

Learning to Love Software: A Bridge Between Theory and Practice

Ellen Lupton, Maryland Institute College of Art, Baltimore, MD, USA

Students love learning software, while faculty often avoid teaching it. This essay argues that a key way to reengage both students and faculty is to approach software as an analytical tool, a means of not only describing and generating meaningful form, but also synthesizing the practical goal of production with the theoretical goal of conceptual development. Software can thus be a bridge between theory and practice. Theory is by definition general: it is a description or model capable of sustaining its relevance across countless unique situations. Practice, on the other hand, is lived, grounded, and specific: it is the ongoing application of ideas to circumstances, a process that in turn reforms our models, warping and transforming our theories. Like theory, software is general, while practice is specific. Software exists prior to any practical use of it, and yet it is designed to anticipate those uses. Tools such as Photoshop, InDesign, and AfterEffects have cast a net around our field of practice, filtering our daily production of typography, symbols, images, and information systems. No theory or pedagogical practice yet exists to address the role these commercially developed and distributed digital technologies play in shaping and describing design's visual language.

Keywords: Bauhaus, digital media, graphic design, Photoshop, software

During the 1990s, cultural recycling and stylistic appropriation became standard operating procedures for graphic designers, inside and outside the classroom (Lupton & Miller, 1999). The dominance of digital tools encouraged this trend by making it easy to endlessly recirculate visual matter. Google searches became research, while cut-and-paste replaced sketch and synthesize. It seemed that the digital remaking of the design industries forced educators to focus energy on teaching software instead of on conceptual development. Many educators viewed software not only as the necessary evil that prepared a student for a design profession but also as a distraction from the honorable task of conceptual thinking. This pragmatic shift was accompanied, on the intellectual side, by the entrenchment of a postmodern world-view that favored

visual studies over form-based pedagogies inherited from the Bauhaus (Lupton & Miller, 1991). In graphic design programs as well as in foundation courses, exercises in parody and appropriation took the place of stale old studies concerning point, line, plane, and grid.

Yet while graphic design educators seemed to turn away from formal analysis towards a more impressionistic, culturally based and referential approach, professional software writers systematically organized image-making into menus of properties, parameters, and filters, converting the Bauhaus theory of visual language into comprehensive tools. Adobe Photoshop, for example, is a systematic study of the features of an image such as its contrast, size, and color model. Quark Xpress and Adobe InDesign are structured explorations of typography; they are software machines for determining formal interests such as leading, alignment, spacing, and column structures as well as image placement and page layout.

These tools have cast a net around our field of practice, filtering our daily production of typography, symbols, images, and information systems. No theory or pedagogical practice yet exists to address the role these commercially developed and distributed digital technologies play in shaping and describing design's visual language.

Ironically, students love learning software, while many faculty (especially those of us aged over 40) avoid teaching it. I argue that a key way to reengage both students and faculty is to approach software as an analytical tool, a means of not only describing and generating meaningful form, but also synthesizing the practical goal of production with the theoretical goal of conceptual development.

Software can thus be a bridge between theory and practice. Theory is by definition general: it is a description or model capable of sustaining its relevance across countless unique situations. Practice, on the other hand, is lived, grounded, and specific: it is the ongoing application of ideas to circumstances, a process that in turn reforms our models, warping and transforming our theories. Like theory, software is general, while practice is specific. Software exists prior to any practical use of it, and yet it is designed to anticipate those uses. Compelling works of design often implement software in unintended ways. For example, Kyle Cooper's (1996) opening titles for the film *The Island of Dr. Moreau* began as distortions of typographic outlines in Illustrator.

FROM BAUHAUS TO SOFTWARE

In the 1920s, faculty at the Bauhaus such as Kandinsky (Kandinsky et al., 1947) and Klee (1953), and other vanguard schools of art, analyzed visual form according to formal parameters: from point, line, and plane to color, texture, pattern, scale, and contrast. The idea of looking at two-dimensional design as a universal, perceptually based *language of vision* shaped design education around the world. Training students to recognize this language, or so the theory went, would help them create forms that everyone could understand (Lupton & Miller, 1991), because the language of the eye is grounded in universal human capacities (Arnheim, 1974). The Bauhaus Vorkurs or Basic Course remains the model – at least nominally – for first-year foundation programs around the world, which aim to expose students to visual concepts common to all the arts (Lupton & Miller, 1991; Wick & Grawe, 2000).

