665.5	509794144
11.00	74488536	8858667.0	503476896
12.00	76282136	9198427.0	505539968
13.00	77950104	10523783	504905120
14.00	80460176	12311050	499878208
15.00	84040624	11871929	506438400
16.00	86308800	12405327	512759104
17.00	88516664	11673684	514161920

FIGURE 7.36

A Table of Numbers Generated by Running a *STELLA* Model

Courtesy of High Performance Systems.

Conclusion

Simulation is an instructional methodology that takes full advantage of the computer for learning and instruction. Simulations improve on tutorials and drills with enhanced motivation, transfer of learning, efficiency, and flexibility. They have the advantages of convenience, safety, and controllability over real experiences; provide a good precursor to real experiences; and are useful for giving learners experiences that are not otherwise possible. On the other hand, simulations are the most challenging of all methodologies to design and develop. The designer needs more understanding of the content and the learners, must attend to many complex factors, and faces more sophisticated programming to implement a simulation model and embed it within an effective program for learning.

REFERENCES AND BIBLIOGRAPHY

A.D.A.M. Software. (1997). *A.D.A.M Interactive Anatomy.* [Computer software]. Atlanta, GA: Author.

Acovelli, M., & Gamble, M. (1997). A coaching agent for learners using multimedia simulations. *Educational Technology, 37*(2), 44–48.

Alessi, S. M., & Trollip, S. R. (1991). *Computer-based instruction: Methods & development* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Alessi, S. M. (1988). Fidelity in the design of instructional simulations. *Journal of Computer-Based Instruction, 15*(2), 40–47.

Alessi, S. M. (2000a). Designing educational support in system dynamics based interactive learning environments. *Simulation & Gaming, 31*(2), 178–196.

Alessi, S. M. (2000b). Simulation design for training and assessment. In H. F. O'Neil, Jr. & D. H.

Andrews (Eds.), *Aircrew training and assessment* (pp. 197–222). Mahwah, NJ: Lawrence Erlbaum.

Alessi, S. M., & Johnson, L. A. (1992). Simulations for dental licensure examinations: Reliability and validity. *Simulations/Games for Learning, 22*(4), 286–307.

Anderson, J. R. (1980). *Cognitive psychology and its implications.* San Francisco: W. H. Freeman.

Andrews, D. H., Carroll, L. A., & Bell, H. H. (1995). The future of selective fidelity in training devices. *Educational Technology, 35*(6), 32–36.

Andrews, D. H., Dineen, T., & Bell, H. H. (1999). The use of constructive modeling and virtual simulation in large-scale team training: A military case study. *Educational Technology, 39*(1), 24–28.

Beaudin, B. P. (1987). Enhancing the transfer of job-related learning from the learning environment to the workplace. *Performance & Instruction, 26*(9/10), 19–21.

Boyle, T., Stevens-Wood, B., Feng, Z., & Tikka, A. (1996). Structured learning in a virtual environment. *Computers & Education: An International Journal, 26*(1-3), 41–49.

Brehmer, B., & Dorner, D. (1993). Experiments with computer-simulated microworlds: Escaping both the narrow straits of the laboratory and the deep blue sea of the field study. *Computers in Human Behavior, 9*(2/3), 171–184.

Broad, M. L., & Newstrom, J. W. (1992). *Transfer of training: Action-packed strategies to ensure high payoff from training investments.* Reading, MA: Addison-Wesley.

Bruner, J. S. (1966). *Toward a theory of instruction.* Cambridge, MA: Harvard University Press.

Burki-Cohen, J., Soja, N. N., & Longridge, T. (1998). Simulator platform motion—The need revisited. *The International Journal of Aviation Psychology, 8*(3), 293–317.

Butterfield, E. C., & Nelson, G. D. (1989). Theory and practice of teaching for transfer. *Educational Technology Research & Development, 37*(3), 5–38.

Carter, G., & Trollip, S. R. (1980). A constrained maximization extension to incremental transfer effectiveness, or how to mix your training technologies. *Human Factors, 22,* 141–152.

Chandler, J. R. (1995). *Air Pollution.* [Computer software]. Stuart, FL: EME.

Chandler, T. N. (1996). System for training aviation regulations (STAR): Using multiple vantage points to learn complex information through scenario-based instruction and multimedia techniques. *Journal of Educational Multimedia and Hypermedia, 5*(3/4), 349–378.

Choi, W. (1997). Designing effective scenarios for computer-based instructional simulation: Classification of essential features. *Educational Technology, 37*(5), 13–21.

