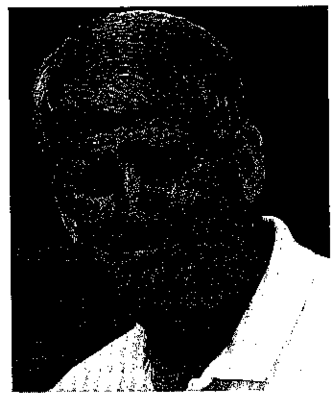
9
Simulation Approach to Instruction

ANDREW S. GIBBONS
Brigham Young University

MARK MCCONKIE
Utah State University

KAY KYEONGJU SEO
University of Cincinnati

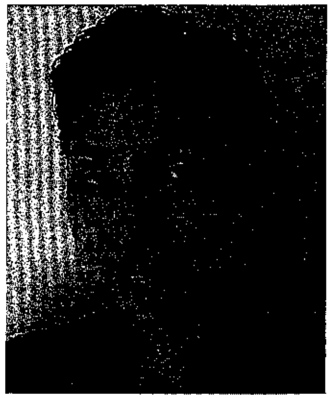
DAVID A. WILEY
Utah State University



Andrew S. Gibbons is department chair in instructional psychology and technology at Brigham Young University. Prior to that, he was a faculty member at Utah State University. He led instructional design projects in industry for 18 years at Wicat Systems, Inc. and Courseware Inc. Dr. Gibbons' work has included large-scale training development projects, reengineering of the development (ISD) process, computer-based instruction, military and commercial aviation training development, and research and development on instructional simulations. Dr. Gibbons' current research focuses on the architecture of instructional designs. He has published a design theory of model-centered instruction, proposed a general layering theory of instructional designs, and is currently studying the use of design languages in relation to design layers as a means of creating instructional systems that are adaptive, generative, and scalable.

Mark McConkie is a PhD candidate in the instructional technology program at Utah State University. In both academic and industry settings, his work has involved the design and development of a broad range of instructional simulations incorporating the principles of model-centered instruction. He has worked for 10 years as an instructional systems design specialist for Parker Hannifin Corporation, a large global manufacturing company, and is currently a Training and Development Manager for its Aerospace Group. He has also worked as a research assistant at the Center for Open Sustainable Learning (COSL), and on an NSF-sponsored curriculum reform grant for the department of Biological and Irrigation Engineering at Utah State University.

Kay Kyeongju Seo is assistant professor of instructional design and technology at the University of Cincinnati in Cincinnati, Ohio. Her research interests include computer-mediated communication, technology integration in K-12 and higher education, multimedia-assisted project-based learning, instructional design languages, Web site usability testing, and instructional simulations and microworlds. Prior to joining the University of Cincinnati, she was assistant professor of educational technology at Rockhurst University in Kansas City, Missouri. At Rockhurst, she focused on preparing in-service and preservice teachers to use technology in their classrooms. Seo received her PhD in instructional technology from Utah State University in Logan, Utah.



David A. Wiley is associate professor of instructional technology at Utah State University, director of the Center for Open and Sustainable Learning, and chief openness officer of Flat World Knowledge. He holds a PhD in instructional psychology and technology from Brigham Young University and a BFA in Music from Marshall University. He has previously been a nonresident fellow at the Center for Internet and Society at Stanford Law School, a visiting scholar at the Open University of the Netherlands, and is a recipient of the U.S. National Science Foundation's CAREER grant. His career is dedicated to increasing access to educational opportunity for everyone around the world.

EDITORS' FOREWORD

Preconditions

Content

- *Integrated skills that consist of multiple complex actions in a fluid sequence and changing circumstances*

Learners

- *All learners*

Learning environments

- *Environments that have the appropriate tools (computers or other materials)*
- *Instructional simulations or microworlds must have augmentations*

Instructional development constraints

- *Simulation or microworld cost must not exceed either available resources (money and time) or benefits*

Values

about ends (learning goals)

- *Understand principles and relationships in dynamic systems*
- *Developing skills for dealing with complex systems*

about priorities (criteria for successful instruction)

- *Effectiveness, efficiency, and appeal can all be maximized, as long as the user population is large enough to make it economical.*

about means (instructional methods)

- *The learning experience should be adaptive, generative, and scalable.*
- *The learning experience should involve authentic tasks and contexts.*
- *The learning experience should involve a dynamic model of physical and/or conceptual systems.*
- *Learner interactions with the model should result in state changes.*
- *The learning experience should have at least one designed augmenting instructional function.*

about power (to make decisions about the previous three)

- *Learners are active participants who have some degree of free agency within the simulated environment.*

Universal Methods

1. ***for the content function***
   - *Create an abstract model first, then a computerized model.*
   - *Select the "right" model.*
   - *Select the appropriate type(s) of models (environment, cause-effect, or performance model).*
   - *Select the appropriate forms of models (semantic networks, production rules, equations, Bayesian networks, system dynamics, or object*

- *Make a limited number of control points available to learners for manipulation of the system.*
- *Align the different kinds of fidelity and resolution of the model with the learning needs, inasmuch as costs allow.*
- *Escalate performance difficulty by supplying progressions of advanced models, advanced problems, or both.*

2. ***for the strategy function (instructional augmentations)***
   - *Design the physical setting and siting.*
   - *Design the social settings (participant roles and patterns of initiative-sharing).*
   - *Describe the properties of models and the range of performances that encompass subject-matter goals, problem solving goals, and learning-to-learn goals within each model.*
   - *Assign instructional scopes (goals) to event blocks and arrange them into sequences (progressions).*
   - *Specify event forms and classes.*
   - *Design augmentation types and rules.*
   - *Use a dramatic context.*
   - *Design the means of supplying problem-related information to learners.*
3. ***for the control function***
   - *Provide user controls for each main function.*
4. ***for the messaging function***
   - *Generate message units that have certain elements of a human tutor.*
   - *Use approaches for structuring messages.*
   - *Use execution-time construction of messages.*
5. ***for the representation function***
   - *Generate and assemble representation elements.*
   - *Change model states according to the cycles of the model.*
   - *Consider giving the learner alternative vantage points.*
   - *Decide what representations a single message should be given.*
   - *Design rules for the execution of the continuously changing displays.*
   - *Design rules for manipulating time and space.*
   - *Design traces that represent trends of change.*
6. ***for the media-logic function***
   - *Execute representations and computations.*
7. ***for the data management function***
   - *Manage data resulting from interactions.*
   - *Design data collection points and variables.*
   - *Decide on interpretation variables and rules.*
   - *Design a data-gathering framework.*
   - *Decide on points at which to give information to the learner.*

—CMR & AAC

SIMULATION APPROACH TO INSTRUCTION

This chapter describes guidelines for the theory-based design of instructional simulations and microworlds. It adds to earlier writing on simulation by Alessi and Trollip (1991), de Jong and van Joolingen (1998), Gredler (2004), Gibbons, Fairweather, Anderson, and Merrill (1997), Munro, Breaux, Paltrey, and Sheldon (2002), and Reigeluth and Schwartz (1989), in an attempt to foster a common knowledge base in this area. Simulation and microworld design comprises many independent sectors of what has become a large and prosperous industry that creates products of great diversity in numerous subject-matter areas and surface forms.

