Eight Views of Instructional Design and What They Should Mean to Instructional Designers

Andrew S. Gibbons

Keywords Design process • Design methods • Design studies • Architecture • Professional

Introduction

The chapters in this volume are evidence of a new drive toward more robust and valid descriptions of design: better descriptions of design for novices and advanced concepts and methods for experienced designers.

A number of scholars are revitalizing the discussion of design within instructional technology, viewing design from different perspectives. Jonassen (2008) asserts the problem-solving nature of design. Rowland (2008) describes how we learn by designing. Bannan-Ritland (2003) places instructional design in context with design research in other fields. Bichelmeyer (2003), Reigeluth (1999), Reigeluth and Carr-Chellman (2009), and Yanchar and his colleagues (Yanchar, South, Williams, Allen, & Wilson, 2010) place stress on the nature of theory and its relation to design. Hokanson and Miller (2009) examine the multiple roles of the designer. Parrish (2005, 2006) explores the aesthetic nature of designing and of the designed artifact. Gibbons and Rogers (2009) propose how an architecture of designed things applies to instructional design. Wilson (2005) reexamines the practice of design. This energetic discussion of design echoes an interest in design which has been rising for decades outside of the instructional technology field, producing a rich literature that informs our own.

The backdrop to this discussion is a tradition of over 50 years of reliance on increasingly simplified descriptions of design in the form of design models. Smith

A.S. Gibbons (✉)
Brigham Young University, Provo, UT 84604, USA
e-mail: andy_gibbons@byu.edu

B. Hokanson and A. Gibbons (eds.), *Design in Educational Technology*, Educational Communications and Technology: Issues and Innovations 1,
DOI 10.1007/978-3-319-00927-8_2,

and Boling (2009) review the assumptions and misconceptions of design models that have evolved over that period. There is room to question whether the notion of a design model adequately describes what we know about design (Gibbons, Boling, & Smith, 2013; Gibbons & Yanchar, 2010). Gibbons and Yanchar (2010) identify a wide range of topics that would be included in a more robust description of design.

Placing Instructional Design in Perspective

Some of these issues can be addressed by viewing instructional design from different perspectives of scale that include its historical context, the environment of designing, the nature of the thing being designed, the thinking processes of the designer, and the conceptual tools the designer wields during design. Figure 1 illustrates eight different views of design that describe it from multiple scale perspectives. Describing these views bridges the conceptual and practical worlds of design at different levels of scale, yielding new questions for exploration.

Organizational View

The first view of instructional design describes the relationship of the designer to the larger organization. Instructional design consumes time, money, and resources. Making quality instructional products requires specialized skills, equipment, and collaboration among members of a team. For this reason, instructional design is

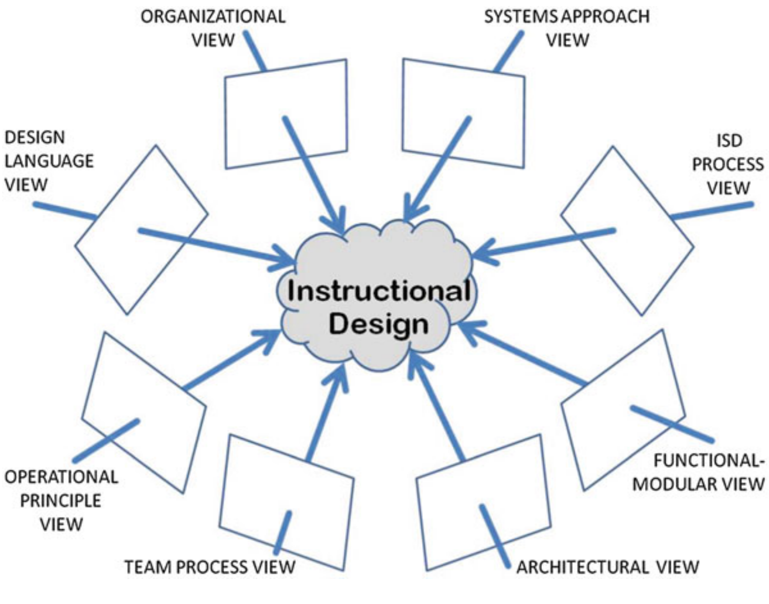

Fig. 1 Design viewed from many perspectives from Gibbons (2013)

normally carried out by a team under organizational sponsorship—a business, a school, the military, a government organization, or a client. When an organization considers funding and staffing a design project, it usually asks the designer what value it can expect as a return on its investment. Instructional designers, therefore, are becoming aware of the value they add to the organization.

The placement of instructional designers within organizations is changing. In the traditional pattern, designers operated in relative isolation within a training department, separated from the operational and administrative functions of the organization, disconnected from everyday operational concerns.

In this scenario the training department was an appendage to, and not really part of, the organizational fabric. Designers were told what to make—not consulted and not included in key organizational decisions. When there was a downturn in the organization's fortunes, the training department was a first candidate for cutting. The training function was most often placed under separate, nonoperational management.

A new pattern is evolving. Organizations are increasingly realizing that training is an important part of their product: something that enhances its value to customers. Organizations realize that there is value in training that supports the product or service, making it easier to use. The value of a workforce that is well trained in product skills and customer relations is being recognized, so organizations are using training to unify their workforce and focus their energies by increasing collaboration among employees. The value of these collaborations for creative problem solving is another value that does not escape organizational notice. As a by-product, organizations are seeing that training and education can help to create and maintain corporate morale through employee buy-in. In short, organizations are recognizing the function of training in creating organizational culture.

Training is increasingly viewed as a fundamental process of a competitive organization: a function essential to the organization's growth and adaptation within a changing environment. Designers must see themselves as creators of value within the organization. To do that, they must understand the values of the organization and how their products and services support them. The designer must understand how value is measured to the organization and what elements of a design lead to value. Designers must sometimes make calculated trade-offs between practical concerns and theoretical issues. The training designers receive must prepare them for this.

Organizations are interested in designers who can speak their language and who understand the rules of the new knowledge economy and the new information-based organization (Kahin & Foray, 2006; see also Drucker, 1989). Research on the value added of the designer and the new role of the designer within the new organization is badly needed.

Systems Approach View

A second view of instructional design—the systems approach view—is historical (Ramo & St. Claire, 1998). The practice of formal instructional design became a topic during World War I, but it became an imperative during World War II.

With the emergence of complex man-machine systems, time needed for training increased just at the time when it was becoming more scarce. Efficiency became the goal of training, and the systems approach became the means of designing training to reach that goal.