At the Bauhaus, faculty did not pursue these goals in a consistent, monolithic way. Johannes Itten (1975) accounted retrospectively for his Bauhaus teaching in the book *Design and form: The basic course at the Bauhaus and later*. As the first leader of the Bauhaus Basic Course, he invited students to experience colour, texture, and shape from a personal, sometimes mystical point of view. In contrast, the Hungarian-born Constructivist Laszlo Moholy-Nagy (1950), who taught at the Bauhaus from 1923 to 1928, believed that the Basic Course should lay bare a universal vocabulary ratified by a shared society and a common humanity (Margolin, 1998). Similarly, Josef Albers, who came to the Bauhaus as a student and then taught alongside Moholy-Nagy in the Basic Course, favoured systematic thinking over personal intuition, objectivity over emotion (Weber, 2004). The Russian émigré Wassily Kandinsky (Kandinsky et al., 1947) contributed theories of geometry

and colour through his teaching and his Bauhaus textbook, *Point and line to plane*, which called for the creation of a basic *dictionary of elements* and a universal “grammar” underlying all the arts.

The Bauhaus was the site of conflicting theories and practices. Whereas Kandinsky (Kandinsky et al., 1947) and Itten's (1975) teaching affirmed the ongoing tradition of painting, Moholy-Nagy and Albers forged the use of new media and new materials. Moholy-Nagy and Albers saw that art and design were being transformed by modern technologies, especially photography, film, and mass production. And yet their ideas were profoundly humanist, always asserting the role of the individual over the absolute authority of a system or method. Design for them is never reducible to its functional program or to a technical description (Lupton & Miller, 1991; Wick & Grawe, 2000).

Since the 1940s, numerous educators have refined and expanded on the systematic approach crystallized at the Bauhaus. They include Moholy-Nagy (1947), whose great work from this period is *Vision in Motion* and Gyorgy Kepes (1944) at the New Bauhaus in Chicago; Itten, Max Bill, and Gui Bonsieppe at the Ulm School in Germany (Spitz, 2002); Emil Ruder (1981) and Armin Hofmann (1965) in Switzerland; and the *new typographies* of Wolfgang Weingart (2000), Dan Friedman, and Katherine McCoy (1990, 1997, 1999a, 1999b, 2002) in Switzerland and the United States. These educators each articulated a structural approach to design from a distinct perspective.

Meanwhile, during the 1960s, postmodern artists and designers began rejecting the idea – cherished by the builders of the Bauhaus and other modernist institutions – that communication might have a universal basis. In part, postmodernism asserts that a cultural artifact can be understood only in terms of a specific place, time, and audience. This relativist position makes it futile to speak of any inherent meaning in an image or object, as all people will bring their own cultural biases and personal experiences to bear on the act of interpretation.

As postmodernism itself became a dominant ideology in the 1980s and 1990s, in both the academy and in the marketplace, the design process became mired in the act of referencing cultural styles or tailoring messages to narrowly defined communities of users (Lupton, 1996).

A POST-POSTMODERN REBELLION

It is now time to look back to the modernist undertaking in a manner informed by, but not beholden to, the postmodern experience – a look back that can use software as a tool of analysis. This perspective might bring with it a changed set of values to the design process. In the interest of exploring such a shift, my colleague Jennifer Cole Phillips and I at Maryland Institute College of Art (MICA) in Baltimore have, over the past two years, been exploring form-based exercises in design, at both the graduate and undergraduate levels.¹

We stepped back and noticed that while our students were adept at understanding design from a cultural point of view, they were less able to build concepts abstractly by, for example, manipulating elements such as scale, contrast, timing/sequence, hierarchy, grids, and diagrammatic structures. In order to strengthen their formal thinking, we started to explore the constructive languages of print and multimedia. The results of those efforts formed the basis of a book, *Graphic design: The new basics* (2008), which aims to introduce form-based thinking to design students in a manner informed by contemporary culture and technology. This essay details the theoretical ideas that led us to pursue this work with our students and to generate the book.