Simulation and microworld designs are more diverse in their surface features than other instructional forms, and it is sometimes not possible to tell from the surface whether a product is a simulation or a look-alike animation. It seems almost impossible to define a set of design principles that apply to the large and varied class of products called instructional simulations unless we identify design-theoretic schemes that show the similarity of product designs under the surface. We have organized this chapter around such a scheme, which is described later. We will attempt to answer the questions:

- What are instructional simulations and microworlds?
- What underlying structural principles relate them together?
- What design principles apply to the entire class of instructional simulations and microworlds?

Defining Instructional Simulation

For the purposes of this chapter, instructional simulations are defined as:

1. One or more dynamic models of physical or conceptual systems…
2. That engage the learner in interactions with the models that result in state changes…
3. According to a nonlinear logic…
4. With supplementation by one or more designed augmenting instructional functions…
5. Employed in the pursuit of one or more instructional goals.

An instructional simulation involves interaction with a dynamic, changing, computable model; new states of the model are determined by the learner's actions toward the model or by its own continuous computations. One can learn from general purpose simulation models, but a general purpose simulation can be considered instructional only if it is augmented with one or more auxiliary instructional functions that assist the learner in some way during learner–model interaction (Gibbons, 2001; Gibbons et al., 1997). An unaugmented model has limited instructional value, can create instructional inefficiencies, and can lead the learner into misinterpretations and misconceptions.

Simulations are used in the training of integrated skills that consist of multiple judgments, decisions, and actions that take place in a fluid sequence in response to changing circumstances.[1] What is learned and practiced with the aid of an instructional simulation is the ability to adapt action to a momentary problem-solving need. Simulations provide practice in carrying out complex tasks and also in selecting which tasks to carry out at a given moment. They can be used to teach learning strategies and to help learners to achieve the capacity for self-directed learning. Instructional simulations do this through augmentation within an environment that includes some degree of scaffolding, coaching, feedback, or on-the-spot instruction tailored to the circumstances and requirements of performance. Because of this dynamism, the order of instructional events, messages, and representations may be decided at the moment of need and constructed at that moment from primitive elements and computed data.

Because simulations represent an adaptive and potentially costly form of instruction, three general criteria apply to the design of instructional simulations (Atkinson & Wilson, 1969):

- *The criterion of adaptivity*—The ability to modify qualities of the instructional experience based on the actions of the learner
- *The criterion of generativity*—The ability to generate some portion of the instructional artifact at the time of use
- *The criterion of scalability*—The ability to produce instructional experiences in greater quantity without corresponding linear increases in cost

Defining Microworld

A microworld is a model-centered environment in which a model *is constructed* by the learner, using parts and tools supplied by the designer (Colella, Klopfer, & Resnick, 2001; Papert, 1993; Rieber, 1996). The model is then used by the learner through guided or self-directed exploration and experimentation to learn principles and relationships regarding the behavior of the model, as constructed, under different experimental conditions. The construction of multiple models and experimentation with each model provides learning about the sometimes complex behavior of simple systems and, ultimately, about how to conduct learning through self-directed experiment—an approach to instruction termed *constructionist* (Kafai, 2006; Papert, 1993).

The microworld can be considered a species of instructional simulation. That is, microworlds possess all of the characteristics of instructional simulations, but the microworld concept has additional, specialized characteristics that not all instructional simulations share:

1. The dynamic model is designed and constructed by the learner.
2. A construction environment created by the designer allows a set of primitive model-building elements to be joined together to embody and illustrate cause–effect model relationships.
3. Tools within the construction environment may be provided by the designer to support learning activities.
4. The set of construction elements is often themed. The elements are like characters in multiple stories that the learner can tell.
5. The learner's nonlinear interactions in constructing models and carrying out model interactions are within a specific range constrained by the set of elements supplied by the designer and the operational commands provided by the designer for use by the learner.
6. One or more guided explorations and exploration supports are normally provided.

The value of microworlds as an instructional tool depends on insightful, disciplined design and execution by the designer of the construction environment and the construction elements. Used casually or carelessly, the concept easily deteriorates into a standard, procedural laboratory exercise. The main effect of the microworld method requires the willing—even eager—participation of learners. Through a scaffolded process they learn how to formulate questions that can be answered through manipulations of constructed models, becoming knowledge producers for themselves. In the remainder of this chapter, we will use the term *simulation* to refer to microworlds as well as other kinds of simulations.

Simulation Architecture

The design architecture of an instructional simulation can be viewed as a composite of many subdesigns, including: (1) a core that computes model state changes; (2) a strategic system of model-experience augmentations; (3) a user control system; (4) a message-generation system; (5) a system for creating surface representation; (6) a system for executing the simulation; and (7) a system for data management.[2] A common approach to simulation design is to center the solution on a software architecture, but doing this indicates an implicit decision that other parts of the design will be forced to fit the software architecture if there is a conflict.

We will describe principles for instructional simulation design under the seven functional headings named above:

1. Content function: Supply model content
2. Strategy function: Implement instructional augmentations
3. Control function: Provide user controls

2. *Editors' note: Note that these correspond to Gibbons' and Rogers' seven layers of design (see chapter*

4. Messaging function: Generate message units
5. Representation function: Generate and assemble representation elements
6. Media-logic function: Execute representations and computations
7. Data management function: Manage data resulting from interactions.

Gibbons and Rogers (chapter 14, this volume) suggest that these functional categories apply to instructional designs in general.

1. Content Function: Supply Model Content

This section describes principles for organizing the content functional module of an instructional simulation. Simulation content takes the form of a dynamic model. In this chapter the term *model* means an invisible but computable entity capable of responding to operations upon it in an unpredictable order. Models are dynamic *replicas* of real or imagined systems.

Principle: Abstract Models

Models should be documented independently, in the abstract, before they are embodied into computer code or instructions for human enactment, in order to avoid confusing the model in the designer's mind with the computer program or social structures that will enact it.[3]

Principle: The "Right" Model(s)

The selection of model(s) for instructional simulations sometimes requires subtle distinctions on the designer's part. Bransford, Brown, and Cocking (2000) remark that "one is struck by the complexity of selecting appropriate models for particular mathematical ideas and processes" (p. 168). It is important for designers to avoid the common error of selecting models that are the most visible and interesting or the easiest to build.[4] One design team was so fascinated with the surface interactivity and multimedia attractiveness of an equipment model (a medical analysis machine) that they forgot to design the performance model (making judgments based on the equipment's output) that was of real interest.