The systems approach is a problem-solving process for highly complex problems. It is not a single procedure but a set of problem-solving tools and techniques used by multidisciplinary teams of scientists and engineers. There is no set order, but as problems are solved, new problems appear, demanding the selection of appropriate tools. The first problem attacked by a team using a systems approach is to ascertain the real problem, which involves in most cases gathering large amounts of data for extensive data analysis. The systems approach is difficult to describe because it is a family of problem-solving methods rather than a formula.

The systems approach involves solving a complex problem viewed in terms of multiple complex interacting systems. The problem is broken down into independent solvable subproblems that involve the coordinated behavior of multiple subsystems. Analyzing problems and testing solutions normally involve quantification of variables.

In the systems approach, a multidisciplinary team consisting of both scientists and engineers works toward a solution. Decisions are based on the best data obtainable, using a wide range of problem-solving methods. Methods are selected according to problem status, not an orderly process. Multiple alternative solutions are explored and evaluated on the basis of multiple, sometimes conflicting, criteria that account for the needs of many stakeholders. System modeling and simulation are often used to test solutions.

Innovation is the goal because problems solved often have few precedents, and the context of problems introduces new variables. The systems approach is a rational approach to finding a practical, usable solution that implements existing theory as well as developing new theory along the way. Life cycle planning is always included in calculations, and human factors are used to fit the solution to the user's needs and abilities.

Robert Gagné edited a seminal work, *Psychological Principles in System Development* (1965), in which processes for engineering the human side of human-machine systems were described in great detail, with specific attention to the training function necessary to prepare humans to operate within a system environment. Soon after *Psychological Principles* was published, Gagné's associates, especially Leslie Briggs, began to popularize the systems approach among instructional designers. This set off a trend in which the systems approach was simplified through several generations of instructional design models (see the next section).

The systems approach was evolved to solve very complex problems. It is closely related to what is practiced today as design-based research (Bannan-Ritland, 2003; Collins, Joseph, & Bielaczyc, 2004; Reeves, Herrington, & Oliver, 2005). The systems approach cannot be equated with the procedural or formulaic process approach represented by existing instructional design models. Problems suitable for the systems approach include many unknowns and uncertainties, which make the problem unique and which influence the order of problem solving, so that one of the major activities of the solver is always to decide which part of the problem to attack next. This is a quality in the solving of instructional design problems that might be reclaimed, as described later.

Bannan-Ritland (2003) suggests that design leading to educational interventions should "move past isolated, individual efforts of design research" and undertake research "that considers both field studies and experimental research methodologies" (p. 21) in programmatic rather than piecemeal studies. What this means to the instructional designer is that every design is an opportunity to learn something from having designed and that chained design efforts over time can be used to create new knowledge, about instruction and about design, much as would occur in an application of the systems approach.

The tendency in instructional design to reduce the systems approach to a process or a model can be reversed by considering each new project and each new design problem as a type of small-scale research and an opportunity to learn about designing. What has been learned from past projects can be chained with what is learned from the present project. Bannan-Ritland (2003) proposed that the challenge to instructional designers is to "draw[s] from traditions of instructional design…product design…usage-centered design…and diffusion of innovations…as well as established educational research methodologies…." (p. 21). Design-based research restores a larger perspective that is lost when the scope of reference is the single project. Bannan-Ritland's comparison of instructional design with research and development processes from several other fields defines a trail of breadcrumbs for researchers in instructional technology.

ISD Process View

A third view of instructional design is the one most familiar to most designers—instructional design models. Instructional technologists at first enthusiastically embraced the systems approach, but it was so complex as a process that designers interested only in creating a product found the tool too large for the job. Not every designer had the goal of creating new knowledge on every project, and most worked under heavy resource constraints and client product expectations.

A process of simplification began to temper the demands of the systems approach and create a design process that fit the hand of this more practically oriented designer, who often worked alone or with a small team. This set off a trend toward instructional design models that bore the title "systems approach" but that increasingly lost resemblance to it. In this melee the original aims, methods, and spirit of the systems approach were largely lost, though the title of "systems approach" was retained. In the hands of average users, design models nominally based on the systems approach became more like formulas to be followed than a method of robust and unpredictable interdisciplinary problem solving (Gibbons et al., 2013; Smith & Boling, 2009). Figure 2 gives a composite view of the core elements that were explicitly part of or implied by design models proposed during this period.

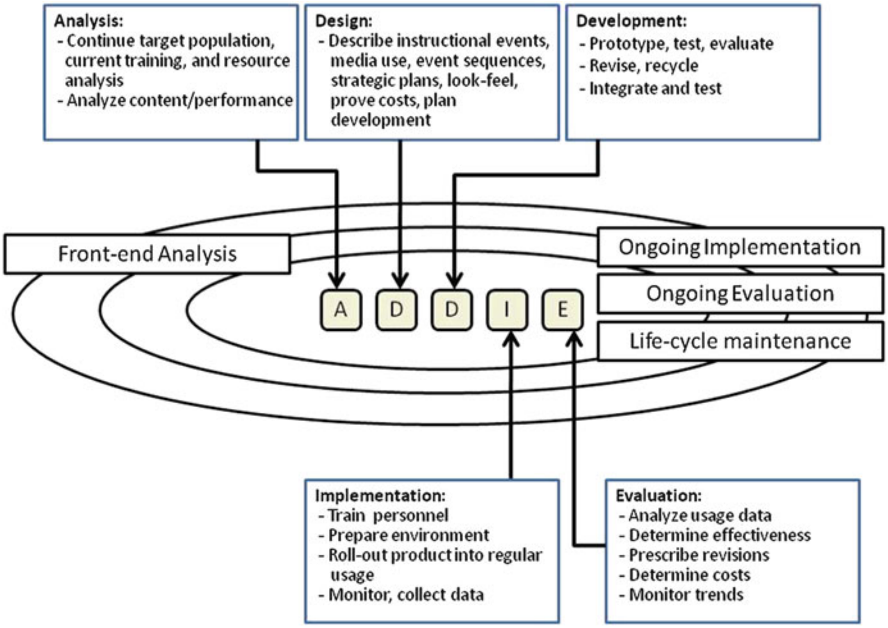

Fig. 2 A composite instructional design model showing the relationship to front-end analysis and after-project implementation, evaluation, and life cycle maintenance processes from Gibbons (2013)

The elements of this model should be sufficiently familiar not to need enumeration for this audience. What is important is that the model concept became so prevalent that for almost 50 years it was regarded as the orthodox approach to instructional design.