Recurring across Bauhaus design theory was the belief not only in a universal way of *describing* visual form, but also in a universal basis for its interpretation – a universal meaning. Reacting against that belief, postmodernism discredited formal analysis as a primary component of thinking and making in the visual arts. Formal study was tainted by its link to universalist ideologies. In contrast, our undertaking recognizes a difference between description and interpretation, between a potentially universal language of making and the limits on a universality of meaning. We also assert that form has intrinsic meanings in and of itself, which are not at all in need of a language-based interpretation. For example, we would argue that the pile of a carpet or the texture of a page does not require a semiotic reading, but instead engages us primarily through its tactility, depth, and resilience. Further, while an interface might use semantic icons such as folders and magnifying glasses, that interface also functions on a phenomenological level, supporting and responding to the intuitive, habituated actions of a user. Therefore, we argue that software tools provide designers with rich, comprehensive models of visual media, but the tools neither presume nor dictate a common basis of interpretation. It is the task of the designer, plunged into the specificities of practice, to make the language relevant to living situations such as audience, context, program, and site, delivering content-rich messages as

well as embodied experiences infused with material and structural qualities.

We are not alone in this post-postmodern rebellion. Stanley Fish (2005) teaches a controversial freshman composition course at Duke University that requires students to invent a new written language, complete with syntax and vocabulary. Bluntly forcing young writers to think exclusively about form and structure, Fish's course rejects subject matter – especially that which is politically tendentious. Fish believes that after learning to understand language as a structured system, his students will be more artful and deliberate when they return to writing English sentences. Many students, he points out, barely know what a sentence is when they arrive at Duke. Similarly, we argue that a close look at the formal properties embedded in software can lead to improved practice in specific situations.

Lev Manovich (2001, 2002, 2005, 2006, 2007) whose theories of new media inform this project at many levels, has sensed a second modernism in the air, as young designers and artists become interested in inventing new forms rather than sampling old ones. In *Remixability and modularity* (2005) he states:

It has become a cliché to say that we live in a 'remix culture' . . . What was referred to in post-modern times as quoting, appropriation, and pastiche no longer needs any special name. Now this is simply the basic logic of cultural production. (<http://www.manovich.net>)

Artist/designers including Future Farmers and Joshua Davis use software to design selforganizing works and to build collaborative tools and networks rather than to quote and appropriate existing images – better to build one's own media than to obsess about what already exists. Evidence of this shift doesn't end with production. As a critic, Manovich (2001, 2005, 2006, 2007) has worked to analyze new media from a formal and structural, rather than content-based, point of view. In his book, *The language of new media* (2001), Manovich states:

What is unique about how new media objects create the illusion of reality, address the viewer, and represent space and time? How do conventions and techniques of old media – such as the rectangular frame, mobile viewpoint, and montage – operate in new media? If we construct an archeology connecting new computer-based techniques of media creation with previous

techniques of representation and simulation, where should we locate the essential historical breaks? (p. 8)

In the aftermath of the Bauhaus, textbooks of basic design such as Itten's *Design and form* (1975), Hofmann's *Graphic design manual* (1965), and Dondos A. Dondis's *Primer of visual literacy* (1973), visual images were described using the elements of scale, contrast, movement, dimension, and balance. These elements make up a spatially organized *syntax of making* that designers use to organize a vocabulary based on point, line, plane, texture, and color. But other elements are also needed to flesh out our understanding of a making-oriented visual language today.

Take a look, for example, at transparency. Understood in a structural or formal way, transparency is a condition in which two or more surfaces are visible through each other. This condition usually yields distortion or interference in that both surfaces must be somewhat compromised in order to be visible simultaneously. In a state of absolute transparency, the closer surface simply disappears, ceasing to be evident to the eye. Transparency is a universal structural condition. We constantly experience transparent or semi-transparent effects in the physical environment: from water, glass, and smoke to Venetian blinds, slatted fences, and perforated screens. Graphic designers across the modern period have worked with transparency. In fact, it was defined and explored by Kepes in *Language of vision* (1944). But transparency has never been as explicitly and as easily available as it is right now via standard software tools.

Transparency enables the display of two images in the same space. A graphic designer can thus use transparency to create a composite of two pictures in order to suggest a conflict or synthesis of ideas such as East/ West, male/female, or old/new. Transparency is also a fundamental compositional device that carries constructive rather than semantic meaning. At the most basic level, every time a graphic designer puts a line of text on top of a picture, he or she is confronted with brute visual competition between the two layers. This condition prompts the designer to search for ways to either

preserve the visibility of both elements, or to allow one to dominate by manipulating elements such as colour, placement, shadows, outlines, and opacity.

