

## A Selection of Quotations on Design Languages and Their Relation to Design Layers

Andy Gibbons and Clint Rogers  
Instructional Psychology and Technology Department  
Brigham Young University

### Purpose

Our main idea is that designers can profit by looking into the nature of designs. Each instructional design shares aspects in common with other instructional designs, and we have tried to identify two of these commonalities, described below under two main headings: design languages, and design layers. We will try to give substance to each of these ideas and show the useful relationship between them.

We caution the reader that these ideas are not scientific, nor are they subject to demonstration by scientific means, so far as we can tell. Instead, these ideas are technological in nature: they describe generative principles that a designer can use in design synthesis. Technological principles (we will refer to them as technological theories or design-related theories) are not *true*, nor are they *mutually exclusive*. A designer may call upon multiple technological principles (theories) in the making of a single design. The existence of one technological principle does not invalidate the existence of other technological principles in the same way that one scientific theory tries to eliminate others from consideration. (See Gibbons, 2003b for a discussion of this view.)

The reason for this caution is that the discussion may try to slip into the scientific “proof” metaphor, which will cause the discussion to stop making sense. Our goal is to explain the utility of thinking of design in terms of design languages and design layers. We will feel successful if these ideas allow you to think of designing in a way that improves the cost of designing, the quality of designs, or the time required to design; and we don’t want you to give up ideas that already work for you.

The most economical way for us to proceed is to provide key statements from our existing work that express the gist of the argument. The reader may find that this initial statement needs more explanation, and in the discussion that follows during the week we will try to provide answers.

### Design Languages

“Design languages, formal or intuitive, lie at the heart of all design and development processes and tools. Instructional designers tend to be unaware of the multiplicity of design languages they use, because in most fields the use of design languages to improve precision and productivity is relatively new. However, the identification and use of design languages in many design and manufacturing fields has greatly benefited the growth and maturation of many

fields over a very short span of years. Instructional design will also benefit from this trend as designers and theorists become conscious of the existence and application of design languages and their related notation systems.” (Gibbons & Brewer 2005, p. 111).

A design language is a set of abstractions used to give structure, properties, and texture to solutions of design problems. Design languages provide building blocks for designs. Design languages provide categories for thinking about design problems. Design languages provide an important link between technological theory and design practice. Design language terms signify objects to be acted upon, actors, actions, concepts, types of relation, composite objects, qualities, and properties.

“Consider the following scenario illustrating a typical design situation:

*“As instructional designer you are asked to support a communication specialist in the design of a **blended learning course** in effective communication skills for a company’s workforce. At a meeting you and the communication specialist are discussing **lectures, discussions, case studies, exercises, online discussion forums**, and other possible **activities**. As the discussion proceeds, you realize that the communication specialist does not recognize some of these terms. Further, you realize that the definitions of the terms that you do both recognize differ, sometimes in significant ways. You begin to find it hard to discuss design details because you are not yet working from a common framework of meaning.”*

“The designer and the communications specialist need a shared set of terms for shaping their design that will be understood in roughly the same way by both. The designer and the communications specialist will draw on personal and public design languages from each’s own training and experience, and new terms will be invented within the scope of the project. A shared language will evolve through a negotiation process common to design in every field (Winograd & Flores, 1987). Notation systems that include drawings, symbols, and words will also evolve to publicly represent the design. This notational expression will be directed toward a specialized audience whose members know the language and its notation well enough to understand the implications of the design for manufacture. The design documentation will therefore be rich and dense and difficult for non-team members to interpret.” (Gibbons, Botturi, Boot, & Nelson, in press).

We should emphasize that what will result from this negotiation will be a project-specific design language that has been worked out by the design team. Within this language will be terms that are commonly used by designers, but the negotiated meaning will be the project’s own, and the meanings that evolve for the project will have much more agreed detail than the general-use terms. Our

common design languages, therefore, are generalized abstractions that have a certain somewhat-vague structure and lack detail. That detail is added through the negotiations of individual teams as they try to find ways to use the patterns supplied by the abstractions to specific design problems.

The building block metaphor does not refer to Lego™-like blocks that simply join together mechanically: it refers to abstractions that can be combined together into a design in a way that the identity of a single abstraction can be lost but still act as a modifier to the whole design. We will try to make this idea more clear later on. Christopher Alexander (1979) provides an example of how terms from a language of design patterns can be combined to create a house design. This theme has been taken up in some parts of the software industry (Gamma, Helm, Johnson & Vlissides, 1995; Gabriel, 1996; Van Duyne, Landay & Hong, 2003).