Christel, M. G. (1994). The role of visual fidelity in computer-based instruction. *Human–Computer Interaction, 9,* 183–223.

Clark, R. E., & Voogel, A. (1985). Transfer of training principles for instructional design. *Educational Communication and Technology Journal, 33*(2), 113–123.

Collins, A. (1994). Goal-based scenarios and the problem of situated learning: A commentary on Andersen Consulting's design of goal-based scenarios. *Educational Technology, 34*(9), 30–32.

Cormier, S. M., & Hagman, J. D. (Eds.). (1987). *Transfer of learning: Contemporary research and applications.* San Diego: Academic Press.

de Hoog, R., de Jong, T., & de Vries, F. (1991). Interfaces for instructional use of simulations. *Education & Computing, 6,* 359–385.

de Jong, T., de Hoog, R., & de Vries, F. (1993). Coping with complex environments: The effects of providing overviews and a transparent interface on learning with a computer simulation. *International Journal of Man–Machine Studies, 39,* 621–639.

de Jong, T., Martin, E., Zamarro, J-M., Esquembre, F., Swaak, J., & van Joolingen, W. R. (1999). The integration of computer simulation and learning support: An example from the physics domain of collisions. *Journal of Research in Science Teaching, 36*(5), 597–615.

de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*(2), 179–201.

Detterman, D. K., & Sternberg, R. J. (Eds.). (1993). *Transfer on trial: Intelligence, cognition, and instruction.* Norwood, NJ: Ablex.

Doerner, D. (1980). On the difficulties people have in dealing with complexity. *Simulation & Games, 11*(1), 87–106.

Duchastel, P. (1994). Learning environment design. *Journal of Educational Technology Systems, 22*(3), 225–233.

Electronic Arts. (1998). *SimEarth.* [Computer software]. Walnut Creek, CA: Author.

Electronic Arts. (1996a). *SimCity Classic.* [Computer software]. Walnut Creek, CA: Author.

Electronic Arts. (1996b). *SimFarm.* [Computer software]. Walnut Creek, CA: Author.

EME. (1999). *Burette.* [Computer software]. Stuart, FL: Author.

Faria, A. J. (1998). Business simulation games: Current usage levels—An update. *Simulation & Gaming, 29*(3), 295–308.

Forrester, J. W. (1961). *Industrial dynamics.* New York: John Wiley & Sons.

Forrester, J. W. (1968). *Principles of systems* (second preliminary edition). Cambridge, MA: Wright-Allen Press.

Forrester, J. W. (1969). *Urban dynamics.* Cambridge, MA: The M.I.T. Press.

Forrester, J. W. (1971). *World dynamics.* Cambridge, MA: Wright-Allen Press.

Foxon, M. (1993). A process approach to the transfer of training. Part 1: The impact of motivation and supervisor support on transfer maintenance. *Australian Journal of Educational Technology, 9*(2), 130–143.

Foxon, M. (1994). A process approach to the transfer of training. Part 2: Using action planning to facilitate the transfer of training. *Australian Journal of Educational Technology, 10*(1), 1–18.

Frame, M. W., Flanagan, C. D., Frederick, J., Gold, R., & Harris, S. (1997). You're in the hot seat: An ethical decision-making simulation for counseling students. *Simulation & Gaming, 28* (1), 107–115.

Frederiksen, J. R., White, B. Y., & Gutwill, J. (1999). Dynamic mental models in learning science: The importance of constructing derivational linkages among models. *Journal of Research in Science Teaching, 36*(7), 806–836.

Funke, J. (1991). Solving complex problems: Exploration and control of complex systems. In R. J. Sternberg & P. A. Frensch (Eds.), *Complex problem solving: Principles and mechanisms* (pp. 185–222). Hillsdale, NJ: Lawrence Erlbaum.

Gagné, R. M. (1954). Training devices and simulators: Some research issues. *The American Psychologist, 9,* 95–107.

Garavaglia, P. L. (1993). How to ensure transfer of training. *Training and Development, 47*(10), 63–68.

Gentner, D., & Stevens, A. (Eds.). (1983). *Mental models.* Hillsdale, NJ: Lawrence Erlbaum.

Gibbons, A. S., Fairweather, P. G., Anderson, T. A., & Merrill, M. D. (1997). Simulation and computer-based instruction: A future view. In C. R. Dills & A. J. Romiszowski (Eds.), *Instructional development paradigms* (pp. 769–804). Englewood Cliffs, NJ: Educational Technology.