The compression of multiple distinct forms into a shared shallow space has belonged to the vocabulary of architecture and decorative design for thousands of years (Rowe & Slutzky, 1993). Traditional patterns, such as plaid, use intersecting lines to build up contrasting areas of overlapping color. Linear elements in classical and modern architecture often appear to intersect, as elements such as columns or window moldings pass in front of or behind each other across a shallow field. As students become conscious of these modes of graphic transparency, they also become able to produce more visually complex and meaningful work (see Figure 1; typographic study by Alissa Faden; Figure 2, study in graphic and digital transparency by Yue Tuo).

Consider the role of transparency in motion-based work, where it is commonly used to design transitions. In place of a straight cut, the video editor can diminish the opacity of an image over time, producing a *fade to black*, or mix two semi-transparent images, a *cross-dissolve*. Such transitions affect the rhythm and style of the overall sequence. These transitions also modulate, in subtle ways, the message or content of the work. For example, when fading to black, the information from one scene is sharply separated from the next. But in a cross-dissolve, time and object relationships intertwine, allowing the possibility of narrative techniques such as foreshadowing. Yet transparency functions also as brute phenomenological fact. The straight cut is perceptually, formally distinct from a fade or a dissolve. This difference exists in motion-based works produced all around the world. Although viewers rarely stop to "interpret" or otherwise question these transitions, a filmmaker or an animator understands them as part their fundamental language of making.

In his essay *After effects, or velvet revolution* (2007), Manovich includes transparency along with compositing, layering, and hybridity as

If one sees two or more figures partly overlapping one another, and each of them claims for itself the common overlapped part, then one is confronted with a contradiction of spatial dimensions.

Figure 1.
Type study: Transparency, Alissa Faden, 2006. Reproduced in *Graphic design: The new basics* (Princeton Architectural Press).

new terms in an expanding dictionary of visual language. These principles are not in themselves new. Indeed, all were explored by designers in the avant-garde period via mechanical and light-chemical technologies. Compositing, the compression of multiple images onto a single surface, is seen in multiple-exposure photographic prints such as El Lissitzky's 1924 work titled, *Self-Portrait (Constructor)* (Tupitsyn, 1999). Layering, the separation of an image into overlapping components, has been a feature of colour printing processes for hundreds of years. Hybridity, the mixing of typographic, photographic, and/or linear images within a simultaneous frame or surface, was actively pursued in the posters, advertisements, and book designs of the twentieth-century avant-garde (Lupton & Miller, 1999).

In today's context, what makes layering, transparency, compositing, and hybridity *new again* is their ubiquitous accessibility through commonly available software. Thus, these tools, along with the effects these tools help to create, have become the universal. Manovich (2007) calls the revolution

"velvet" because it happened without being much remarked upon. Suddenly, we are here. In contrast, the historical avant-garde movements broke with history by exposing the language of production and inventing a new form of language that remains alive today. Even in the 1910s and 1920s, the technologies of mechanical reproduction were all well established. These technologies included halftone process (1880s), photography (1840s), lithography (1780s), and letterpress (1450s). Avantgarde artists and designers used these existing technologies in new ways, producing situations of disjunction, collision, and simultaneity (Rothschild, Lupton, & Goldstein, 1999).

Over the course of the 1990s, several software packages emerged as industry standards. Many of them are now consolidated in the product line of a single company, Adobe, whose Creative Suite is licensed and taught in countless schools around the world. Even outside the Creative Suite, many programs mirror its closely integrated interfaces, adopting what has become a common language. For example, AfterEffects has been called "Photoshop with a timeline", and InDesign has been called "Illustrator with pages". One can envision a day when one massive application will allow the authoring of all forms of media: still and moving, vector and bitmap, print and multimedia, 2D and 3D. That program might synthesize the most widely accepted features of various software programs, tested and refined over a period of time in response to user feedback and market forces. Such a software program could constitute a working description of a visual language that will, like verbal languages, continue to evolve. Learning that language of making, and learning to expand, overthrow, or replace it, will be a crucial part of both design education and progressive professional practice.