Principle: Three Kinds of Models

The simulated model may consist of a single model or a suite of models that interact. Models used for instructional simulations are of three kinds: (1) environment models; (2) cause-effect system models; and (3) performance models. The *environment* model provides a context for the cause–effect and human performance models. The environment model is seldom the focus of instruction, but the designer must design it, because it generates events that influence the states of the other two models. *Cause–effect* models mimic cause–effect systems that are natural or human-made. An *expert performance* model observes and interprets these other two models, makes predictions concerning their future states, discovers or explains cause–effect relationships, operates or controls the models, responds to unexpected changes in model state, and influences the models toward a desired future goal state.[5]

Principle: The Form of Models

Models can take the form of:

Semantic networks. These consist of conceptual networks of meaning nodes and the named relationships among them—propositional knowledge.

Production rules. These are "if…then…" rules constituting decision-action subject-matter.

Equations. These include mathematical and logical formulas that describe a system's behaviors without describing the inner workings of the system. Such collections of formulas normally also specify an order in which the formulas are to be applied, with the output of earlier computations being used as input variables to subsequent computations.

Bayesian networks. These networks define probabilistic pathways between system states. Bayesian networks can be used to represent both conceptual and performance-related content.

System dynamics models. System dynamics models (Milrad, Spector, & Davidsen, 2002; Spector, 2000) identify the elements of a system, along with the input and output variables associated with each element. Formulas or rules are associated with element-to-element relationships and allow computations to update system values in response to events and variable value changes. System dynamics models are useful for describing in detail the cause–effect linkages within a complex system.

Object models. Model content can be captured as a collection of objects, each with its own identity, variable set, behaviors, rules for recomputing internal variable values, and rules for communicating information to other objects through interobject messages.[6]

3. *Editors' note: This does not describe what the instruction should be like, rather what the planning process should be like. Therefore, it is instructional-planning theory, not instructional-event theory.*
4. *Editors' note: Again, this is instructional-planning theory, and it is at a very imprecise level of*

5. *Editors' note: Again, this is instructional-planning theory at a very imprecise level.*
6. *Editors' note: This whole section on forms of models is purely descriptive. There is no design theory here. However, it is certainly useful for designers to understand the forms. This shows the value in*

Principle: Input/Output Variables for Each Model

A learner learns from model interaction by observing effects (or output states) that are linked to direct and indirect causes (input states and operations of the model). A limited number of model control points must be made accessible to learners through which input variables can be manipulated. The model uses these values to recompute its output states. The simulation should make clear the specific input variables that the learner will be able to influence, as well as variables that can be influenced by the environment model and other models in the simulation. The designer should also identify the recomputed values that are to be shown to the learner as model outputs. Once this list of variables is obtained, it must be continually maintained as the design evolves, because it is the connecting link between model features and learning performance requirements.

Principle: Fidelity and Resolution Levels

Models by definition lack fidelity and resolution when compared with the system that is being modeled. We call this mismatch the level of *denaturing* of the model (Gibbons, 2001; Gibbons et al., 1997). Denaturing is inevitable because media cannot represent reality faithfully, especially reality taken out of context. A representational medium therefore changes either the exactness (fidelity) of the representation or its granularity (resolution). The fidelity and resolution levels of models must match the learning need within the constraints of simulation costs.

Fidelity. Fidelity describes the degree of resemblance between reality and its model. Fidelity is clearly a factor in the transfer of learning from model situations to real situations. There are few clear guidelines on choosing fidelity levels,[7] and there is much street wisdom that is easily discredited by real evidence. There are many dimensions to fidelity, and a designer must prioritize them according to the given design problem:

- Fidelity related to learner action-taking
  - *Task fidelity*—How closely the actions taken by the learner resemble actions taken in real environments
  - *Environmental fidelity*—How closely the sensations in simulated response environments resemble the sensations produced in real environments (smells, spaces, sounds, etc.)
  - *Haptic fidelity—How realistic control actions feel as they are executed*
- Fidelity related to processing of learner actions by the model
  - *Speed/Timeliness*—The resemblance between model speed and response timing and real speed and timing
  - *Accuracy*—The accuracy of model computations compared to real outcomes
  - *Exactness*—The degree to which the operations of the modeling mechanism resemble real mechanisms
- Fidelity related to the external representation of the model's states and actions
  - *Realism*—The degree to which sensory experiences from external representations of the model conform to sensory information available in real settings

Fidelity issues are critical in instructional simulations because much information is conveyed through nonverbal channels: through visual and auditory channels for most simulations, and in the case of virtual reality environments through kinesthetic and haptic sensations. Learner processing of poorly designed, poorly synchronized, nonverbal information can produce tacit misconceptions in the learner. Reigeluth and Schwartz (1989) suggest that several factors should influence the degree of fidelity in instructional simulations: (1) potential for creating a cognitive overload; (2) ability to promote transfer of learning; (3) possible motivational appeal; and (4) cost.[8] Lathan, Tracey, Sebrechts, Clawson, & Higgins (2002) frame fidelity decisions in a way that suggests three approaches a designer might take in varying fidelity: media replacement, filter and threshold, and transformation.

Resolution. The resolution of a model can be defined as the level of detail or granularity at which it represents the reality to the user. The resolution of a model is somewhat analogous to the resolution of the computer screen: the higher the level of resolution, the more the detail can be discriminated and, up to a point, the more the information conveyed. In the case of dynamic models, resolution can be measured in terms of: (1) inputs and outputs of the model and (2) the unit of timing used as the basis for recomputing model values.

Instruction using simulations can be described as the progressive disclosure of a system's complexity (Burton, Brown, & Fischer, 1984; Gibbons et al., 1997; White & Frederiksen, 1990). Resolution that systematically changes as instruction progresses can be used as a means of evolving model complexity over time, disclosing increasing levels of system detail. Lesgold (1999) proposes that designers should incorporate into their models those elements that the learner can use in reasoning and leave alone esoteric and detailed descriptions of technical systems from which learners cannot benefit. In this view, a designer can often select the details of a model on the basis of the inputs and outputs that the learner must observe and act upon.

Principle: Model (and Problem) Growth Patterns

As experience with a set of models accumulates, performance that was once a challenge to the learner becomes easy, and the designer can escalate difficulty by supplying progressions of advanced models, advanced problems, or both.

7. Editors' note: This is to say that design theory, specifically instructional-event theory, has not yet

Progressions of increasingly complex models are described extensively by White and Frederiksen (1990, 1998). Instructional use of increasingly complex practice worlds is described by Burton et al. (1984).

2. Strategy Function: Implement Instructional Augmentations

Strategy design involves describing the context of instructional settings, social arrangements, goals, resource structures, and events supplied by the designer to augment the learner's interaction with the model. This has been referred to as the "environment of instruction." We will refer to these as the *instructional augmentation* of the model and also refer to it as the *learning companion* function.[9] Within the strategy's functional design there are many distinct subfunctions. The designer must specify:[10]

- Physical settings (classrooms, terminals)
- Social settings that include participants, role expectations, initiative rules
- The structure of instructional goals
- The assignment of goals to event blocks and the sequencing of event blocks
- The specification of event forms and classes
- The strategy (augmentation) rules for event classes
- The use of dramatic context
- The means of supplying problem-related information to learners

These constitute a set of designed site, social, goal, time, event, and product structures within which multiple agents, including learners, peers, instructors, and software can carry out a dynamic instructional conversation.

