Coming at Design from a Different Angle: Functional Design

Andrew S. Gibbons and P. Clint Rogers

Abstract This chapter defines a non-traditional view of the structure of instructional designs that holds implications for the practice of instructional designing: what we call a theory of design layering. We discuss designing in many fields as a backdrop against which we examine this theoretical view of instructional design. We describe our design approach briefly as a set of propositions. Finally we examine the implications of the design layering approach for everyday design practice and the relation of instructional theories to instructional design.

Keywords Communication · Design layering · Design metaphors · Functional design · Instructional design theory · Instructional technology

What happens at the moment of design is a tantalizing mystery. Philosophers (Polanyi, 1958) and pragmatists (Jones, 1992) have puzzled over it, circling at the edges of the question but finding it hard to reach the heart. Conceptions of design found in the literature of many fields call up the figure of the blind gurus and the elephant ("it is very like a rope," "it is very like a tree"). This chapter defines a non-traditional view of the structure of instructional designs – a different way of looking at the elephant – that holds implications for the practice of instructional designing. As we have expressed elsewhere (Gibbons & Rogers, Submitted):

> ...This architecture of instructional theory accomplishes the following: it gives designers a tool to create quality designs more consistently, it can facilitate communications about designs and theories, it can allow designers to work efficiently in design teams with a greater degree of mutual understanding, it suggests functionalities for more advanced and productive design tools, and it allows experienced designers to convey design knowledge and judgment to novices more quickly.

In this chapter, we will concentrate on the implications of this view of instructional designing – what we consider to be a layered design theory – for

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

L. Moller et al. (eds.), *Learning and Instructional Technologies for the 21st Century*,
DOI 10.1007/978-0-387-09667-4_2,

everyday instructional design practice. First we will discuss designing in many fields as a backdrop against which we can examine our particular view of instructional design. This will provide a context to show where our ideas originate. Then we will describe our design approach briefly in the form of a set of propositions. Finally we will examine some of the implications of this approach for everyday design practice.

Design Metaphors from Many Fields

Design theorists take their perspective from the design problems they encounter most often, from traditions in their fields, and from design priorities they consider most important. Lacking direct answers, they often invoke metaphors:

- Design is a *reflective conversation* with the problem (Schön, 1987). This includes reflecting after each move to determine what you have learned from it. There is no particular order in designing, but one decision can be taken as an anchor ("imposing a discipline"), following which the designer plays out its implications. Decisions can be undone ("breaking the discipline") and a new anchor chosen as a beginning point for playing out new "what-ifs."
- Design as the progressive *placement of constraints*. This view stipulates that designing involves exploring regions of feasible solutions; that feasibility is derived from the constraints chosen by the designer; that choosing constraints is how a designer uses his knowledge. Every exercise of design knowledge is construable as the choice of a constraint on the space of possible inventions. The purpose is to make explicit design knowing and reasoning: "the accumulation of constraints becomes the invention" (Gross, Ervin, Anderson, & Fleisher, 1987). In this view, the designer can also deliberately select constraints as boundaries within which creative solutions will be sought, narrowing the focus to designs only within those bounds. Stokes (2006) proposes this as the method used by many art and design masters to stimulate breakthroughs that lead to new schools of design thinking.
- Design as *search*. Simon (1999) describes design as a search in which missing means-end relations are sought. Design consists in part of looking for the contents of a black box whose inputs and outputs are known but whose contents can take many forms. Tsourikov (Garfinkel, online) has embodied this approach in "invention machine" software that searches a large pool of means-end mechanisms in terms of inputs and outputs required by a specific problem to return all known mechanisms that can potentially be made to fill the box.
- Design as the *application or assembly of patterns* (Alexander, 1979). Alexander describes architectural design in terms of the assembly of elements from an abstract pattern language. Designs using these languages contain structures

matched to the activities of those who will use them and bring a "living" quality to the design. Polanyi (1958) describes the existence of "operational principles" that embody the abstract mechanism of a technological artifact.