That future application, let us call it Ubershop, a synthesis of many applications, would probably contain foundational elements of Adobe's Photoshop, the first image editing software to

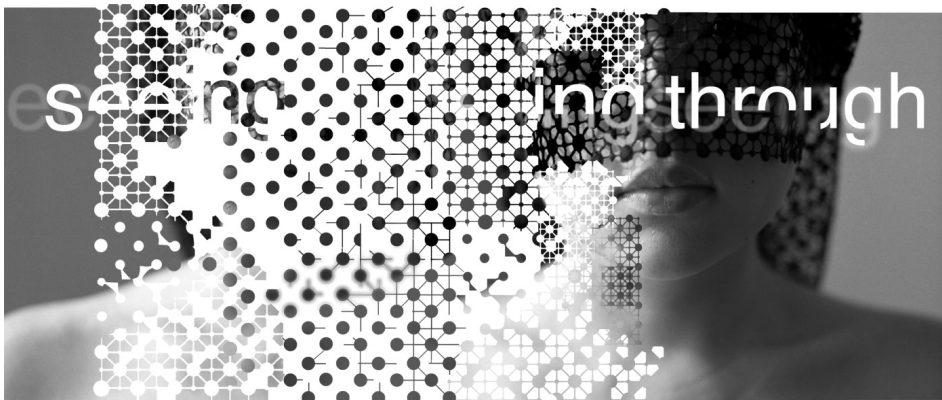


Figure 2.
Physical, Graphic, and Digital Transparency, Yue Tuo, 2007. Reproduced in *Graphic design: The new basics* (Princeton Architectural Press).

become culturally pervasive. Used by graphic designers, photographers, animators, architects, illustrators, Web designers, and millions of amateurs, Photoshop provides a basic model for many other applications. The verb “to photoshop” has entered the general vocabulary as slang, referring to any digital manipulation of an image, as in “Let’s photoshop that zit”.

Photoshop was invented by Thomas and John Knoll, the sons of an avid amateur photographer and college professor, who kept a darkroom in the family basement (Story, 2000). The two brothers produced an experimental application called Display in 1987, tackling such problems as how to represent gray scales on an Apple Macintosh Plus. The following year, they turned their research into ImagePro, a commercial product designed to manipulate and correct digital images that had been created in other software. Adobe licensed the product in 1989, releasing Photoshop 1.0 in February 1990 after 10 months of further development. From the beginning, Adobe marketed Photoshop as a general-use, rather than a professionals-only, tool.

The marquis, lasso, magic wand, and eye dropper, which date back to Photoshop’s earliest toolbars, represent Photoshop’s guiding principle: the ability to select specified elements in an image and perform changes on that selection. Photoshop’s early toolbars also included pencil, brush, and type tools for authoring marks within the program, similar to those used in various paint applications that were popular at the time. The act of *selection*, however, rather than creation, is at the heart of Photoshop, and it is this principle that made the product original and uniquely useful. Although Photoshop has evolved into an ever more powerful authoring tool, its origins in the alchemical darkroom – a place of dodging, burning, correcting, and balancing – remain central to its function.

A major advance in the Photoshop interface occurred in 1994, when the Layers feature separated the image in a new way, based not on colour relationships among pixels but rather on a sequence of actions performed over time.² Photoshop automatically creates a new layer whenever the user conducts certain actions, such as cut-and-paste or adding text. Layers enable parts of the image to be manipulated independently

of the rest. Any layer can be filtered, transformed, masked, multiplied, and so forth, and these activities can each be tweaked and reversed ad infinitum. Adjustment layers allow global features of the image, such as levels and curves, to be saved as separate sets of data, which can be revised or discarded at a later time. The source file becomes an archaeology of its own making, a stack of elements seen simultaneously in the main window, but represented as a vertical sedimentation in the Layers palette. The History palette, introduced in 1998, took the idea of Layers to a micro-level, making a record of nearly every action performed, and rendering the principle of layers explicitly temporal.