“It is a truism that designers do not invent anew the primitive elements of their designs each time they make one. Instead, they work with a familiar set of primitive structures at some level of granularity that they tend to use and reuse from design to design. Whether the designer is conscious of using these primitive building blocks or not, they constitute the primitive terms of that designer’s design languages. What is less often the case is that designers are conscious of *multiple* such languages that they use and decompose their design problems into sub-problems in terms of the different languages they hold.” (Gibbons 2003a)

### Design Layers

Most instructional designers do not think of their designs in this way, and the literature on instructional design does not teach it. The problem that asserts itself when a designer does begin to think in terms of design languages is that it becomes apparent that there are literally hundreds of languages that every designer uses: most of them implicit languages embedded in habitual or traditional design practices.

Schön (1987) describes the organization of these languages into “domains”. Each domain represents a particular design sub-problem to be solved, and the languages of the domain—which may be many and varied—supply different sets of constructs that can be used in framing a solution.

“Table 1 below provides a sampling of the many domains Schön names that are related to the solution of a problem in architectural design.

Table 1. Schön’s domains of an architectural design (excerpted from Schön, 1987).

<b>Domain</b>	<b>Definition</b>	<b>Examples</b>
Siting	Features, elements, relations of the building site	“Land contour”, “slope”, “hill”, “gully”
Organization of space	Kinds of space and relation of spaces to one another	“A general pass-through”, “outside/outside”, “layout”
Form	1. Shape of building or component 2. Geometry 3. Markings of an organization of space 4. Experienced felt-path of movement through a building	“Hard-edged block”  “A geometry of parallels” “Marks a level of difference from here to here” “Carry the gallery through and look down into here, which is nice”
Structure/technology	Structures, technologies, and processes used in building	“A construction module for these classrooms”
Building character	Kind of building, as sign of style or mode of building	“Warehouse”, “hangar”, beach cottage” ...
Building elements	Buildings or components of buildings	“Gym”, “kindergarten”, “ramp”, “wall”, “roof”, “steps”

Notice that the right-hand column (which is Schön’s) contains an assortment of terms used within the domain to supply solution constructs. These are a very small sampling of terms from the design languages that pertain to each domain. Schön describes the design community’s relationship to languages in this way:

*Aspiring members of the linguistic community of design learn to detect multiple references [language terms], distinguish particular meanings in context, and use multiple references [language terms] as an aid to vision across design domains. (Schön, 1987, p. 61)*

“Brand (1994) also shows how the design of buildings may be approached in terms of separate but connected and aligned ‘layers’. A similar layered structure of designs can be found in many other design fields as well, not because it has been invented there, but because it is the nature of designs to be so structured (Schön, 1987).” (Gibbons, 2003).

Layered designs can be found everywhere: basically in any design field. Computers are designed in terms of layers, as is the software that operates on the computer. Originally software was physically built-in to computer hardware. It was a major technological discovery that the software structures could be abstracted away from the hardware. The layer principle is inherent in the modular structure and economics of the digital world described by Baldwin and Clark (2000) in their book *Design Rules: The Power of Modularity*.

Approaching design problems in terms of domains (layers) and the languages that pertain to them opens up a new way of thinking to the designer. It means that design problems can be addressed in terms of the functionalities of the thing being designed, rather than in terms of the process used to design it.

Instructional designs have many things in common:

- Each design defines a particular body of *content* that is *structured* in a particular way. Jonassen’s book *Task Analysis Methods for Instructional Design* (with Tessmer and Hannum, 1999), for instance, describes many different structuring principles by which instructional content can be organized and captured. These provide the structural basis for the design languages of the *Content Layer* of an instructional design.
- Each design employs deliberately-chosen *strategies* for bringing the learner into experiences with the content structures. A design typically describes the siting of instruction, its setting, its social roles and responsibilities, the segmenting of time, the grouping and sequencing of content-related events, perhaps a cover story, and patterns of interactive “conversation” that can occur between the learner and the instruction (whether live or mediated). We call these concerns the domain of the *Strategy Layer*, and there are multitude languages that pertain to structuring within this domain.
- Each design includes either implicitly or explicitly a set of *controls* by which the learner can communicate into the instructional experience. Controls constitute a lingo in which the learner makes requests for information, gives commands, expresses choices, asks questions, gives answers, and a number of other learner-determined acts. We call this domain of an instructional design and all of the design languages pertaining to it the *Control Layer*.
- Each design includes a plan for structuring messages to the learner. These are not messages in text or graphic form, but in the form of abstract intentions to communicate. These intentions at the *Message Layer* are

expressed in message design languages that are then translatable into expressions in many different media or living surface forms.