Gist, M. E., Bavetta, A. G., & Stevens, C. K. (1990). Transfer training method: Its influence on skill generalization, skill repetition, and performance level. *Personnel Psychology, 43*(3), 501–523.

Gist, M. E., Stevens, C. K., & Bevetta, A. G. (1991). Effects of self-efficacy and post-training intervention on the acquisition and maintenance of complex interpersonal skills. *Personnel Psychology, 44*(4), 837–861.

Goodyear, P., Njoo, M., Hijne, H., & van Berkum, J. J. A. (1991). Learning processes, learner attributes and simulations. *Education & Computing, 6,* 263–304.

Guckenberger, D., & Guckenberger, E. (1996). *Above real-time training theory: Implications of neuroscience, cognitive psychology, & learning theory in simulator training.* Orlando, FL: Institute for Simulation and Training.

Hays, R. T. (1980). *Simulator fidelity: A concept paper* (Technical Report 490). Alexandria, VA.

Herz, B., & Merz, W. (1998). Experiential learning and the effectiveness of economic simulation games. *Simulation & Gaming, 29*(2), 238–250.

High Performance Systems. (1987). *STELLA.* [Computer software]. Lyme, NH: Author.

Holsbrink-Engels, G. A. (1997). Computer-based role-playing for interpersonal skills training. *Simulation & Gaming, 28*(2), 164–180.

Interactive Magic. (1996). *Capitalism.* [Computer software]. Research Triangle Park, NC: Author.

Interactive Magic. (1997). *Capitalism Plus.* [Computer software]. Research Triangle Park, NC: Author.

Jarz, E. M., Kainz, G. A., & Walpoth, G. (1997). Multimedia-based case studies in education: Design, development, and evaluation of multimedia-based case studies. *Journal of Educational Multimedia and Hypermedia, 6*(1), 23–46.

Johnson, W. B., & Norton, J. E. (1992). Modeling student performance in diagnostic tasks: A decade of evolution. In J. W. Regian & V. J. Shute (Eds.), *Cognitive approaches to automated instruction* (pp. 195–216). Hillsdale, NJ: Lawrence Erlbaum.

Kang, S.-H. (1996). The effects of using an advance organizer on students' learning in a computer simulation environment. *Journal of Educational Technology Systems, 25*(1), 57–65.

Keegan, M. (1995). *Scenario educational software: Design and development of discovery learning.* Englewood Cliffs, NJ: Educational Technology.

Keller, J. M., & Suzuki, K. (1988). Use of the ARCS motivation model in courseware design. In D. H. Jonassen (Ed.), *Instructional designs for microcomputer courseware* (pp. 401–434). Hillsdale, NJ: Lawrence Erlbaum.

Keys, J. B. (1997). Strategic management games: A review. *Simulation & Gaming, 28*(4), 395–422.

Kinnear, J. (1997). *Kangasaurus: Transmission Genetics.* [Computer software]. Stuart, FL: EME.

Kinnear, J. (1998). *Catlab.* [Computer software]. Stuart, FL: EME.

Lepper, M. R., & Chabay, R. W. (1985). Intrinsic motivation and instruction: Conflicting views on the role of motivational processes in computer-based education. *Educational Psychologist, 20*(4), 217–230.

Lesgold, A., Eggan, G., Katz, S., & Govinda, R. (1992). Possibilities for assessment using computer-based apprenticeship environments. In J. W. Regian & V. J. Shute (Eds.), *Cognitive approaches to automated instruction* (pp. 49–80). Hillsdale, NJ: Lawrence Erlbaum.

Li, Y., Borne, I., & O'Shea, T. (1996). A scenario design tool for helping students learn mechanics. *Computers & Education: An International Journal, 26*(1-3), 91–99.

Lopez, G. (1994). *Population Concepts.* [Computer software]. Stuart, FL: EME.

Lopez, G. (1998). *Community Dynamics* . [Computer software]. Stuart, FL: EME.

Malone, T. W., & Lepper, M. R. (1987). Making learning fun: A taxonomy of intrinsic motivations for learning. In R. E. Snow & M. J. Farr (Eds.), *Aptitude, learning, and instruction: III. Conative and affective process analysis* (pp. 223–253). Hillsdale, NJ: Lawrence Erlbaum.

McAteer, E., Neil, D., Barr, N., Brown, M., Draper, S., & Henderson, F. (1996). Simulation software in a life sciences practical laboratory. *Computers & Education: An International Journal, 26*(1-3), 101–112.

McHaney, R. (1991). *Computer simulation: A practical perspective.* San Diego: Academic Press.