Layers appear today in nearly every graphics application, from Photoshop and Illustrator to Flash, FinalCut, and AfterEffects. Further, although layers have become a digital tool, their structure is based on a general application in the physical world, which speaks to historical methods of assembling images for reproduction. Most printing techniques require that a colour image be separated into at least four layers before it can be reproduced. Each colour requires its own stone, plate, film, screen, ink cartridge, or toner drum. While contemporary technologies automate this process, making it more or less invisible to the designer, the act of separating a printed work into layers remains a crucial phase of the production process. Prior to the early 1990s, *mechanicals* played a key role in the quality of the final printed piece. Mechanicals were made up of stacks of precisely aligned sheets of acetate, paper, and board that represented each colour or varnish, which the printers would work with as offset plates on the press. The designer or production artist adhered every element of the page, including type, and images or shapes, to the appropriate layer on the mechanical, so that any element touching or passing behind any other element was on its own distinct layer.

Historically, designers have tended to leave the layered character of the printed page invisible to the viewer, but experimental work has uncovered visual possibilities by making the layers themselves come forward as separate elements. The Swiss designer Wolfgang Weingart experimented extensively in this manner in the 1970s and ’80s, shifting the films used to create offset plates to produce unexpected



Figure 3.
La Lutte Continue (The Fight Continues),
Jan van Toorn, 1989.

textures and juxtapositions. The Dutch designer Jan van Toorn has used cut-and-paste techniques to create images whose complex surfaces suggest political action and unrest, as seen in Figure 3, where van Toorn has used a photomechanical process to mix layers of diverse media, from handmade to mass produced.

Many designers have explored an off-register or misprinted look, seeking rawness by exposing the layers of the printing process or the layered organization of the digital image file. Contemporary designers including Ryan McGuinness and Joshua Davis create graphic images composed of enormous numbers of layers that overlap in arbitrary, uncoordinated ways, allowing the layers to maintain separate identities in the final piece. Their work reveals an active engagement with how things are made, in both physical and virtual worlds. Experiments such as these show how an explicit concern with a software feature might also develop into a pedagogical tool aimed at concept development and visual thinking.

Instructors can help students explore the structures and metaphors needed to solve design problems by using features that are embedded in the tools we use. This way of teaching bridges theory and practice by approaching these pragmatic tools in a more critical and exploratory way. For example, a problem I developed for my Graphic Design II class uses layers to carry students from the physical world to the virtual one, ending in a time-based piece: the layers of a cut-paper collage ultimately become layers that change in time. The project is done first by hand, and then in Illustrator and Flash. Layers from Illustrator are imported directly into Flash for manipulation in the timeline. The beginning of that process is shown in Figure 4 (layers study by Kelly Horigan).³ I explain the process in this excerpt from the website.

1. Provide each designer with four sheets of coloured paper, 6-x-6-inches square. [We used origami paper]. Cut a square window into a larger sheet of paper so that you can move the coloured sheets around and experiment with different designs. Cut, shift, and overlap the elements to produce a collage that shows parts of all four layers.
2. Translate the collage into digital layers using Adobe Illustrator or another drawing application. Each physical layer becomes a separate layer in the digital file. Generate new compositions by digitally changing the colour, scale, transparency, orientation, and position of the digital layers.
3. Use your digital composition as a style frame for a sequential animation. Design a sequence, approximately ten seconds long, that loops: that is, it begins and ends on an identical frame. To plan your sequence, create a nine-panel storyboard.
4. Import your style frame into a digital animation program (Flash), distributing each layer of the style frame to a layer in the timeline to create strata that change over time. (<http://gdbasics.com/index.php?slayersdp3>)

LARGER IMPACT

Working with layers can foster a visual literacy of making in everyone, from children to professional art students. While 20 years ago the idea of visual literacy centred on how to read media in a critical way, today visual literacy centers on production, on learning the authoring tools. Writing has replaced reading. Moreover, those tools are shared across a mass global society. For example, when my daughter was six, I showed her how to build a