- Design as a *social process* that involves defining common terms, forming goals, and reaching consensus (Bucciarelli, 1994). "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).
- Design as *engineering*. Vincenti (1990) describes an engineering process that requires the application of several categories of technological knowledge, including abstract concepts, material and mechanism behavior data, and practical designing know-how. To the degree the designer possesses all of these kinds of knowledge at some level, designs become possible. Design can be pursued as a method for creating new shared, public knowledge as well as being a routine pursuit for the creation of new designs.
- Design as *prototyping and iteration*. Schrage (1999) describes the progressive invention of products and systems through constant iteration in modeling and testing. Problems become understood in the process of modeling, and fit with context becomes understood through rapid cycles of testing and revision. The spreadsheet is one of the most powerful computer tools for modeling business solutions because multiple scenarios can be generated and tested in principle in extremely short cycle times. Scenario generation encourages the designer to explore the principles that generate solutions in order to avoid random brute combinatorics.
- Design as *tinkering* with the application of *problem solving methods*. Through knowledge of a wide spectrum of design techniques, designers can advance toward solution, changing techniques as the needs of the solving process change with the stage of solution (Jones, 1992).
- Design as the *application of process or category system*. Design can be made accessible to a larger audience of designers by formulating sound principles into process descriptions (Branson & Grow, 1987) or by simplifying understanding of complex phenomena using taxonomic category systems (Gagné, 1985).

Each of these metaphors affords the designer a window into the heart of designing, but none of them captures the entire experience of designing.

The dominant metaphor of instructional design for the largest number of designers for nearly 40 years has been the application of processes and category systems. However, recently the search for alternative approaches has intensified.

Functional Design

Our design theory is expressed as a set of propositions below. A full treatment of this approach to design and its relation to an architecture of instructional and instructional design theories is given in Gibbons and Rogers (Gibbons & Rogers, Submitted).

1. *Design problem solving involves the decomposition of a design problem into sub-problems of solvable size*. A major part of designing involves clarification of the design problem itself. Once the problem is identified, or as intermediate steps toward the problem are identified, the task of designing becomes to decompose the complete problem into sub-problems of solvable size. This principle is expressed in the work of Simon (1999), Schön (1987), Alexander (1979), and others in diverse design fields and is implicit in virtually all current instructional design models.
2. *It is possible to consider a principle of problem decomposition that decomposes the design problem in terms of the functions to be carried out by the type of artifact being designed.* Schön (1987) describes architectural designs in terms of sub-problem "domains" that represent major functional divisions of the entire design problem. Gibbons (2003) describes a set of functional problems that are solved in the design of the general case of instructional design problems:

   - Content – The designer may specify the nature of the elements used as divisions of the subject-matter. Numerous theories of content structure can be found in theories of instructional design, including the content divisions of cognitive apprenticeship (Collins, Brown, & Newman, 1989), the objectives' taxonomies of Gagné (1985) and Merrill (Merrill & Twitchell, 1994), the description of semantic nets (Carbonell, 1970), and the knowledge structure theory of John R. Anderson (1993). Jonassen, Tessmer, and Hannum (1999) and Gibbons (1977) identified and classified numerous pre-design analysis methodologies. Each methodology can be seen as an expression of a theory about the structure of learnable content.
   - Strategy – The designer may specify space, time, event, social, and interaction structures that define occasions through which the learner can experience the content structures. The content theorists named above (who represent only a sub-set of all of the theorists who could be named) all have in common the fact that they have made specification

of a theory of strategy structures through which their theory's content structures can be experienced.