Numerous textbooks were written—at first for practicing designers, but eventually for novices and school teachers—describing a mostly standard process. What tended to vary from model to model was the grouping of design tasks. In this way, what was originally promoted in the name of the systems approach began to look little like its namesake. Design models became associated with the designations instructional systems design/development (ISD) and ADDIE. Gibbons et al. (2013) give a more detailed description of model proliferation.

The introduction of instructional design models was a major step forward for what had been a relatively disorganized instructional design world. But no sooner had the innovation of design models become popular than some problems became apparent. Many designers began to notice that what the ISD model told them to do didn't match what common sense and expediency told them they had to do to get the job done. The ISD narrative didn't describe what they really had to do in the real world (Cox & Osguthorpe, 2003; Rowland, 1992; Yanchar et al., 2010). Often designers found that the models led them to certain kinds of solution more easily, and over time design solutions began to look more and more similar.

At the same time, it became harder to design other kinds of things, such as simulations, collaborative learning, and games. Designers also found ISD models hard to apply in projects for culture, attitude, learning in informal settings, and other socially contexted and learner-centered methods.

Designers noticed that design projects were accompanied by decisions that had already been made, which seemed to eliminate the need for some design processes, but models didn't explain how to adapt themselves to these unexpected situations. Some model builders (mostly large organizations) took the specification of ISD processes to the extreme, defining processes in such great detail that the documentation of the process stood taller than the designer who used it. Some organizations insisted that the model processes be applied exactly as specified, leaving the designer no latitude for invention, innovation, or adjustment. Designers following process models discovered that it was hard to know how to inject theory into their designs, especially since many models came to include built-in theoretical commitments. Finally, some designers felt that ISD described how to carry out administrative and managerial functions at the periphery of the design without telling them how to actually determine the structures and details of a design.

Over time, the design model became recognizable as a special case of a general engineering model, adapted for application within instructional design and not an instance of the systems approach. Models from the very beginning (with Gagné and later with Briggs) incorporated domain-specific assumptions that limited their generalizability. For example, task analysis appeared in Gagné's original man-machine process formulation, despite the fact that not all design problems yield appropriate results when task analysis is used. Over time, highly simplified models created for use by untrained designers became the most well known. For example, the Interservice Procedures for Instructional Systems Development (IPISD; Branson et al., 1975) promulgated by the Army Training and Doctrine Command. Simplified models became used by constraint by a large number of novice military and government designers as cookbooks, so they became the most familiar face of instructional design to a large number of practitioners, many of whom later decided to make a career of instructional design in the growing commercial world.

The history and prevalence of instructional design models is one of the reasons for a conference on the future vision of instructional design such as this symposium. Placing design models in perspective with other design descriptions is one of the purposes of this paper, and that requires elevating other views of design, since models have been the predominant theme in the instructional design process literature for over 40 years.

Functional-Modular (Layer Design) View

A fourth view of instructional design can be termed a functional-modular view. This view is based on analyzing the functions of the designed artifact. It is based on the philosophy that designed artifacts can be characterized in terms of decomposable functional "layers" within which the designer addresses more detailed design questions (Gibbons & Rogers, 2009).

To obtain the benefits of layered design, one does not give up ISD design principles, since a general engineering process still raises important questions during design creation, especially at the higher levels of design project management. However, the order of design decision making changes at more detailed design levels.

The functional-modular view of design assumes a distinction between scientific and technological theory and that there are at least two types of technological theory: design theory and domain theory (Gibbons & Rogers, 2009). Instructional theory is a type of domain theory. Instructional theories are instances of domain theory; they pertain to the design of instruction and supply the elements incorporated into designs.

Functional-modular (layer) theory, on the other hand, is a design theory. It creates an architectural framework within which multiple domain theories pertaining to each layer can populate the design.

Functional-modular theory is applied in fields other than instructional design: in business, computer design, software design, architecture, and engineering. Examples of this include:

- Donald Schön (1987), in *Educating the Reflective Practitioner*, describes how an architectural design problem consists of numerous subproblems, each having its own principles, standards, and design terms, specialists, and domain theories.
- Stewart Brand (1994) likewise describes the layers of a building's design, noting that when a designer uses layering deliberately, a building's usable lifetime is extended because as layers aged unevenly they can be changed independently without destroying the entire edifice.
- Baldwin and Clark (2000) describe how the principle of modularity, which is based on the principle of design layers, is the economic factor that made the modern personal computer, with its replaceable functional modules (boards, drives, etc.), possible. Early computers were monolithic in their designs, so changing one part of the system meant disrupting the whole system design. Functional-modular separation changed that irreversibly.
- Fowler (2003) describes the enterprise architecture of software that increasingly forms the core mechanism that businesses use to carry out their essential functions. He explains the structure of this software in terms of three main layers which can be changed independently: "most nontrivial enterprise applications use a layered architecture of some form…." (p. 2).
- The software that forms the Internet is structured in terms of functional layers. Software protocols, the bits of software by which the Internet works, carry out their functions within the structure of multiple functional layers. Competing layer models have been proposed, some with four layers, some with five, and some with seven.
- Ericsson and Erixon (1999) describe the concept of *modular product platforms*, a design principle that considers a marketable product to consist of a family of reconfigurable components that can be assembled in different combinations to form different versions of the product. Separation of modules is a layering process. A module, or layer, may be defined for many different reasons, based either on conceptual or practical concerns.

Uyemura (1999) describes the value of thinking in terms of design domains with reference to digital system design:

> The detail of interest to you at a particular time depends on the level where you are working. Sometimes you will be interested only in the overall function of a complex unit, whereas at other times you may need to understand every element that goes into making a basic unit. The power in this approach derives from the fact that the important aspects vary with the level…. (p. 18)

There is evidence that instructional designers tend to think of designs in monolithic, unsubdivided terms. Frequently designers will refer to the configuration of their design in terms of a dominant school of thought, such as "this is a constructivist design" or "this is direct instruction." As Uyemura shows, this is not true in other, more mature design fields. Automotive and aeronautical designers think of their designs in terms of the systems and subsystems they incorporate. An auto designer might be expected to describe several subsystem influences on the design: "This model has rack and pinion steering, a V-6 overhead cam engine, manual transmission, and is equipped with the stabilizer package."

The instructional design field will gravitate toward more detailed descriptions of designs as the field matures and it becomes commonly understood that many subsystems are required to complete a design, each part of the design being dominated by its own design theories and philosophies. This evolution, which is already underway, has escaped notice. Instructional design teams today consist of multiple specialists representing multiple specialized domains, including artists of specialized kinds, writers, assistant designers, subject-matter experts, programmers, assessment experts, evaluators, and implementation specialists. Each of these roles contributes expertise to one or more layers of a design using principles and theories that pertain to just their specialty. The more complex the design, the larger the number of specialists required.