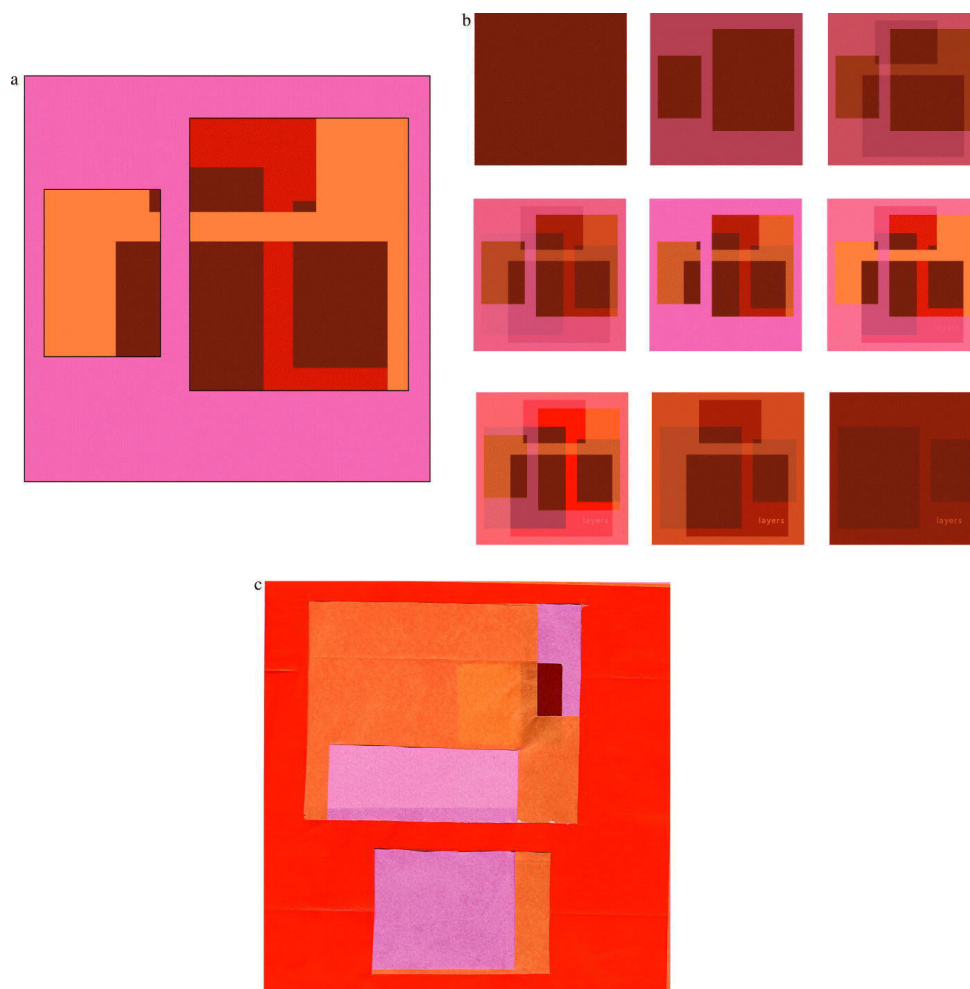


Figure 4.
Physical, Digital, and Temporal
Layers, Kelly Horrigan, 2006.
Reproduced in *Graphic design:
The new basics* (Princeton
Architectural Press).

face using Photoshop layers and then I taught her how to selectively turn them on and off to create different portraits. Labeling each layer (blue eyes, green eyes, and so on) helped her see the file as a database of assets. These same principles structure the Flashbased fashion and makeup games she plays on the Internet. Learning to see how these games are put together allows her to think more like a producer. She was soon assembling her own virtual paper dolls from Googled images. Like transparency, layering is a universal structural condition, occurring throughout the physical world and employed in the visual arts across history and cultures. Now, it is central to the increasingly pervasive language of digital media and production. It connects the virtual world with the physical one, and it connects time-based and still media.

Media critic Steven Johnson (2005) sees layering as part of the growing complexity of popular media. Describing *The Sopranos* in his book *Everything bad is good for you*, Johnson identifies *multiple threading* as the use of numerous parallel

but connected stories: “a single scene in *The Sopranos* will often connect to three different threads at the same time, layering one plot atop another” (p. 69). The wonderful graphic timelines in Johnson’s book compare the simple-minded TV crime dramas of the 1970s with those of today. Not surprisingly, Johnson’s timelines resemble those used in animation software. Looking at electronic games, television, and the Internet, Johnson urges us to ignore the content of these media, which is often violent or banal, and to look, instead, at their structure, which is layered and complex. An intricately plotted production such as *The Sopranos* or the game *Doom* challenges the mind to keep track of multiple threads, a process that Johnson thinks is making people smarter.