- Control – The designer may specify a set of controls (in fact, a limited language) in which communications can be expressed from the learner to the instructional source. This may include a defined language of mouse positions and clicks, a menu language, a limited set of verbal terms, or gestures and natural language expressions. Directly manipulable interfaces such as simulators and virtual worlds pose a special challenge in this respect. Design languages for the creation of control systems receive relatively little attention in the literature, except in the work on the subject of interaction like that of Crawford (2003), but a visit to Google Earth © demonstrates the great variety of control communications possible and desirable. Increasing reliance on non-sequenced forms of instructional strategies like those found in instructional simulations will require theorists to address the subject of control systems more fully in the future.
- Message – Intelligent tutoring systems, and in the future, intelligent Web-based object systems that must compose some portion of the instructional message adaptively at the time of use will have to deal with the structures of message generation and construction systems. They will confront the problem of defining the structural properties of message-making algorithms. Message structures are only abstractions or tokens, as are all of the structures described up to this point. But these abstract structures all converge on the representation domain (layer), which is described next. Examples of discussions of design languages related to messaging systems can be found in the literature of intelligent tutoring (see, for example, Clancey & Shortliffe, 1984).
- Representation – The designer may specify a set of rules for converting abstract structures from all of the other domains (layers) into representations that have sensory properties. In addition, the rules must specify the coordination and synchronization of message elements delivered in different media formats (Mayer, 2001). Design languages for composing and creating representations themselves are so numerous and so present in the literature of instructional design as to prohibit exhaustive enumeration. A design language related to a particular type of information-rich graphical illustration can be seen in the books and Web resources created by Tufte (Tufte, online). Representation design languages are by far the most numerous and detailed because the representation domain (layer) of designs is the first non-abstract domain and the one most noticed by even the most inexperienced instructional designers.
- Media-logic – The designer may specify the structures of execution by which representations are enacted for the learner. Live instructors require directions for conducting interactions, for posing and managing the environment for solving problems, and for providing coaching, feedback, and other support functions of instruction. Automated delivery systems also require directions and logic for providing the same functions. Design

languages for media-logic can bee seen partly in development software manuals. Design languages for live instructor activities are currently conflated theories of coaching, interaction management, problem-based instruction tutoring, and so forth and deal with the *manner and means* of execution of basic instruction-related acts by the instructor.

- Data management – The designer may specify data structures and data processes for the collection, storage, analysis, interpretation, and reporting of data resulting from learner interactions. It is these data that make possible adaptivity of instruction to the individual. Design languages for this complex domain are also described in the literature of intelligent tutoring systems.

3. *Decomposition of design problems in terms of artifact functions and sub-functions (domains, layers) allows the designer to concentrate on sub-problem solutions in detail while at the same time relating sub-problem solutions to the whole*. According to Schön (1987):

> Each move [tentative decision during design] has consequences described and evaluated in terms drawn from one or more design domains. Each has implications binding on later moves. Each creates new problems to be described and solved. Quist [a designer] designs by spinning out a web of moves, consequences, implications, appreciations, and further moves. (p. 57)

4. *Within each sub-problem domain, additional decomposition is possible*. This method of problem localization promotes design modularization. Baldwin and Clark (2000) describe the application of functional design decomposition and its implications for modularization of designs. Their tracing of the history of computer designs is a story of increasing modularity that Baldwin and Clark claim is the foundation of the economic success of the modern computer industry and the key factor responsible for its explosive growth. Brand (1994) illustrates the practical benefits of this kind of design modularization by describing contrasting styles of building design: a style in which design layers are modularized and can change over time independently, and a less-desirable style in which design layers are intermixed and in which changes to one part of the design can result in the destruction of other parts as well. At present, most instructional designs inter-mix design domains in such a way that graceful aging of individual layers of the design and their independent replacement is impossible. Consequently, most instructional designs have a short half-life and are difficult and costly to modify.
5. *For each domain or layer of the design thus defined, many design languages already exist, and new design languages can be created that provide terms appropriate to the solution of sub-problems within that domain*. Schön (1987) describes:

> It is not difficult to see how a design process of this form might underlie differences of language and style associated with the various schools of architecture. Designers might differ, for example, in the priorities they assign to design domains at various stages of