Principle: Setting and Siting of Instruction

Setting refers to the physical instructional environment and its furnishings. Siting involves the design and configuration of virtual places and their connectivity. These represent major commitments for design and infrastructure-building. Settings for the use of instructional simulations today include homes, specially designed and situation-related training suites, PDA-friendly wireless environments, and electronic classrooms and labs (Dede, 2004; Schauble & Glaser, 1996). Because of this, siting raises new issues of speed, responsiveness, presence, context, persona, and immediacy. Designs may need to include the configuration of software architectures for creating temporary networks of learning sites.

9. *Editors' note: This approach of identifying terms that other researchers have used is highly recommended for helping to build a common knowledge base.*

10. *Editors' note: In a certain sense, this is guidance (design theory) because it tells the designer*

They may also include determining communication channels among sites and the design of shared representational workspaces.

Principle: Social Context

Social context must be an important consideration in designs, because the increasing use of multiuser simulations forces the designer to consider the implications of the societies they automatically create. A social context causes the designer to define the forms and patterns of communication that can take place among learners, instructors, and instructional agents. In some cases, such as the training of team activity, the social context is determined by the nature of the task. In other cases, social context may be tailored by the designer specifically to represent other values, such as the need to overcome cultural barriers (Cole, Engestrom, & Vasquez, 1997), the need to learn cooperation skills, or the need to form or take advantage of an existing community of practice (Lave & Wenger, 1991). An excellent summary of social issues is given by Lehrer and Schauble (2006). Two aspects of the social context are of special importance to simulation design: participant roles and patterns of initiative-sharing.

The learner's role in a simulation-centered activity is dynamic, changing sometimes moment-by-moment. It involves the sharing of initiative with peers, instructors, and instructional agents (Colella et al., 2001; Gibbons et al., 1997; Johnson, Rickel, & Lester, 2000; Resnick, 1997; Rieber, 2004). Roles must shift as instruction progresses, placing increasingly greater responsibility for learning tasks and initiatives on the learner.[11] Instructional support must fade, and learner initiative must increase. Lave and Wenger (1991) suggest that specific role assignments are not as critical as the fact that role expectations come to exist within a community and that they be known to, and accepted by, both learner and instructor. Some details of role assignments, however, do appear to make a difference. Problem-based learning (Barrows, 1998; Barrows & Tamblyn, 1980) and reciprocal teaching (Brown & Palincsar, 1989) depend on learner acceptance and execution of a precisely defined role. Failure to carry out the role can result in failure of the strategy. In both cases, learners participate in modeling expert performance within a context of scaffolding that gradually fades.

Rules for initiative sharing become more important as the role of the learner grows. Instructional simulations can offer the learner opportunities not only for responding, but also for taking the initiative in areas of strategy implementation, nonsequential event control, or spontaneous messaging and conversation. Microworlds in most cases *require* that the learner take the initiative for one or more of them. Space does not allow for detailing of the dimensions of strategy in which initiative can be shared with, negotiated with, or placed on the learner and the different patterns of initiative-taking that can be constructed from these dimensions (Fox, 1993). A simulation designer, however, must deliberately plan

11. *Editor's note: This method, helping learners to assume greater responsibility for their own*

the possible initiative states, their patterns of changing, and the mechanism for negotiation of the change.

Principle: Instructional Goals

Traditional views of instructional objectives are challenged by the nature of simulations, whose instructional goals simultaneously may include subject-matter goals, problem solving goals, and learning-to-learn goals (Collins, Brown, & Newman, 1989). The fragmented, content-centered goals of the past conflict with performance-centered, integrated (skills, attitudes, values) goals required for simulation design.

The scope of action within a simulated environment is so broad and the integration of goals so extensive that to define instructional goals as a list of separate statements is impractical. A more useful method is to describe the properties of the subject model(s) and to define the desired range of performances learners should be able to demonstrate with respect to models (Gibbons, Nelson, & Richards 2000a, 2000b, 2000c).[12] This form of instructional goal can be generally expressed as:

Within <environment>
one or more <actor>
executes <performance>
using <tool>
affecting <system or system process>
to produce <outcome, artifact>
having <properties, qualities>.

During design this can be used as a beginning point for determining more detailed requirements through deduction for the environment, cause-effect, and expert performance models.

The global objective is formed in terms of the ranges of values that can be used in each of the variable positions in these statements. This creates a bridge between analysis and design processes by identifying goals in terms of the three types of models that must be created, and at the same time it defines a *simulation scope*: the outer bounds of what must be simulated (Gibbons et al., 1997).

Principle: Specific Instructional Event Goals and Goal Progressions

Individual, transient, momentary *instructional scopes*—instructional goals—can be framed against the backdrop of this comprehensive simulation scope and related to specific time-bound instructional events. These consist of statements

12. *Editors' note: This is curriculum design-theory, appropriately integrated with instructional-event theory and instructional-planning theory.*

that momentarily restrict the ranges of the variables described above to within particular values. Individual instructional scopes defined in this way can be used to form progressions of models, trajectories of performance expectation, and performance assessment points (White & Frederiksen, 1990). The *work model* concept proposed by Bunderson, Gibbons, Olsen, & Kearsley (1981) and the *elaboration* concept proposed by Reigeluth (1999) give useful terminology, process, and product guidelines to the designer.

Many principles can be used for sequencing the progression of instructional scopes (goals). These can include ordering for rapid coverage, ordering for maximum coverage in minimum time, ordering in step sizes matched to the learner's rate of progress, ordering according to cognitive load, and ordering to maximize exposure to the widest range of problems.[13] A general sequencing principle for instruction in skills is described by van Merriënboer (1997).

Instructional simulations may: (1) make use of a fixed sequence of problems; (2) provide dynamic computation of problem sequences; (3) allow learner selection of problem sequences; or (4) permit the learner to form the instructional goal. Whichever of these is used, the selection of problems and their sequence is critical to maintaining and adjusting levels of engagement and challenge during simulation instruction. Vygotsky's concept of the zone of proximal development (1978) implies a sequencing principle based on learner readiness.

Selection of goal sequences also implies that problems can be scaled with precision along multiple dimensions of difficulty and indexed according to those dimensions. Studies of domain structure by Bunderson, Wiley, and McBride (chapter 15, this volume), Strong-Krause (2001), and McBride (2005) suggest that this kind of indexing may be possible, but the process of scaling is complex at present and requires very large databases.