Layer design theory as described by Gibbons and Rogers (2009) names seven design layers, or domains, of an instructional design explaining that there may be more or fewer layers, depending on the insight of the designer. These layers represent major functions carried out by an instructional artifact. Each layer represents a subproblem of the original design problem, and each layer in turn decomposes into sub-layers that have all of the properties of a layer. Figure 3 illustrates the following layers named by Gibbons and Rogers:

- *Content layer*. An instructional design contains—implicit or explicit—a description of the structural nature of that which is to be taught. There are implicit or explicit units into which the subject matter and performances are divided. Teachers divide subject matter into parcels that associate with units, lessons, and activities. Instructional designers identify facts, concepts, tasks, rules, and so forth, and associate them with behaviors to form instructional objectives, but the content structure is only one element of an objective.
- *Strategy layer*. An instructional design must specify the physical organization of the learning space, the social organization of participants, their roles and responsibilities, instructional goals that consist of a content element and a performance element, the allocation of goals to time structures called "events," and strategic patterns of interaction between the learner and the instruction. These things are

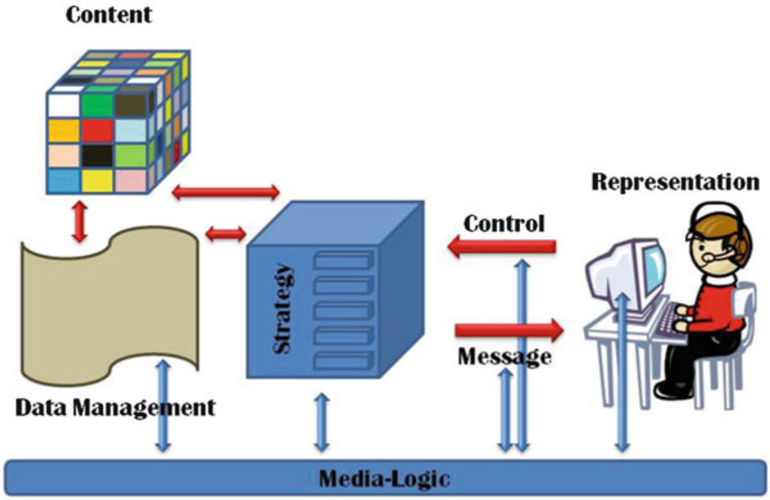

Fig. 3 The system of major layers proposed by Gibbons and Rogers (2009). Different designers perceive different layers, and layers subdivide as new technical and theoretical knowledge emerges or according to practical considerations from Gibbons (2013)

the concerns of the strategy layer. This layer has many sub-layers, each one corresponding to the concerns just listed and more.

- *Message layer*. A design must specify the units of tactical communication—the elements of the instructional conversation. These are the message structures through which the instruction communicates with the learner in a conversational manner. The units identified within the message layer are chosen because of their ability to carry out the larger strategic plan at a detailed exchange-by-exchange level.
- *Control layer*. A design must specify the control devices through which the learner expresses messages and actions to the instruction, along with a language that attaches meaning to inputs from the controls so that the learner's meaning can be analyzed and interpreted.
- *Representation layer*. A design must specify the representations that make message elements visible, hearable, and otherwise sense-able: the media representation channels to be used, the rule for assigning message elements to media channels, the form and composition of the representation, the synchronization of messages delivered through the multiple channels, and the concrete, tangible representations of content.
- *Media-logic layer*. A design must specify the rules and mechanisms for executing the functions of *all* of the other layers as well as the rules and mechanisms for communications with the environment outside of the instruction.
- *Data management layer*. A design must specify data to be captured, archived, analyzed, interpreted, and reported.

Layers are a natural result of the evolution and maturation of a design field. Layers emerge as new technological knowledge accumulates. The list above does not constitute a single, "standard" set of layers, because not only are designers' perceptions of useful layers in a state of constant change at the detailed level, but every layer is subject to splitting into sub-layers, modifying, and growing as technical knowledge and theoretical insight grow.

Moreover, different designers can "see" different layers. They may see the layers that other designers see, but some designers "see" additional layers that others have difficulty discerning. To that extent their layer definitions are different from those of other designers, and to the extent that these are useful and productive layers, they constitute the designer's competitive advantage: a value-added. Private layers allow a designer to think about design in more detail and nuance, and they lead to new design experiences, new experiments, which lead to new design insight and understanding.

Shared or public layers give a designer the ability to subdivide large design problems into smaller, solvable problems without losing the integrity and coherence of the larger design. They give a design team a common set of languages for describing the entire design as well as its constituent sub-designs.

Architectural View

A fifth view of design is the architectural view, as described by Blaauw and Brooks (1997). The architectural view describes how a designer can bring abstract ideas into a design in a way that gives coherence to the design, and this happens at the finest level of detail, at the heart of the design.

Blaauw and Brooks, who are computer designers, distinguish three stages in the evolution of a design: *architecture*, *implementation*, and *realization*. These are stages of *design*, not *manufacture*, and they are accomplished in parallel, interacting with each other, with the designer moving from one to the other as understanding of the design emerges. These stages involve design decisions at different levels of abstraction. They attempt to describe how a vague idea emerges from the fog in a designer's mind, takes shape, and eventually hardens into a plan—a design. The designer's mind moves back and forth between these stages, and they mutually influence each other.

The three stages of the evolution of a design are described using the example of designing an analog clock (one with hands):

- *Architecture*. The architecture of an analog clock consists only of (1) pointers or indicators to register the current hour and minute and (2) the spatial positions on the clock that correspond with hours and minutes—spatial positions that the pointers or indicators can be made to designate at a given moment.
- This specifies the clock's (1) conceptual structure and (2) functional behavior as seen by the user, but nothing more. Notice the things that are not mentioned in this description of the architecture: not the size and shape of the hands, their

placement, their pattern of motion, their direction of motion, their color, the material they're made of, nor their style, not the placement of the numerals, or whether there will even be numerals. The architecture describes the clock only in terms of those abstract functions essential to time-telling. Moreover, the description of the elements of the architecture is completely free of detail. There is no mention of dimension, physical structure, nor any property.
- *Implementation*. The implementation describes the mechanisms of the clock and how they operate together. It describes how the clock's functions (described in the architecture) are made to happen. These mechanisms are described in terms of energy and information transmission.
- Blaauw and Brooks show several ways the abstract architecture could be implemented for a clock. They point out that the key elements of this particular implementation problem are (1) how to power the movement of either the pointers or the things pointed to and (2) how to transmit that power through a mechanism that causes the pointer or pointed-to to be in the correct position at any given time. Notice that this divides the clock design problem into two fairly independent subproblems—the power mechanism and the motion mechanism. Notice also that there are again no surface details specified. Blaauw and Brooks explain: "the implementation…is the *logical* organization of the inner structure of a designed object" (p. 5, emphasis added). That is, how the clock is made to tell time. Consider at this point how many different surface designs of clock could be generated from this level of abstract description. This is the generative kernel of the design. Together, the architecture and the implementation embody the operational principle of the design as it is described by Polanyi (1958) and Vincenti (1990).
- *Realization*. The realization describes all of the remaining details of the design. (Remember that this is still *just design*, not manufacture.)