Meridian Creative Group. (1996). *Oval Office: Challenge of the Presidency.* [Computer software]. Erie, PA: Author.

Microsoft. (1989). *Microsoft Flight Simulator.* [Computer software]. Redmond, WA: Author.

Milheim, W. D. (1995). Virtual reality and its potential applications in education and training. *Machine-Mediated Learning, 5*(1), 43–55.

MECC. (1995a). *Africa Trail.* [Computer software]. Minneapolis: Author.

MECC. (1995b). *MayaQuest.* [Computer software]. Minneapolis: Author.

O'Neil, H. F. J., Allred, K., & Dennis, R. A. (1997). Use of computer simulation for assessing the interpersonal skill of negotiation. In H. F. O'Neil (Ed.), *Workforce readiness: Competencies and assessment* (pp. 229–254). Mahwah, NJ: Lawrence Erlbaum.

O'Neil, H. F. J., Chung, G. K. W. K., & Brown, R. S. (1997). Use of networked simulations as a context to measure team competencies. In H. F. O'Neil (Ed.), *Workforce readiness: Competencies and assessment* (pp. 411–452). Mahwah, NJ: Lawrence Erlbaum.

Osgood, C. E. (1949). The similarity paradox in human learning: A resolution. *Psychological Review, 56,* 132–143.

Papert, S. A. (1980). *Mindstorms.* New York: Basic Books.

Pappo, H. A. (1998). *Simulations for skills training.* Englewood Cliffs, NJ: Educational Technology.

Parry, S. (1990). Ideas for improving transfer of training. *Adult Learning, 1*(7), 19–23.

Pierian Spring Software. (1997). *BioLab Frog.* [Computer software]. Portland, OR: Author.

Povenmire, K., & Roscoe, S. N. (1973). The incremental transfer effectiveness of a ground-based general-aviation trainer. *Human Factors, 15,* 534–542.

PowerSim. (1999). *PowerSim.* [Computer software]. Bergen, Norway: Author.

Practising Law Institute. (1996). *The Interactive Courtroom.* [Computer software]. New York: Author.

Psotka, J. (1995). Immersive training systems: Virtual reality and education and training. *Instructional Science, 23*(5-6), 405–431.

Reigeluth, C. M. (1979). In search of a better way to organize instruction: The elaboration theory. *Journal of Instructional Development, 2*(3), 8–15.

Reigeluth, C. M., & Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. *Journal of Computer-Based Instruction, 16*(1), 1–10.

Rieber, L. P. (1996). Animation as feedback in a computer-based simulation: Representation matters. *Educational Technology Research and Development, 44*(1), 5–22.

Rieber, L. P. (1996). Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational Technology Research and Development, 44*(2), 43–58.

Rieber, L. P., & Parmley, M. W. (1995). To teach or not to teach? Comparing the use of computer-based simulation in deductive versus inductive approaches to learning with adults in science. *Journal of Educational Computing Research, 13*(4), 359–374.

Rigney, J. W., & Lutz, K. A. (1976). Effect of graphic analogies of concepts in chemistry on learning and attitude. *Journal of Educational Psychology, 68,* 305–311.

Roberts, N., Anderson, D., Deal, R., Garet, M., & Shaffer, W. (1983). *Computer simulation: A system dynamics modeling approach.* Reading, MA: Addison-Wesley.

Roscoe, S. N. (1971). Incremental transfer effectiveness. *Human Factors, 13,* 561–567.

Roscoe, S. N. (1972). A little more on incremental transfer effectiveness. *Human Factors, 14,* 363–364.

Ruben, B. D., & Lederman, L. C. (1982). Instructional simulation gaming: Validity, reliability, and utility. *Simulation & Games, 13*(2), 233–244.

Salas, E., Bowers, C. A., & Rhodenizer, L. (1998). It is not how much you have but how you use it: Toward a rational use of simulation to support aviation training. *The International Journal of Aviation Psychology, 8*(3), 197–208.

Schank, R., & Cleary, C. (1995). *Engines for education.* Hillsdale, NJ: Lawrence Erlbaum.

Schneider, W. (1985). Training high-performance skills: Fallacies and guidelines. *Human Factors, 27,* 285–300.

Simulation & Gaming: An International Journal of Theory, Design, and Research. Beverly Hills, CA: Sage.

Simulations Plus. (1998a). *Future Lab: Circuits for Physical Science.* [Computer software]. Lancaster, CA: Author.

Simulations Plus. (1998b). *Future Lab: Gravity for Physical Science.* [Computer software]. Lancaster, CA: Author.