Principle: Standard Event Forms

Our most common shared design language terms describing event forms include *lesson, lab session, recitation session*, and so forth. These terms are generic and do not have sufficient precision to guide a simulation design without additional specification. A simulation designer should define a small number of event forms that can be repeated using different specific content. Event forms define the granularity of instructional elements that can be sequenced. Events during an instructional simulation may occur in unpredictable orders, depending on the flow of responses that make up the learner's side of an instructional conversation. Event forms define the patterns this conversation can employ. A simulation experience in this sense is an emergent phenomenon.

13. *Editors' note: While this describes several methods for sequencing, to be design theory it would need to indicate when to use each method (situationalities), and it would also be helpful to provide more precision—details about how to use each method.*

Principle: Augmentation Types and Rules

Augmentation in an instructional simulation supplies that part of the instructional experience (including information and interactions) not generated by the model. Augmentations may consist of many different functions, including: providing reminders, suggesting analogical or metaphorical associations, directing attention, performing parts of a difficult task for the learner, giving suggestions or hints, providing explanations, demonstrating, delivering assessments, evaluating or guiding self-evaluation, diagnosing causes, prescribing action, posing questions or challenges, and many other functions. In instructional simulations augmentations work in parallel and in synchronization with model functions, providing information and interaction that is relevant to current learning tasks.

Augmentations support the learner's actions of model observation, model interpretation, decision making, organization and processing of new information, recalling, acting, judging, valuing, and evaluating. The model generates information that can be made available to the learner about model operations; augmentations supply additional information that helps the learner to use model information to learn.

Augmentations chosen for inclusion in a design make up the *learning companion* functions for the simulation, whether they are human or computer administered. The learning companion is especially important in helping the learner to become self-aware and self-directed with respect to learning processes: to realize that deliberate metacognition can improve learning-directed initiatives (Collins et al., 1989; Elen & Clarebout, 2005). The plan for augmentation functions must specify a set of primitives to be used in augmentation events and a set of rules describing the conditions under which augmentation events occur.[14] The augmentation plan should provide for a staged diminishment of augmentation that encourages increasing learner reliance on personal decision-making, judgment, and initiative.[15]

Principle: Dramatic Context

Instructional problems can be posed within the context of hypothetical situations occurring within simulated, fictional, or imaginary performance settings. Scenarios using narrative techniques can create an imaginary world that involves symbolic personas in representative dramatic roles that have goals and encounter and overcome obstacles to reach them (Schank & Berman, 2002). These can be completely realistic or completely imaginary (Barab, Thomas, Dodge, Carteaux, & Tuzun, 2005; CTGV, 1992; Metcalf-Jackson, Krajcik, & Soloway, 2000).

14. *Editors' note: This defines a set of situationalities for selecting the different types of augmentations (methods). However, the level of precision is such that specific situationalities for each augmentation are not provided.*

15. *Editors' note: These last two sentences provide some design theory (primarily instructional-*

The use of a scenario or story setting for the problem implies the need to design the problem solving environment, which may include something as uncomplicated as a simple interface or as complex as a world made up of connected information-providing locations which the learner navigates to obtain critical information (Gibbons et al., 1997). The latter implies the need for the designer to create an integrated information and world-location structure.

Principle: Problem Information Structure

The *information structure* for an individual learning problem includes the set of data or facts required to pose it (the problem statement) and solve it (the solving resources). The information includes data to be used by the model specifying the problem environment, initial problem states, desired terminal states, and important intermediate solution states. The information structure is necessary to stage a problem, judge solutions, and provide augmentations like coaching and feedback. This information can be captured in several forms: as a database, a set of production (if…then) rules, or as a computable mathematical model. It is significant that the activities of problem-based learning "tutors" in dispensing problem information must be clearly specified, as well as the information itself, for the instructional method to work as expected (Barrows, 1988).

3. Control Function: Provide User Controls

The control function of a simulation design describes the means by which a learner can convey messages that influence the unfolding of the content, strategy, or other dynamic elements of the experience. The design of control systems can be challenging because learner actions take place within a dynamic context and must make use of a history of previous information and control exchanges.

Principle: Controls for Each Main Function of the Design

Control systems consist of sets of special-purpose controls that serve needs related to several simulation functions: (1) controls that allow the learner to act upon the model; (2) controls that adjust patterns of augmentation; (3) controls that adjust the representation of the model or the viewpoint from which the learner can observe the model; and (4) controls over personal data reporting for monitoring outcomes, performance, progress, trends, history, and scheduling. Goals for simulation and microworld control design include: maintaining simplicity, keeping controls transparent, determining priorities among controls, and planning conditions under which controls may be made available or withheld.

Regarding controls that act upon the model (#1 above), Crawford (2003) proposes that "the first rule of all interactivity design is to start with the verbs"

"During this process you are specifying what the user does, *not* what the screen looks like or what the program will do or how it will work" (p. 94, emphasis in the original). The design of multiuser control systems adds complexity, but it is also approachable using Crawford's linguistic metaphor. New actuation devices make available new terms in the language of controls, including body-part movement, gesturing, sounds, balance displacement, spoken words, whole-body displacement (locomotion), eye movement, facial expression, panel touch, and others (see Moggridge, 2007). These devices supply "verbs" with which a learner can communicate control to the model.

Augmentation controls (no. 2) are for expressing the strategic decisions the designer makes accessible to the learner. In microworld design, this includes controls that modify levels of support or request help. In the design of microworlds the design of control systems is especially important because learners are asked to construct models and operate them to conduct experiments. Designing an efficient and intuitive control set for model construction and operation represents an additional challenge. In terms of strategy, Crawford's nouns are augmentation elements that can be acted upon and his verbs express actions that can be applied to them.

Controls over representation (no. 3) must allow the learner to select elements of the representation to be visible or hidden and must allow the learner to select a vantage point for viewing the environment and its contents. They must also allow the learner to navigate the visible and virtual world created by the model, its augmentations, and all of the data-containing locations associated with them. Representations of models must change state as the variable values of the models themselves change. Moreover, models can be viewed from different perspectives or in different manifestation modes (schematic, realistic, surface, inner workings, etc.). Crawford's nouns and verbs in this instance are representational elements or effects that can be operated upon and operations that can be applied to them.

Finally, the designer must make information available that the learner can use in making decisions. The data reporting function can give the learner much information about progress, status, and scheduling. Data reporting controls (no. 4) must be designed that give the learner access to this information. In this case, Crawford's nouns and verbs consist of data elements and operations upon them.

4. Messaging Function: Generate Message Units

Message structures are the basic building blocks of simulation-to-learner communication. An interactive conversation with a model takes place at the level of numerous individual messages that pass between the learner and the simulation. Message structures are also the vehicle by which other functions of the simulation—the augmentations, the controls, and the data management—express themselves to the learner. Message structure is an important design feature that are used by *live* instructors and tutors during the conversational process (Barrows, 1998).

The messaging language created by the designer is the complement of the control language described earlier. Together they define the quality of the inbound (through controls) and outbound (through messages) communications that can take place. They constitute the heart—invisible and abstract—of the design of a learning interface.