Blaauw and Brooks call these the design's "geometries, strengths, tolerances, and finishes" (p. 5), which includes the physical placements of individual design elements, their connections with each other, their material specifications, their size, shape, color, texture, and appearance. Blaauw and Brooks point out that if the clock is to be handmade, some of these realization decisions may be left undefined and be allocated to the craft worker (who is both a detail designer and a manufacturer). If the clock is to be mass-produced, however, the realization of the design is completed to the minutest detail and fully documented, ready to be sent to manufacture.

Both the architecture and implementation stages of a design are abstract. A novice designer does not normally think in abstract terms, but an expert designer is able to. It is, in fact, one of the indicators of an expert instructional designer to be able to see below the surface of the design into its interior—to the abstract parts of the design that represent why it works. These inner workings operate by conveying energy and information. They determine how energy and information are transferred, transformed, stored, regulated, and delivered to where they are to be applied. This idea is elaborated below in the discussion of operational principles.

Team Process

A sixth view of design can be called the Team Process view. Most instructional design is carried out by multidisciplinary teams. Just as there are private design skills, there are also team design skills.

Team design is a method for disciplining and coordinating the creative efforts of design team members across several phases of activity. Bucciarelli (1994) describes the challenge of coordinated effort and shared mindset within a design team:

> Shared vision is the key phrase: The design is the shared vision, and the shared vision is the design—a (temporary) synthesis of the different participants' work within object worlds. Some of this shared vision is made explicit in documents, texts, and artifacts—in formal assembly and detail drawings, operation and service manuals, contractual disclaimers, production schedules, marketing copy, test plans, parts lists, procurement orders, mock-ups, and prototypes. But in the process of designing, the shared vision is less artifactual; each participant in the process has a personal collection of sketches, flowcharts, cost estimates, spreadsheets, models, and above all stories—stories to tell about their particular vision of the object…. The process is necessarily social and requires the participants to negotiate their differences and construct meaning through direct, and preferably face-to-face exchange. (p. 159)

The team innovation process can be described as repeating cycles of activity for (1) the conceptual unfolding of the design and (2) the day-to-day management of schedules, people, resources, and client relationships. These come together to define a process that alternates between (1) periods of specialty design activity carried out by individuals and (2) periods of team-led integration, refactoring, and fitting of sub-designs together and then evaluating the design by the team as a whole. Judging takes into account the changing environment of the design, including stakeholder criteria and resources.

The alternation between specialty design and joint fitting of the design elements with each other takes place in a constant cycle of low-stakes specialty-to-specialty collaborations and high-stakes integration and judging events. This reverberating process refines, focuses, disciplines, and eventually produces a final design. Part of project planning involves deciding the frequency of these cycles. Informal events may take place daily, but design team leadership sets schedules for major design coordination and integration points. Projects using virtual teams must pay more careful attention to the timing and scheduling of formal design coordination events.

Operational Principle View

A seventh view of design pertains to abstract concepts called operational principles and how they are incorporated into designs. The best way to see operational principles at work is to examine a Rube Goldberg machine at work. Goldberg machines are seen more commonly of late—from elaborate contraptions in music videos to serious educational use of them in teaching STEM subjects, where learner-produced contraptions are used in design and problem solving.

In a Goldberg machine a trigger event sets off a chain reaction of other events, until some trivial action occurs—a plate is washed, or a shoe is polished. Though Goldberg machines involve concrete things like wood, metal, and animals, these are concrete manifestations that hide inside something more abstract and invisible: the transfer of energy and information through a chain of events to a final destination where they accomplish some desired outcome. In physics terms, these physical machines deal with potential and kinetic energy and their transfer through the interaction of mechanisms.

At each point in an event chain, energy is supplied at a mechanical part and passed along the chain. What you see in a Goldberg cartoon is a physical embodiment, but what you don't see is the invisible transfer of energy and information that occurs as springs pull trap doors open and levers are pressed. Ironically, though we feel we see how the machine does its work, a physicist would say that it is the *invisible* transfers of energy in Goldberg machines that actually do the work.

A Goldberg machine can use basic principles like lever, spring, and inclined plane in multiple places in the same contraption; in one place it looks like a trap door, and in another it looks like a teeter-totter. The *abstractions* behind the surface manifestations are referred to as *operational principles.* Operational principles exist in every energy-using system. Operational principle is a term proposed by Michael Polanyi (1958) to describe how things can be made to work. It is not a scientific concept but a technological one. An operational principle is an abstract germ of an idea used at the   of a design to generate a hundred or a thousand different surface designs, all based on the same underlying principle of operation.

For example, designs of virtually all airplanes today are based on a single operational principle identified by George Cayley in the early 1800s. Cayley refined the challenge of flight into a single solvable problem statement: "to make a surface support a given weight by the application of power to the resistance of air" (Vincenti, 1990, p. 208). Note that Cayley's principle does not specify the size, shape, material, or relative dimensions and proportions of the surface or size of the power source.

What Cayley devised was not the design for a single airplane but the essential pattern for a million airplane designs—a basic pattern of the distribution and balancing of forces from which an endless number of specific designs could be generated. When the Wright Brothers flew successfully, they credited Cayley's idea, which they incorporated into all of their machines. When Curtiss improved the concept of flight controls, it was on a plane designed according to Cayley's operational principle for flight. As the variety of specific flyable designs multiplied, virtually all of them incorporated Cayley's operational principle. Today, thousands and thousands of specific airplane designs exist, all based on Cayley's principle, from the smallest experimental craft to the largest passenger liner.

Different values can be assigned to the variables of a Cayley design:

- The placement of the engine (forward or backward-facing, centered or distributed on the wings)
- The placement of the wing surface (above the body, below the body, forward, aft)

- The shape of the wing (flat, thick, tapered)
- The type of power used (reciprocating, turbine, jet)
- The means of propulsion (propeller, jet exhaust)

Everything is free to vary that does not nullify the central operational principle. This is what makes the number of possible combinations multiply.