> the process. They might focus less on the global geometry of buildings, as Quist does, than on the site or on the properties and potentials of materials. They might let the design depend more heavily on the formal implications of construction modules. Their governing images might be framed in terms of building character, and they might allow particular precedents to influence more frankly the order they impose on the site. But whatever their differences of languages, priorities, images, styles, and precedents, they are likely to find themselves, like Quist, in a situation of complexity and uncertainty that demands the imposition of an order. From whatever sources they draw such an initial discipline, they will treat its imposition on the site as a global experiment whose results will be only dimly apparent in the early stages of the process.
>
> (p. 65)

Benefits to the Design Process

Benefits of many kinds accrue to the design methodology from the above view of design architecture. According to Gibbons and Rogers (Gibbons & Rogers, Submitted):

> The specific layers of a design evolve and change based on their utility to the designer, according to a number of factors that include design constraints, design criteria, resources, tools, new technology, new construction methods, available designer skills, and designer awareness. Each design includes its own unique combination of layers at the most detailed level. At the most detailed level, layers are created or destroyed according to the decisions and dynamics of a given project.

In general, the design process within this view comes to consist of design by successive constraint placement (Gross et al., 1987; Stokes, 2006) more than it is adherence to a process. Despite process views of design which represent the prevailing doctrine, successive constraint placement is probably a more accurate description anyway of what actually takes place in design projects. An example may show why this view is almost inescapable.

It is common for design problems to be given to a designer by a client with particular constraints already imposed. The client may specify a particular delivery medium or may require a particular instructional strategy to be used. This corresponds to certain commitments having been made within certain design domains (layers) as we described them above. A process approach to design would dictate design steps to be taken in a systematic fashion to complete the project.

However, without violating the general outlines of the process approach, it is easy to see that an examination of the design problem in the light of domains or layers of the problem reveals layers within which those constraints are placed. It is possible using the interaction of layer designs described by Schön above to trace the implications of these existing constraints to design decisions among the individual layers. Certain decisions that would normally be executed during a process may be already constrained, whereas other decisions (some involving process steps) may be made irrelevant. Each constraint placed, either by pre-decision or by the designer's own decision, cascades to constrain the decisions that are necessary or possible within other layers. Therefore, selection of a video medium will have the immediate effect of imposing constraints on some strategy, message, control, representation, media-logic, and data management layer decisions. If the computer medium is pre-selected by constraint, different constraints are rippled to the other layers, and a completely different set of decisions is presented to the designer.

This interaction of layer decisions is an important factor in everyday design-making that process approaches neglect. Designing in terms of layers allows the designer not only to apply traditional processes as they are appropriate, but also to reconfigure the design process to the requirements of the individual project. It allows the designer to tailor an order of decision-making within and between the layers and sub-layers that might be somewhat different than what might be dictated by standard process models.

Other practical benefits that accrue from the layered approach are as follows:

- Modularization of the design itself – Brand's description of building designs (Brand, 1994) demonstrates the value of creating designs in layers that are relatively independent, interacting only at defined interface points. Different layers are allowed to age at different rates, and changes or updates to one layer can be made without destroying the functions of other layers.
- Sharing of layer-related design tasks among design team specialists – As delivery systems increase in sophistication, so do the possible designs for the medium, and so does the range of skills required to design and develop a product. Early involvement of disparate skill sets into a design project is highly desirable to take advantage of judgment and design conceptions that involve specialized areas of the design. Considering the design as a layered structure more clearly defines tasks that can be assigned to specialists, enabling them to participate earlier in a way that balances attention to detailed decisions with concerns for the integration of the whole design.
- Identification of the minimum design decisions essential to allow early initiation of prototyping and testing cycles – Schrage (1999) describes the value of early and rapidly cycled prototyping as a means of refining the design problem and fitting design solutions within their context of use.