Principle: Message Elements

Messaging can be implemented using the fully adaptive conversational capabilities of a human tutor or can be simplified and routinized for implementation on a computer. Computerizing messaging constrains a design according to Fox (1993), and adjustments like the following may be required:

- Interruptions by the learner should be possible
- Thing reference (pointing) and object sharing should be more formalized
- Silences should be flagged with intent
- Communication of backchannel cues (emotional states, body language, attitude) should be facilitated
- Multiple sequential messages should be possible from the same speaker without a break (e.g., musings "aloud")
- Short delays in correction might be deliberately used to signal to the student the need to think and respond again
- Ways should be found to make the learner's process actions (thinking) known to the tutor.

All of these have the effect of replacing in technology-based tutoring some of the subtle cues that are lost from human-human tutoring.

Principle: Approaches for Structuring Messages

Several theorists and researchers have described systems for: (1) structuring conversational instruction or (2) analyzing the message structure of instructional conversations (for instance, see Horn, 1997; Merrill, 1994; Sawyer, 2006; Simon & Boyer, 1974; Smith & Meux, 1970).[16]

Principle: Execution-Time Construction of Messages

Messages can be constructed from a set of primitives at the time of instruction, in response to a momentary strategic need. For example, most designers provide

16. *Editors' note: These different approaches should be accompanied by situationalities that*

Table 9.1 Message Elements that might be Included in a Typical Feedback Message Following a Learner Action

Message element	Message intent
<right/wrong notification>	That was not an appropriate choice at this point.
<the learner's response>	You increased the temperature to <value>.
<expected answer>	You should have decreased the temperature or increased the pressure in the deposition chamber.
<why correct/expected>	This would lower condensation index below <value> and initiate deposition.
<correct principle>	Deposition cannot take place when the condensation index falls in the area above the Critical Value line shown in red on the Condensation Index chart.

some type of feedback during instruction to follow learner responses: either to confirm or to correct. These messages are usually a composite of smaller message elements like those shown in Table 9.1. A designer's personal design philosophy might choose a different set of message elements and a different way of combining them, but the principle of message generativity would still apply. Therefore, the example in Table 9.1 shows only one of many possible combinations.

Drake, Mills, Lawless, Curry, and Merrill (1998) use this method to generate all of the necessary presentation-, demonstration-, and practice-related messages for a simulation that teaches the operation of a canal lock system. One advantage of message generation is that the composition rule can be made context-sensitive, so that messages can strategically increase or decrease the degree of support given. This is a requirement for truly adaptive instruction.

5. Representation Function: Generate and Assemble Representation Elements

New technologies for computer-generated representation are developing rapidly. One of the forces behind this rapid advance has been the need for better visualization systems for simulations. Today the creation of two- and three-dimensional worlds through representation is an important area of innovation.

The representation function of a simulation design is the most visible and tangible. Its design involves all sensory elements of the simulation experience—sights, sounds, tactile sensations, and kinesthetic sensations. The representation function design describes all sensory experiences that will be staged and how they will be integrated and synchronized. All of the structures described to this point for content, strategy, controls, and messages are abstractions and become visible only through representation design. Therefore, representation is the bridge that links abstract design elements with specific symbolic media elements.

The representation of a simulation can be dynamic and can be constantly changing. Simulation representations are driven by and reflect current model, strategy, control, and messaging function states. Microworld designers must provide the visual and other sensory elements of an interesting and intuitive model-building work space as well as a model-running work space.

The representation function is driven by messages. It maps message elements to representation elements. Computer-based simulation representations can be created in two main ways: (1) from static, premade representation elements drawn together to satisfy a momentary need; or (2) through data-generated graphical forms whose position and motion are computed. Some simulations combine these methods.

Modularity of the representation functions can allow the style of representations to be changed without disturbing the workings of other functions. Zen Garden (Mezzoblue.com) for instance, gives an example of how the innovation of cascading style sheets (CSS) allows identical message elements to be given multiple, diverse visual surfaces. CSS is a tool for modularizing some of the functions of representation.

Principle: The Representation Refresh Cycle

The most important task of representation in a simulation is to make the operations of an otherwise invisible model visible (Collins et al., 1989; Gibbons, 2001). The representation of model states must change according to the cycles of the model.

Principle: View Perspectives and Styles

A model can be viewed from different vantage points. Moreover, it can be viewed in different stylized forms. Designers should determine whether there is advantage in giving the learner alternative viewing positions in real or metaphorical space. Adequate model representation may also require multiple styles, such as schematic, conceptual, or realistic representations. Views of a cause–effect model being operated upon may include a realistic view of controls and a symbolic view of the internal effects of control manipulations.

Principle: Message-to-Representation Mapping Rules

Every message that can be formed must be given representation. In some cases, messages and representations are mapped one-to-one, meaning that for a single message element there is a single stored or generated representation. In some cases, however, a designer may specify that a single message be given multiple representations simultaneously, or that multiple messages be combined for representation by a single representation element. For example, single message unit (<correct answer>) giving feedback following a learner's response, may map to multiple simultaneous representation elements: the learner's response turns green, a success tone is heard, a text message appears. The value of thinking in terms of message units and representation elements separately becomes clear in this case: a single rule can be expressed to describe the relation between message units and representation elements. Similarly, a rule can be expressed that causes a particular combination of message units to trigger a single representation.

Principle: Display Assembly and Coordination Rules

The coordination of a simulation display involves synchronization. Mayer (2001) has described the importance of timing in the appearance of media events related to the same message unit. It is significant that some of the most popular development software is based on a timeline structure for the coordination of media representations. Future tool versions may allow the designer to coordinate media events with message units, strategic events, and model events as well. Rules for the execution of the continuously changing displays must also be designed.

Principle: Rules for Manipulating Time and Space

Instructional simulations telescope time and space; they speed up processes that are slow and retard processes that are fast so that they can be observed. Simulations are time machines that interleave learner actions with model actions in warped time. Repeated practice under simulated conditions can concentrate greater amounts of practice into instructional time. Altered time can contribute to the efficiency of learning, but in order to prepare learner performances for the real world, timing must eventually tend toward real-world speeds as the learner's ability permits.

Space is likewise manipulated by simulations. Virtually all simulations create action spaces larger than the simulation itself requires. Simulations compress space by allowing learners to zoom (in and out) and navigate imaginary spaces. A simulation designer should thoughtfully address maximization of time and space.

Principle: Rules for Representation Trace

A simulation's effectiveness depends in part on the ability of the learner to detect changes in represented model states in response to actions on the model. However, information of even greater value lies in visual vectors that we will call *traces* that represent *trends* of change. Using a representation trace, a learner can pose and answer questions using the information history the trace provides. A trace preserves a record of interactions among input variable settings, internal model forces, and output variables.

A simulation designer must consider how to choose traces to make visible, how to allow the learner to construct traces, how to facilitate navigation through

comparison are facilitated, how to highlight important elements of traces, and how to help students to see patterns within traces and interpret them. Examples of trace are provided by Edward Tufte (1997), Wurman (1997) and Edgerton (MIT-Libraries, n.d.).