Rube Goldberg machines and airplane designs are relevant to a discussion of instructional design because every human-made artifact incorporates one or more operational principles. Therefore, designed instructional products have their effect through an operational principle that defines the transfer of energy and information through actions and artifacts and the sensations they produce. Clark (2009) describes the operational principle concept, calling it the "active ingredient." Clark describes a systematic, four-stage research and development cycle that can be used to isolate "active ingredients" of instruction through experiments and then apply them in real-world settings: "…Active ingredient analysis…yields a recipe for constructing [a new] intervention that reflects the critical elements of the [laboratory] intervention that worked under controlled conditions" (p. 17).

Clark says that caution "must be exercised so that we do not simply group the treatments that share the same name" (p. 13). He warns against using common labels of things that resemble each other on the surface. What we should learn to see, he says, is "both novel and critical" and "we must look more deeply" (p. 13).

> Effective intervention design requires identifying the "active ingredients" or the key structural elements of the interventions or research treatments that have been found in…experiments to influence our chosen outcomes…. There are no rules yet for conducting this kind of analysis, but it is clear that we must look beyond the labels researchers give to their treatments in published articles and analyze the operations they implemented and their presumed impact on people and organizations…. The active ingredients we need as the core of a new technology are the causal agents in the experiments that were surveyed in [research]. We have evidence that these ingredients influence the problems we want to solve at the deepest structural level and so they must be the centerpieces in a solution. (pp. 13–14, emphasis added)

An instructional design incorporates an operational principle. It can transfer, transform, and conduct energy and information through a series of physical and intellectual mechanisms invisibly to bring about a desired result. When designs work, it is not by chance, it is because there is an operational principle active. Every design that achieves its intended results does so through an operational principle. If a designer designs without awareness of operational principles, an effective design will still achieve its effect through the operational principle incorporated into the design without the designer's explicit knowledge of it.

It is possible to discover the operational principles of a working artifact through a method of subtraction. A design that works can be whittled down in successive trials until it breaks and no longer works properly. At the point of breakage, something essential has been lost and has to be restored. The boundary of a principle has been crossed. Then trials continue, dissecting out other features until they break. This method works in a practical setting—usually over the span of multiple trials, such as in rapid prototyping or multiple evaluation and revision cycles. If designers can identify the operational principles they use in advance and apply them in a deliberate manner, the number of required cycles can be reduced.

An example of applying an operational principle would be represented by adopting "conversation" as the most basic design commitment and causing all other design considerations to revolve around it. Everyday conversations represent a dynamic and temporary structure held together by invisible forces of attraction and repulsion. Attraction is analogous to a magnetic or gravitational attraction between people. The opposing force of repulsion consists of anything that reduces commitment to the conversation: boredom, conflicting goals, or discomfort. There are many ways of establishing and maintaining attraction during an instructional conversation. At the same time, the opposite forces of repulsion are in competition, tending to drive the conversation apart. These forces—attraction and repulsion—hold the conversation together in a kind of dynamic tension so long as the feelings of attraction are sufficiently strong on both sides of the conversation.

If we were to compare this with the operational principle of Cayley and the variables that influence aircraft design, we would search for force-creating instructional acts that can be substituted into the attraction and repulsion sides of the equation. In a separate publication (Gibbons, 2013), I propose an extensive list of actions that create attractive and repulsive forces that can exert sustaining influence on an instructional conversation. A shorter list of these is provided in Table 1, which shows how they pertain to holding together a conversation at the beginning, in the middle, and at the end.

How does the operational principle concept relate to stock literature terms such as "motivation," "engagement," "participation," and "interaction"? These terms are used to describe goals and methods of instruction. They represent ideals. Operational principles describe the actions and therefore the forces behind these terms that allow them to be realized. They describe the inner working of emotional and intellectual forces that influence moment-by-moment changes in the learner and sustain the learner's commitment to exercise the agency to remain in the conversation or refuse it.

The entries included in Table 1 do not constitute a philosophical or theoretical statement beyond a commitment to the concept of conversation as the metaphor of instruction. They illustrate how individual actions during instruction introduce pulses of energy or information into an instructional conversation, either strengthening its attractive force or reducing it. For example, substituting "invite" for "compel" gives a much different dynamic to the conversation. A designer of "problem-based learning" may use "compel" rather than "invite," but it can be seen that different forces are set in motion by this choice.

It is no wonder, then, that with many such substitutions possible during the design of problem-based learning, there is great variability in problem-based learning research findings. It seems worthwhile, therefore, to consider describing instructional treatments in research reports in sufficient detail to allow the reader to discover firsthand the operational principles embedded in the treatments as easily as we read a Rube Goldberg contraption.

Table 1 Representative actions on both sides of an instructional conversation that either increase attraction or increase repulsion during different stages of an instructional conversation

Possible instructional action			Possible learner actions		
Initiating the conversation					
Invite	Contact	Rouse	Desire	Show interest	Attend
Tantalize	Welcome	Entice	Continue	Respond	Refuse
Announce	Entreat	Startle	Ignore	Answer	Notice
Approach	Puzzle	Offer			
Wake	Appeal				
Securing commitment to continue					
Propose	Challenge	Persuade	Counter	Accept	Refuse
Suggest	Bargain	Counter	Decline	Trust	Contract
Promise	Retract	Request	Consent	Continue	Join
Agree	Contract	Specify	Bargain	Propose	Request
Pester	Offer	Require	Ask		
Fascinate	Enlarge	Excite			
Conducting the conversation					
Display	Respect	Exhibit	Plan	Analyze	Deduce
Assist	Scaffold	Anticipate	Imagine	Suggest	Deliberate
Reason	Aid	Praise	Produce	Act	Choose
Counsel	Adjust	Provide	Meditate	Use	Ask
Debate	Encourage	Judge	Practice	Exercise	Consider
Charge	Reassure	Cooperate	Interpret	Invest	Respect
Argue	Portray	Feedback	Debate	Trust	Digest
Honor	Serve	Set stage	Theorize	Notice	Discover
Adapt	Comfort	Explain	Decipher	Connect	Try
Introduce	Cite	Measure	Respond	Explore	Observe
Dare	Discern	Test	Question	Cooperate	Converse
Inspire	Uplift	Critique	Dispute	Experience	Disregard
Dramatize	Collaborate	Guide	Cooperate	Reflect	Anticipate
Model	Socialize	Refer to	Articulate	Collaborate	Investigate
Transferring responsibility to the learner and terminating					
Culminate	Agree	Evaluate	Assess	Celebrate	Award
Finalize	Rate	Certify	Validate	Reminisce	Commit

Design Language View

The eighth view of design can be referred to as a design language view. Design is in one sense a linguistic exercise, but the terms of designing do not necessarily exist in written language. They exist in the many public and private design languages in the mind of the designer and in the shared, public concepts of a profession.