Benefits in the Application of Theory to Designs

Gibbons and Rogers (Gibbons & Rogers, Submitted) describe another benefit of conceiving the architecture of designs in terms of layers and the design languages that are related to them:

> Within the context of this view of [layered] instructional design theory, we propose that an instructional theory can be described as a set of specialized, mutually-consistent design languages containing defined terms that are distributed across multiple design layers. This insight unifies the concepts of design layers and instructional theory and, more importantly, shows the relationship between instructional design theory and

> instructional theory. Design theory provides the structural framework within which specific instructional theories can be analyzed and compared. Instructional theories dwell within a framework of layers...

Instructional theories can be viewed as collections of reserved terms selected or created by theorists – terms that pertain to elements used within different domains (layers) by that theorist for the solution of design problems. For instance, the primary categories of cognitive apprenticeship described by Collins et al. (1989) are Content, Method, Sociology, and Sequence. They define the Content category in terms of four types of knowledge whose differing structures designers should consider facilitating with their designs. This clearly corresponds to the content layer described in design layering theory. The remaining categories specify structures a designer would arrange so that interactions, social context, and ordering of interaction events can provide the learner access to and beneficial experience with the content elements. These three categories therefore correspond to three different sub-layers within the strategy layer.

Likewise, across 20 years and 4 editions of *The Conditions of Learning* Gagné described 4 versions of a learning objective taxonomy. Each of the categories in the taxonomy suggests to the designer a type of learning content. Therefore, Gagné's taxonomy – and the numerous other taxonomies created during the same time period – represents content layer design languages, and each type within the taxonomy represents a design language term. In addition, Gagné's nine events of learning represent structures for segmentation of instructional experience. They are strategic structures, and the events represent terms in a strategic design language.

These examples show that the theory of design layering can provide a common framework for analyzing, examining, and comparing instructional theories. Using it, it quickly becomes evident that different theorists choose certain layers as key focal points for their theories and minimize or remain silent about terms that might be associated with some others.

Moreover, it also becomes evident that theorists can concentrate exclusively on a particular layer: Mayer (2001) establishes a set of privileged terms within the representation design layer. Crawford (2003) does the same for the control layer, at the same time describing useful terms that can be associated with the strategy layer. The anthology of classroom observation category systems compiled by Simon and Boyer (1974) can be seen in this light as a collection of taxonomies containing design language terms associated with several layers pertinent to the description of live instruction. As already mentioned, software manuals for development tools contain the terms of the design language associated with the particular tool. A study of the terms contained in a single manual of this type reveals that there are actually several sub-languages associated with every tool, including a language made up of terms describing the development environment itself (for instance, "stage" and "actor"), one describing the elements manipulated and arranged within that environment (for instance, “library object” and “property”) to form a product, and usually one describing special scripting languages that function within the context provided by the tool’s controlling metaphor. These terms must be matched with the designer’s terms that describe actions and activities that fall outside the province of the language of the tool but which are necessary for the creation of the designer’s intended interactions (demonstrations, practice occasions, etc.).

Because the terms of design languages can be related to both practical decisions during design and theoretical structures, the principle of design languages explains one way in which theory enters designs. Rather than expecting a single instructional theory to provide an adequate basis for an entire design, we should expect that numerous theories might be called upon. Indeed, if a designer normally makes decisions within most or all of the layers for a given design, the designer can justify the design at a much more detailed level than was possible before. With analysis at this level of detail, it may become apparent which areas of a particular design have a sound theory basis and which do not. Design layers also provide a framework that allows the designer to select and integrate theories that pertain to different layers, mixing and matching them and becoming aware of which theories are compatible and which amplify the effects of which others. Researchers should find that exploration of the layers and correlation of existing theories with them will identify areas of design for which additional theoretical attention is needed.