Simulations Plus. (1998c). *Future Lab: Optics for Physical Science.* [Computer software]. Lancaster, CA: Author.

Softkey Multimedia. (1996). *Amazon Trail.* [Computer software]. Knoxville, TN: Author.

Starfield, A. M., Smith, K. A., & Bleloch, A. L. (1990). *How to model it: Problem solving for the computer age.* New York: McGraw-Hill.

Sternberg, R. J., & Frensch, P. A. (1993). Mechanisms of transfer. In D. K. Detterman & R. J. Sternberg (Eds.), *Transfer on trial: Intelligence, cognition, and instruction* (pp. 25–38). Norwood, NJ: Ablex.

Su, Y.-L. D. (1984). *A review of the literature on training simulators: Transfer of training and simulator fidelity.* (Technical Report No. 84-1). (Eric No. ED 246 864). Atlanta, GA: Georgia Institute of Technology.

Swaak, J., van Joolingen, W. R., & de Jong, T. (1998). Supporting simulation-based learning: The effects of model progression and assignments on definitional and intuitive knowledge. *Learning and Instruction, 8*(3), 235–252.

Tom Snyder. (1997). *Choices, Choices: Taking Responsibility.* [Computer software]. Watertown, MA: Author.

Thiagarajan, S. (1998). The myths and realities of simulations in performance technology. *Educational Technology, 38*(5), 35–41.

Thurman, R. A. (1993). Instructional simulation from a cognitive psychology viewpoint. *Educational Technology Research and Development, 41*(4), 75–89.

Towne, D. M. (1995). *Learning and instruction in simulation environments.* Englewood Cliffs, NJ: Educational Technology.

Towne, D. M., de Jong, T., & Spada, H. (Eds.). (1993). *Simulation-based experiential learning.* Berlin: Springer-Verlag.

Trollip, S. R. (1979). The evaluation of a complex, computer-based flight procedures trainer. *Human Factors, 22*(1), 47–54.

Underwood, G., Underwood, J., Pheasey, K., & Gilmore, D. (1996). Collaboration and discourse while programming the KidSim microworld simulation. *Computers & Education: An International Journal, 26*(1-3), 143–151.

van Joolingen, W. R., & de Jong, T. (1991). Characteristics of simulations for instructional settings. *Education & Computing, 6,* 241–262.

van Joolingen, W. R., & de Jong, T. (1997). An extended dual search space model of scientific discovery learning. *Instructional Science, 25*(5), 307–346.

Veenman, M. V. J., & Elshout, J. J. (1995). Differential effects of instructional support on learning in simulation environments. *Instructional Science, 22*(5), 363–383.

Watson-Papelis, G. (1995). *Simulator effects in a high fidelity driving simulator as a function of visuals and motion.* Orlando, FL: Institute for Simulation and Training.

Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge.* Los Altos, CA: Morgan Kaufmann.

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

Wolfe, J. (1985). The teaching effectiveness of games in collegiate business courses: a 1973–1983 update. *Simulation & Games, 16*(3), 251–288.

Yellon, S. (1992). M. A. S. S.: A model for producing transfer. *Performance Improvement Quarterly, 5*(2), 13–23.

SUMMARY OF SIMULATIONS

Use simulations instead of actual experience when the latter is unsafe, costly, very complex, or logistically difficult.

Use simulations instead of other media or methodologies when motivation, transfer of learning, or efficiency need to be increased.

Simulations can be used for any or several of the phases of instruction.

Use a short title page.

Give objectives, including the instructional purpose of the simulation.

Be clear whether the simulation teaches *about* something or *how to do* something.

Give directions when they are first needed and allow users to retrieve directions at any time.

Do not use overly detailed graphics. Provide just as much detail as is necessary to convey the necessary information.

Thoroughly understand the phenomenon before you try to develop an instructional simulation.

Use simulation languages to create and refine the underlying simulation model.

Use modes of presentation and user action that enhance fidelity.

Use lower fidelity for beginning learners.

Use higher fidelity for advanced learners.

Perceived fidelity may enhance motivation and learning more than actual fidelity.

Use immediate feedback (regardless of fidelity) for beginning learners.

Use natural feedback (regardless of immediacy) for more advanced learners.

In physical and iterative simulations, analysis of the fidelity of underlying models and presentations is usually critical.

In procedural and situational simulations, analysis of the fidelity of learner actions and system reactions is usually critical.

Allow the learner to return to initial choices.

Allow internal restarting.

Allow temporary termination at any time.

Provide restarting after temporary termination.

Clear any displays and give a final message at the end.