6. Media-Logic Function: Execute Representations and Computations

The media-logic function executes representations and carries out the logical operations that allow simulation events to occur. This can also include calculations and data gathering. Not all simulations are computer-based, but all simulations require that the media-logic functions be carried out. The media-logic function of a design enacts the conceptual structures from all of the other functional areas of the design, giving them dynamism and synchronizing their operations. According to Gibbons et al. (2001), "this is the place where the designer's abstract instructional constructs and concrete logic constructs [of] the development tool come together."

The media-logic design must incorporate: (1) instructions for executing the models; (2) instructions for coordinated execution of the augmentations; (3) instructions for accepting learner responses and control actions; (4) instructions for forming messages; (5) instructions for providing representations to the learner; and (6) instructions for collecting, storing, processing, and displaying personal data the learner can use in decision making about learning.

Media-logic for human-based simulations can be challenging, because human tutors require guidelines that are robust to individual differences while preserving the discipline of the design. Computer-based designers must take into account available hardware and software infrastructure; distribution methods; location of processing; processing and networking loads; facilities for delivery; availability of development skills, tools, and processes; availability of expertise in subject-matter; life cycle software plans; and security issues.

Modularization of the media-logic function is an important consideration for long-term maintainability of a simulation. Baldwin and Clark (2000) describe how modularization of computer operating system software designs by IBM set the stage for a revolution in the economics of both computers and software. It has been proposed that the monolithic nature of earlier systems has been an obstacle to integration with other advanced simulation-centered tutoring components. A modular architecture for simulation design can provide advantages from component reuse, improved maintainability, rapid adaptation, and support for a range of tutorial applications as shown by Munro, Surmon, and Pizzini (2006).

7. Data Management Function: Manage Data Resulting from Interactions

The data management function of an instructional simulation supports and from the existing concept of computer-managed instruction (CMI) by adding functions that incorporate recent actions by the learner into decision making in which the learner participates. The data management function supports the learner's negotiation of instructional event sequences; it records large amounts of data at a (potentially) high degree of granularity, it stores the data, analyzes it, interprets it, and provides reports to the learner that can be used to monitor progress and make instructional decisions.

In the principle statements below, the wording suggests that the data collection and analysis functions are assumed to be computerized. However, the principles pertain to noncomputerized simulations as well.

Principle: Sampling Points and Variables

Data collection points are defined in terms of the computational cycle of the models, which will be either event-driven or time-driven. At data collection points, values are recorded for specific variables and stored for analysis and interpretation—some immediate, and some later.

Principle: Interpretation Variables and Rules

Interpretation variables accumulate the values that result from interpretation of raw interaction data. They store interpreted values that are determined by rules and formulas supplied by the designer. Interpretation variables may themselves be subjected to further analysis aimed at reaching conclusions about the state of the learner's knowledge, the attitude of the learner, and the growing skill levels of the learner.

Principle: Data Gathering Framework

The designer must specify the timing of the analysis and interpretation cycle events. These events use interpretation variable values in making instructional decisions. The data gathering framework describes these operations and the storage and reporting of data to the learner for decision making as well.

Principle: Reporting

The simulation designer should specify the points at which information is made available to the learner. These points should at least coincide with decision making points at which the learner is given choices. Learner progress and performance data may be shown to take advantage of the opportunity to teach the learner how to use data reports in making learning decisions.

Conclusion

Simulations are particularly powerful and effective because they enhance student vigilance and scanning skills, enhance student integration of skills in varied performance contexts, adjust to varied learning rates through dynamic performance scopes, and help learners see patterns over time in dynamic systems. We have described a number of specific principles which can guide design, and while space limitations have kept us from including *all* of the possible guidance principles, we believe that this chapter helps identify the current state of knowledge for a common knowledge base for the design of instructional simulations.

References

Alessi, S. M., & Trollip, S. R. (1991). *Multimedia for learning* (3rd ed.). Boston, MA: Allyn & Bacon.

Atkinson, R. C., & Wilson, H. A. (Eds.). (1969). *Computer-assisted instruction: A book of readings.* New York: Academic.

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: Vol. 1. The power of modularity.* Cambridge, MA: MIT Press.

Barab, S., Thomas, M., Dodge, T., Carteaux, R., & Tuzun, H. (2005). Making learning fun: Quest Atlantis, a game without guns. *Educational Technology Research and Development, 53*(1), 86–107.

Barrows, H. S. (1988). *The tutorial process* (Rev. ed.). Springfield: Southern Illinois University School of Medicine.

Barrows, H. S. (1998). The essentials of problem-based learning. *Journal of Dental Education, 62*(9), 630–633.

Barrows, H. S., & Tamblyn, R. M. (1980). *Problem-based learning: An approach to medical education.* New York: Springer.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (2000). *How people learn: Brain, mind, experience, and school* (Rev. ed.). Washington, D.C.: National Academy Press.

Brown, A. L., & Palincsar, A. S. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser.* Hillsdale, NJ: Erlbaum.

Bunderson, C. V., Gibbons, A. S., Olsen, J. B., & Kearsley, G. P. (1981). Work models: Beyond instructional objectives. *Instructional Science, 10,* 205–215.

Burton, R. R., Brown, J. S., & Fischer, G. (1984). Skiing as a model of instruction. In B. Rogoff & J. Lave (Eds.), *Everyday cognition: Its development in social context.* Cambridge, MA: Harvard University Press.

CTGV. (1992). The Jasper experiment: An exploration of issues in learning and instructional design. *Educational Technology Research and Development, 40*(1), 5–34.

Cole, M., Engestrom, Y., & Vasquez, O. (1997). *Mind, culture, activity: The seminal papers from the Laboratory of Comparative Human Cognition.* Cambridge, UK: Cambridge University Press.

Colella, V., Klopfer, E., & Resnick, M. (2001). *Adventures in modeling: Exploring complex, dynamic systems with StarLogo.* New York: Teachers College Press.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser.* Hillsdale, NJ: Erlbaum.

Crawford, C. (2003). *The art of interactive design.* San Francisco: No Starch.

de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*(3), 179–201.

Dede, C. (2004). *Emerging technologies that enable distributed-learning communities.* Paper presented as the Keynote speech to the Conference of the Association for Educational Communications and Technology.

Drake, L., Mills, R., Lawless, K., Curry, J., & Merrill, M. D. (1998). *The role of explanations in learning environments.* Paper presented at the Annual Meeting of the American Educational Research Association, San Diego.

Elen, J., & Clarebout, G. (2005). Support in learning environments: Touching the limits of instructional design. *Educational Technology, 45*(5), 44–47.

Fox, B. A. (1993). *The human tutorial dialogue project: Issues in the design of instructional systems.* Hillsdale, NJ: Erlbaum.

Gibbons, A. (2001). Model-centered instruction. *Journal of Structural Learning and Intelligent Systems, 14*(4), 511–540.