Conclusion

We have outlined a theory of design architecture that we feel represents a step forward toward our ability to describe what happens, or rather what can happen, at the moment of design. As we proposed at the beginning, we believe this approach can be used to improve the consistency and quality of designs, facilitate communication about designs and theories, and allow designers to work efficiently in design teams and with greater mutual understanding. It may suggest functionalities for advanced design tools and may also be a useful framework for introducing new designers to the languages of design they will use.

References

Alexander, C. (1979). *The timeless way of building*. New York: Oxford University Press.

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

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

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

Branson, R. K., & Grow, G. (1987). Instructional systems development. In R. M. Gagné (Ed.), *Instructional technology: Foundations* . Hillsdale, NJ: Lawrence Erlbaum Associates.

Bucciarelli, L. L. (1994). *Designing engineers*. Cambridge, MA: MIT Press.
Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer-aided instruction. *IEEE Transactions on Man-Machine Systems*, *11*: 190–202.
Clancey, W. J., & Shortliffe, E. H. (Eds.). (1984). *Readings in medical AI: The first decade*. Reading, MA: Addison-Wesley.
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: Lawrence Erlbaum Associates.
Crawford, C. (2003). *The art of interactive design*. San Francisco: No Starch Press.
Gagné, R. M. (1985). *The conditions of learning* (4th ed.). New York: Holt, Rinehart & Winston.
Garfinkel, S. (2006, April 15). Genius minus perspiration. *MIT Technology Review (online)*. Retrieved from cache.boston.com/technology/tr/september/benchmark1.shtml.
Gibbons, A. S. (1977). *A review of content and task analysis methodology* (Technology Report #2). San Diego, CA: Courseware, Inc.
Gibbons, A. S. (2003). What and how do designers design? A theory of design structure. *Tech Trends*, *47*(5), 22–27.
Gibbons, A. S., & Rogers, P. C. (Submitted). The Architecture of Instructional Theory. In C. M. Reigeluth & A. Carr-Chellman (Eds.), *Instructional-design theories and models, volume III*. Mahwah, NJ: Lawrence Erlbaum Associates.
Google Earth. (2006, April 15). Retrieved from earth.google.com.
Gross, M., Ervin, S., Anderson, J., & Fleisher, A. (1987). Designing with constraints. In Y. E. Kalay (Ed.), *Computability of design*. New York: John Wiley & Sons.
Jonassen, D. H., Tessmer, M., & Hannum, W. H. (1999). *Task analysis methods for instructional design*. Mahwah, NJ: Lawrence Erlbaum Associates.
Jones, J. C. (1992). *Design methods* (2nd ed.). New York: Van Nostrand Reinhold.
Mayer, R. E. (2001). *Multimedia learning*. Cambridge, UK: Cambridge University Press.
Merrill, M. D., & Twitchell, D. G. (Eds.). (1994). *Instructional design theory*. Englewood Cliffs, NJ: Educational Technology Publications.
Polanyi, M. (1958). *Personal knowledge: Towards a post-critical philosophy*. New York: Harper Torchbooks.
Schön, D. A. (1987). *Educating the reflective practitioner*. San Francisco: Jossey-Bass Publishers.
Schrage, M. (1999). *Serious play: How the world's best companies simulate to innovate*. Boston: Harvard Business School Press.
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, Inc.
Simon, H. A. (1999). *The sciences of the artificial* (3rd ed.). Cambridge, MA: MIT Press.
Stokes, P. D. (2006). *Creativity from constraints: The psychology of breakthrough*. New York: Springer Publishing Company.
Tufte, E. (2006, April 15). Retrieved from www.edwardtufte.com/tufte/.
Vincenti, W. G. (1990). *What engineers know and how they know it: Analytical studies from aeronautical history*. Baltimore: Johns Hopkins University Press.

Springer

http://www.springer.com/978-0-387-09666-7

Learning and Instructional Technologies for the 21st Century
Visions of the Future
Moller, L.; Harvey, D.M. (Eds.)
2009, XVI, 232 p. 20 illus., Hardcover
ISBN: 978-0-387-09666-7