An observer watching an animated robot dressed as Abraham Lincoln can maintain detachment, realizing that the robot consists of individual joint articulations, each of which has only a few position states. An animated fountain likewise is made up of perhaps 300 identical water jets, each of which has only about ten distinct spurt patterns. The observer realizes that what seem to be moving walls of water are simply the coordinated actions of patterns of jets which have been timed precisely. Likewise, the robot's seemingly human postures and movements are synchronous, timed sequences of relatively uncomplicated joint motions.

These examples provide an insight into one aspect of design languages: Designers join together relatively simple primitive elements into structures whose enacted experiences convey information and produce emotions. At one end of the spectrum of abstraction are design language terms that define composite effects: "walls" of water, moving "shapes," playful "randomness," and awe-producing "order"—all calculated to produce an emotional reaction. The viewer recognizes these as symbols seen in the everyday world, and so they are gross terms the designer uses to convey a message to and evoke an emotion in the viewer.

The designer may have a name for each effect. But the designer may also have names for the individual elements—abstractions at a different level of detail—that lead to these effects: the crooking of a finger, the lifting of an eyebrow, and the rotation of the neck joint. These are much more detailed and mechanical terms in a design language for robots. The creation of the grand effects from small mechanical motions involves the conscious use of design language abstractions at multiple levels—terms that can be given names so that a team of designers can express and talk about an evolving design both in detail and in broad terms. In the process of calculating an effect, there may be *translations* required between languages at these different levels.

The value of design languages is found as much in their translation uses as in their communication uses. The mechanical acts of the robot do not create the desired effect when they are performed randomly. Only when they are part of a larger pattern do they come to have impact. In order to achieve this impact, the designer must translate the terms of a grand effect—the sweeping gestures, the expressions—into individual robotic motion acts and sequences of acts. In the end, the robot has no idea of the experience it produces for the user, but it faithfully performs its individual acts, and the effect of the suite of acts produces the effect: Viewers feel emotion and obtain information.

Design languages evolve as a technology matures. One measure of the maturity of the design field is the precision with which designers can discuss their designs in design language clearly and unambiguously. Design languages not only allow professionals to communicate generally about their work, but individual teams use design languages by inventing additional terms shared only by the team. Sometimes design languages are used in a closed circle to describe trade secrets which constitute a source of advantage. In the past such languages provided the basis for craft guilds to protect against competition and retain economic advantage.

A design language is a set of conceptual building blocks for describing designs and the conduct of designing. The vocabulary of a design language exists in two senses: (a) as thought structures in the mind of an individual and (b) as named entities that have verbal or symbolic identifiers that make them public. Every designer possesses and uses a number of design languages, though few designers are conscious of them as languages. Not all design languages have specific verbal terms. That becomes evident when two designers are conversing about a possible alternative and one or the other begins to use hyphen-connected phrases (e.g., “that-thing-we-did-on-the-last-project”). Many language terms born as hyphenated phrases are later given a single-word name as usage of the innovation catches on and people need to talk about it more often.

Public design languages use the syntax of a native language, substituting design language terms—which are nouns, verbs, and modifiers—into standard sentence patterns. When this happens, a conversation between two professionals becomes hard to interpret. Multiple design languages are required in designing an artifact. When Edison first began to invent, he had no idea of the number of design languages this would eventually entail:

> … Technologists [like Edison] are tied into less obvious meaning systems [professional worlds] for the development, appreciation, production, funding, operation, maintenance, social control, evaluation, and distribution.... These…functions are likely to be distributed among different groupings in society.... Paper must be filed with financial backers, government regulators, technical R&D departments, sales forces, material suppliers, production machinery producers, and shop floor designers. (Bazerman, 1999, pp. 336–337)

Edison’s light bulb invention spawned hundreds of design language terms: bulb, filament, base, contacts, and so forth. These of necessity found their way into the documentation of many other team members responsible for placing the light bulb on the market and into homes, offices, and workshops. As the technology continued to develop, additional terms perforce crept into usage because additional new parts of the invention also had to be named: socket, lead, terminal, connector, switch, and so forth. In the end, an entire electrical generation and distribution system had to be created, along with a multitude of new design language terms.

New design languages and language terms come into being in many ways, including the following examples:

- With the introduction of a new theory
- As growing expertise creates new technical concepts
- As new instructional techniques are developed
- As new hardware and software concepts are introduced
- As new kinds of artifact evolve
- As authors invent new terms in the literature
- As new theories are developed
- As professional cultures develop
- As new patterns of product usage are invented

Some design language terms are not shared with others, either because they are subtle and we find it hard to articulate them or because we choose not to share them in order to preserve an advantage. The continuous evolution of design languages, expressed and unexpressed, is the key to continued learning and improvement in any field.

Conclusion

The views of design in this paper join other views described in this symposium. Together they suggest how conceptual tools from multiple design disciplines can inform the thinking of the instructional designer. Instructional design can and should begin to tap into the relevant literature from other design fields.

A shift can be seen toward design processes that make use of traditional, classical concepts while encouraging the inclusion of new, imaginative processes and structures not suggested by traditional approaches. This paper encourages us to consider design as a bridge between a completely conceptual world of vague theoretical ideas on the one hand and a completely practical physical world of results and goals on the other.

Design by its nature begins with fuzziness. It is the process of drawing out of nowhere solutions to practical problems through the creation of artifacts, processes, and experiences. It is in this respect an act of magic. This sleight of hand becomes possible only as the designer begins to see things that others can't see or didn't see and learns to manipulate invisible structures of experience.

Seeing, to a designer, must take place at different levels of scale. It must employ gigantic levers in the form of experiences that last days, weeks, or even years. At the same time, it must be sensitive to minute forces set in motion by a glance, a word, or a motion.

The designer's seeing must also encompass the very abstract and the very concrete without being seduced by the very concrete. The history of technology in general, and in individual fields specifically, records in every case a journey from robust concrete concepts to wispy, ethereal abstractions. The progress of a technology depends on this journey. The digital computer as a concept began with the quest for mechanical devices to perform mathematical calculations. Who would have in those days imagined that the concept of a computer would ultimately be expressed in device-less terms: in the form of a model whose many subsequent realizations in device form would outlast generations of changes in device technologies, with little need for revision of the original conceptual model?