Gibbons, A. S., Fairweather, P. G., Anderson, T., & Merrill, M. D. (1997). Simulation and computer-based instruction: A future view. In C. R. Dills & A. J. Romiszowski (Eds.), *Instructional development paradigms* (pp. 769–805). Englewood Cliffs, NJ: Educational Technology.

Gibbons, A. S., Lawless, K. A., Anderson, T. A., & Duffin, J. R. (2001). The web and model-centered instruction. In B. R. Khan (Ed.), *Web-based training.* Englewood Cliffs, NJ: Educational Technology.

Gibbons, A., Nelson, J., & Richards, R. (2000a). The nature and origin of instructional objects. In D. Wiley (Ed.), *The instructional use of learning objects.* Bloomington, IN: Association for Educational Communications and Technology.

Gibbons, A. S., Nelson, J., & Richards, R. (2000b). *Model-centered analysis process (MCAP): A pre-design analysis methodology* (White Paper). Idaho Falls, ID: Center for Human-Systems Simulation, Idaho National Engineering and Environmental Laboratory (DOE).

Gibbons, A. S., Nelson, J., & Richards, R. (2000c). *Theoretical and practical requirements for a system of pre-design analysis: State of the art of pre-design analysis* (White Paper). Idaho Falls, ID: Center for Human-Systems Simulation, Idaho National Engineering and Environmental Laboratory (DOE).

Gredler, M. E. (2004). Games and simulations and their relation to learning. In D. H. Jonassen (Ed.), *Handbook of research on educational communications and technology* (2nd ed., pp. 571–583). Mahwah, NJ: Erlbaum.

Horn, R. E. (1997). Structured writing as a paradigm. In C. R. Dills, A. Romiszowski, & A. J. (Ed.), *Instructional development paradigms.* Englewood Cliffs, NJ: Educational Technology.

Johnson, W. L., Rickel, J. W., & Lester, J. C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education, 11,* 47–78.

Kafai, Y. B. (2006). Constructionism. In R. K. Sawyer (Ed.), *Cambridge handbook of the learning sciences* (pp. 35–46). Cambridge, UK: Cambridge University Press.

Lathan, C. E., Tracey, M. R., Sebrechts, M. M., Clawson, D. M., & Higgins, G. A. (2002). Using virtual environments as training simulators: Measuring transfer. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 403–414). Mahwah, NJ: Erlbaum.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral practice.* Cambridge, UK: Cambridge University Press.

Lehrer, R., & Schauble, L. (2006). Cultivating model-based reasoning in science education. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences.* Cambridge, UK: Cambridge University Press.

Lesgold, A. (1999, July, 1999). *Intelligent learning environments for technical training: Lessons learned.* Paper presented at the Advanced Training Technologies and Learning Environments, Langley Research Center, Hampton, VA.

Mayer, R. E. (2001). *Multimedia learning.* Cambridge, UK: Cambridge University Press.

McBride, R. (2005). *Toward a domain theory of fluent oral reading with expression.* Provo, UT: Brigham Young University.

Merrill, M. D. (1994). The descriptive component display theory. In M. D. Merrill & D. G. Twitchell (Eds.), *Instructional design theory.* Englewood Cliffs, NJ: Educational Technology.

Metcalf-Jackson, S., Krajcik, J. S., & Soloway, E. (2000). Model-It: A design retrospective. In M. Jacobson & R. B. Kozma (Eds.), *Innovations in science and mathematics education: Advanced designs for technologies and learning.* Mahwah, NJ: Erlbaum.

Mezzoblue.com. (2006, December 12). Zen garden. Retrieved from http://www.mezzoblue.com/zengarden/alldesigns/

Milrad, M., Spector, J. M., & Davidsen, P. I. (2002). Model facilitated learning. In S. Naidu (Ed.), *Learning and teaching with technology: Principles and practices.* London: Kogan Page.

MIT-Libraries. (n.d.). Harold Eugene Edgerton, 1903–1990: Papers, 1889–1990; Manuscript Collection—MC 25. from http://libraries.mit.edu/archives/collections-mc/mc25/index.html#collection

[illegible] (2007). *Designing interactions.* Cambridge, MA: MIT Press.

Munro, A., Breaux, R., Paltrey, J., & Sheldon, B. (2002). Cognitive aspects of virtual environment design. In K. M. Stanney (Ed.), *Handbook of virtual environments* (pp. 415–434). Mahwah, NJ: Erlbaum.

Munro, A., Surmon, D. & Pizzini, Q. (2006). Teaching procedural knowledge in distance learning environments. In R. Perez & H. O'Neal (Eds.), *Web-based learning: Theory, research, and practice.* Mahwah, NJ: Erlbaum.

Papert, S. (1993). *Mindstorms: Children, computers, and powerful ideas* (2nd ed.). New York: Basic.

Reigeluth, C. M. (1999). The elaboration theory: Guidance for scope and sequence decisions. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: Vol. 2. A new paradigm of instructional theory* . Mahwah, NJ: Erlbaum.

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.

Resnick, M. (1997). *Turtles, termites, and traffic jams: Explorations in massively parallel microworlds.* Cambridge, MA: MIT Press.

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. (2004). Microworlds. In D. H. Jonassen (Ed.), *Handbook of research for educational communications and technology* (2nd ed., pp. 583–603). Mahwah, NJ: Erlbaum.

Sawyer, R. K. (2006). Analyzing collaborative discourse. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences.* Cambridge, UK: Cambridge University Press.

Schank, R. C., & Berman, T. R. (2002). The pervasive role of stories in knowledge and action. In M. C. Green, J. J. Strange, & T. C. Brock (Eds.), *Narrative impact: Social and cognitive foundations.* Mahwah, NJ: Erlbaum.

Schauble, L., & Glaser, R. (1996). *Innovations in learning: New environments for education.* Mahwah, NJ: Erlbaum.

Simon, A., & Boyer, E. G. (1974). *Mirrors for behavior III: An anthology of observation instruments.* Wyncote, PA: Communication Materials Center in Cooperation with Humanizing Learning Program, Research for Better Schools.

Smith, B. O., & Meux, M. O. (1970). *A study of the logic of teaching.* Urbana: University of Illinois Press.

Strong-Krause, D. (2001). English as a second language speaking ability: A study in domain theory development. *Dissertation Abstracts International,* DAI-A 61/12, 4746.

Tufte, E. R. (1997). *Visual explanations: Images, quantities, evidence, and narrative.* Cheshire, CT: Graphics.

van Merrienboer, J. (1997). *Training complex cognitive skills: A four-component instructional design model for technical training.* Englewood Cliffs, NJ: Educational Technology Publications.

Vygotsky, L. S. (1978). *Mind and society: The development of higher mental processes.* Cambridge, MA: Harvard University Press.

White, B. Y., & Frederiksen, J. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 24,* 99–157.

White, B. Y., & Frederiksen, J. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction, 16*(1), 3–118.

Wurman, R. S. (1997). *Information architects.* New York: Graphis.