- Each design therefore requires a *Representation Layer* that gives expression to message units in a form that can be sensed—audio, visual, kinesthetic, haptic, olfactory, etc. A single message unit (<That’s correct>) can be represented in many different forms (green highlight appears on display, “ding” sound, text on display, voice saying the words “That’s correct!”, etc.). Therefore, the mapping of message units onto media representations constitutes the outward-directed channel of communication to the learner from the instruction. The Control Layer and the Message and Representation Layers together carry out the functions of two-way communication and conversation between the learner and the instruction.
- Each design includes a set of instructions for carrying out the media representations in a particular order and with particular timing and synchronization. Whether the order of media events is pre-determined and fixed or decided at the time of instruction, the *Media-Logic Layer* of the design defines the rules for executing it. Live instructors require this information as well as computers.
- Finally, each design includes directions for the capture, storage, analysis, reporting, and interpretation of data generated by an instructional encounter. This data is used in making future decisions concerning the path of instructional events—either as the learner examines personal data and makes decisions or as the instruction makes choices based on discerned patterns of learning. This domain of design we call the *Data Management Layer*.

The functions we have listed are carried out by virtually all instruction through a combination of human and technological means and therefore must either be designed or defaulted. One of the purposes of describing design languages and design layers (domains) is to bring more of our design decisions out of the default category and into a public process, which may be totally human- or team-determined or in which some of the design decisions can be aided by automated means.

To describe what we see as the benefits of layer-language thinking, we will resort to quoting again:

“Several benefits may be expected from [design language] research:

- Improved tools for the support of all stages of instructional design, from initial conception to documentation of the completed design.
- Clearer relationships between design decisions and structures, features, and qualities of designs.
- Improved techniques for facilitating the formation of instructional design teams and their rapid movement toward productive activity.

- Improved approaches for the training of new instructional designers.
- A broader view of new designers with respect to the spectrum of design languages used by the many specialized members of their design teams.
- New understanding of the nature of instructional designing and new processes that tailor the design process to the design problem.”
- Insight into the nature of instructional theory, instructional design theory, and their relationship. (Gibbons et al, in press)

Andy Gibbons

<http://webpub.byu.net/asg33/index.html>

## References

- Alexander, C. (1979). *The timeless way of building*. New York: Oxford University Press.
- 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.
- Gabriel, R. P. (1996). *Patterns of software: Tales from the software community*. New York: Oxford University Press.
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Reading, MA: Addison-Wesley.
- Gibbons, A., Botturi, L., Boot, E. & Nelson, J. (in press). Design languages. In M. Driscoll, M. D. Merrill, J. van Merriënboer and J. M. Spector (Eds.), *Handbook for research for educational communications and technology (3rd ed.)*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Gibbons, A. S. (2003a). Learning objects and the properties of a design [Title changed from printed program]. Paper presented at the Annual meeting of the American Educational Research Association, Chicago, April.
- Gibbons, A. S. (2003a). The practice of instructional technology: Science and technology. *Educational Technology* **43**(5): 11-16.
- Gibbons, A. S. & Brewer, E. K. (2005). Elementary principles of design languages and notation systems for instructional design. In J. M. Spector, C. Ohrazda, A. Van Schaack and D. Wiley (Eds.), *Innovations in instructional technology: Essays in honor of M. David Merrill*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Jonassen, D. H., Tessmer, M. & Hannum, W. (1999). *Task analysis methods for instructional design*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Schön, D. A. (1987). *Educating the reflective practitioner*. San Francisco, CA: Jossey-Bass Publishers.
- Van Duyne, D., Landay, J. & Hong, J. (2003). *The design of sites: Patterns, procedures, and processes for crafting a customer-centered Web experience*. Boston, MA: Addison-Wesley.
- Winograd, T. & Flores, F. (1987). *Understanding computers and cognition: A new foundation for design*. Reading, MA: Addison-Wesley.

## Other reading

- Gibbons, A. S. (2003). What and how do designers design?: A theory of design structure. *Tech Trends* **47**(5): 22-27.
- Gibbons, A. & Rogers, P. (2006). *Coming at design from a different angle: Functional design*. Paper presented at the Annual Research Symposium of the Association for Educational Research and Technology, Bloomington, IN, June.
- Gibbons, A. & Rogers, P. (in press). The architecture of instructional theory. In C. Reigeluth and A. Carr-Chellman (Eds.), *Instructional-design theories and models, Vol III*. Mahwah, NJ: Lawrence Erlbaum Associates.