The imaginations of instructional designers, especially novice designers, are so easily captured by the allure of the "bright lights and loud noises" offered by today's production technologies that it takes experience to see beyond these things to the invisible qualities of a design that really matters. Nor is this descent into the rabbit hole of abstraction one where a designer ever touches bottom. Hence, the reason for every designer to be taught from the beginning that design expertise is not a destination but a lifelong commitment to constant refinement of the ability to observe and notice things that didn't seem to be there before.

Add to this the complication that an advancing technology of design is no longer a singles sport. The lone designer who could do it all is an extinct species. The social nature of designing makes it therefore, in one view, a linguistic exercise in which the dual challenge is to bring the thinking of a team into focus—both to allow the cross-specialty communication of technical aspects and to allow the sharing of visions and imaginations that lie entirely within no one's particular domain. Instead, new domains are invented as designers see more.

This paper began by describing a need for better, more robust descriptions of design to feed the growth of experienced designers as well as educating novices. Perhaps by teaching richer views of design, simplistic conceptions of design can be avoided among new designers, and the lifelong growth can become an expectation. Perhaps also experienced designers can find questions to advance their personal insights that will lead them on and on throughout a career of discovery that gives them the value as a professional rather than as a craft worker. For instructional designers of both types, it may be that this fascinating journey is just beginning.

References

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

Bannan-Ritland, B. (2003). The role of design in research: The integrative learning design framework. *Educational Researcher, 32*(1), 21–24.

Bazerman, C. (1999). *The languages of Edison's light*. Cambridge, MA: MIT Press.

Bichelmeyer, B. (2003). Instructional theory and instructional design theory: What's the difference and why should we care? *IDT Record*. Retrieved February 17, 2010, from http://bit.ly/9HkisA.

Blaauw, G., & Brooks, F. (1997). *Computer architecture: Concepts and evolution*. Reading, MA: Addison Wesley Longman.

Brand, S. (1994). *How buildings learn: What happens after they're built*. New York: Penguin Books.

Branson, R. K., Rayner, G. T., Cox, J. L., Furman, J. P., King, F.J., & Hannum, W. J. (1975, August). *Interservice procedures for instructional systems development* (5 vols.) (TRADOC Pam 350–30). Ft. Monroe, VA: U.S. Army Training and Doctrine Command (NTIS Nos. AD-A019 4860-AD-A019 490).

Bucciarelli, L. L. (1994). *Designing engineers*. Cambridge, MA: MIT Press.

Clark, R. E. (2009). Translating research into new instructional technologies for higher education: The active ingredient process. *Journal of Computing in Higher Education, 21*(1), 4–18.

Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design research: Theoretical and methodological issues. *The Journal of the Learning Sciences, 13*(1), 15–42.

Cox, S., & Osguthorpe, R. (2003). How do instructional design professionals spend their time? *TechTrends, 47*(3), 45–47.

Drucker, P. (1989). *The new realities*. London, UK: Mandarin.

Ericsson, A., & Erixon, G. (1999). *Controlling design variants: Modular product platforms*. Dearborn, MI: Society of Manufacturing Engineers.

Fowler, M. (2003). *Patterns of enterprise application software*. Boston: Addison-Wesley.

Gagné, R. M. (Ed.). (1965). *Psychological principles in system development*. New York: Holt Rinehart & Winston.

Gibbons, A. S. (2013). *An architectural approach to instructional design*. New York: Routledge.

Gibbons, A. S., Boling, E., & Smith, K. M. (2013). Instructional design models. In M. Spector, M. D. Merrill, J. Elen, & M. J. Bishop (Eds.), *Handbook of research on educational communications and technology* (4th ed.). New York: Springer.

Gibbons, A., & Rogers, P. C. (2009). The architecture of instructional theory. In C. M. Reigeluth & A. Carr-Chellman (Eds.), *Instructional-design theories and models: Vol. 3. Building a common knowledge base*. New York: Routledge.

Gibbons, A. S., & Yanchar, S. (2010). An alternative view of the instructional design process: A response to Smith and Boling. *Educational Technology, 50*(4), 16–26.
Hokanson, B., & Miller, C. (2009). Role-based design: A contemporary framework for innovation and creativity in instructional design. *Educational Technology, 49*(2), 21–28.
Jonassen, D. (2008). Instructional design as design problem solving: An iterative process. *Educational Technology, 48*(3), 21–26.
Kahin, B., & Foray, D. (2006). *Advancing knowledge and the knowledge economy*. Cambridge, MA: MIT Press.
Parrish, P. (2005). Embracing the aesthetics of instructional design. *Educational Technology, 45*(2), 16–25.
Parrish, P. (2006). Design as storytelling. *TechTrends, 50*(4), 72–82.
Polanyi, M. (1958). *Personal knowledge: Towards a post-critical philosophy*. New York: Harper Torchbooks.
Ramo, S., & St. Claire, R. K. (1998). The systems approach, anaheim, CA: KNI Incorporated. Retrieved from http://www.incase.org/productspubs/doc/systemsapproach.
Reeves, T., Herrington, J., & Oliver, R. (2005). Design research: A socially responsible approach to instructional technology research in higher education. *Journal of Computing in Higher Education, 16*(2), 97–116.
Reigeluth, C. (1999). *Instructional-design theories and models: Vol. 2. A new paradigm of instructional theory*. Mahwah, NJ: Lawrence Erlbaum Associates.
Reigeluth, C., & Carr-Chellman, A. (2009). *Instructional-design theories and models: Vol. 3. Building a common knowledge base*. New York: Routledge.
Rowland, G. (1992). What do instructional designers actually do? An initial investigation of expert practice. *Performance Improvement Quarterly, 5*(2), 65–86.
Rowland. (2008, March 27). Design and research: Partners in educational innovation. Keynote address to the Design and Technology SIG, American Educational Research Association, New York City.
Schön, D. A. (1987). *Educating the reflective practitioner*. San Francisco, CA: Jossey-Bass.
Smith, K., & Boling, E. (2009). What do we make of design? Design as a concept in educational technology. *Educational Technology, 49*(4), 3–17.
Uyemura, J. (1999). *A first course in digital systems design: An integrated approach*. Belmont, CA: Brooks/Cole.
Vincenti, W. (1990). *What engineers know and how they know it: Analytical studies from aeronautical history*. Baltimore, MD: Johns Hopkins University Press.
Wilson, B. (2005). Broadening our foundation for instructional design: Four pillars of practice. *Educational Technology, 45*(2), 10–15.
Yanchar, S. C., South, J. B., Williams, D. D., Allen, S., & Wilson, B. G. (2010). Struggling with theory? A qualitative investigation of conceptual tool use in instructional design. *Educational Technology Research and Development, 58*, 39–60.