In his essay *Generation flash* (2002), Manovich argues that *software critique* is replacing *media critique*. The postmodern project unpacked the ideological baggage of mass media. Manovich notes that in the 1980s Barbara Kruger, Louise Lawler, and Richard Prince repurposed the archive of

mass communication and exposed the politics of representation. The new *software critique* is less politicointellectual and more technical, concerned not so much with what things mean as with how they are made. A rising generation is concerned less with challenging the systems of mass media than with building alternative communication networks. In that respect, Manovich argues that what we say is ceding to how we say it. Form has trumped content.

Manovich (2002) explains that the term *Generation flash* refers not to a specific software program but rather to a broader cultural movement. It is telling, nonetheless, that he plucked the name of his essay from the commercial realm (now part of the Adobe brand empire). Flash straddles the domains of the GUI toolbar and code-based authoring. It recalls the interface models of Photoshop and Illustrator while also speaking the language of code, albeit through its own idiosyncratic vocabulary and syntax. It is this straddling that again informs this emerging language of making – the tools of production also provide tools for development.

But it is important to note that not all software bows down to the Adobe interface gods. Processing, a software offering created by Benjamin Fry and Casey Reas, invites artists and designers to generate imagery through code. With its direct syntax and elegant interface, Processing enables users with minimal programming experience to create rule-based animations and interactive or self-evolving works. Interestingly, Processing.org's reference page resembles the table of contents of a post-Bauhaus design textbook, with master terms such as *Structure, Shape, Color, Image, Transform*, and *Typography* mixed in with geekier terms such as *Data, Control, Input, Math*, and *Constants*. Like Flash Action Script, Processing demands the designer to think in a new way, putting aside the timeline and the toolbar and learning to write instructions for generating marks on the screen and defining their behaviour. These software options and constraints point to the evolving nature of a language of making. If programming were to become a more common aspect of design training, it too might be integrated into the general and specific elements of theory and practice, conceiving and producing.

CONCLUSION

Software is a legitimate and necessary element of what we do in the classroom, even if our pedagogy never goes beyond production concerns. By considering the role of software from concept formation to finished product, instructors can make the process of teaching software intellectually rewarding to students and faculty alike. Additionally, the products themselves will benefit. Students will not simply see Illustrator as a playground without meaning, but as a commonplace of bounded exploration. *Software critique* involves looking at how interfaces both limit and enable our work, and how they connect concepts to real and virtual worlds. This means working with commercial applications (the industry standard) as well as exposing our students to code. Further, based on the fact that developers created software tools in ways that both mirror and inspire formal interests, we owe a debt of gratitude to the professionals who created these commercial applications. They are among the most significant design artifacts of our time.

Starting from an analytical approach to design thinking, within a software environment, each producer can animate design's core structures from his or her own point of view and from his or her own place in the world. After analysis comes more useful and interesting synthesis.

NOTES

1. Our visual research can be viewed at <http://www.gdbasics.com> (retrieved on November 1, 2007).
2. A product with a layers feature was also developed at this time by HSC, called Live Picture, but it did not compete successfully with Photoshop. The details can be found at: http://www.computerarts.co.uk/in_depth/features/the_history_of_photoshop (retrieved November 1, 2007).
3. Other fully developed examples can be found by going to <http://www.gdbasics.com> (retrieved November 1, 2007).

REFERENCES

- Arnheim, R. (1974). *Art and visual perception: A psychology of the creative eye* (Rev. ed.). Berkeley: University of California Press.
- Dondis, D. (1973). *Primer of visual literacy*. Cambridge, MA: MIT Press.
- Fish, S. (2005, May 31). *Editorial: Devoid of content*. New York Times.
- Hofmann, A. (1965). *Graphic design manual: Principles and practice*. New York: Van Nostrand Reinhold.

- Itten, J.** (1975). *Design and form: The basic course at the Bauhaus and later*. New York: Van Nostrand Reinhold.
- Johnson, S.** (2005). *Everything bad is good for you: How today's popular culture is actually making us smarter*. New York: Penguin.
- Kandinsky, W., Dearstyne, H., & Rebay, H.** (1947). *Point and line to plane: Contribution to the analysis of the pictorial elements*. New York: Solomon R. Guggenheim Foundation for the Museum of Non-Objective Painting.
- Kepes, G.** (1944). *Language of vision*. Chicago, IL: Paul Theobald.
- Klee, P.** (1953). *Pedagogical sketchbook*. New York: F. A. Praeger.
- Lupton, E.** (1996). *Mixing messages: Graphic design in contemporary culture*. New York: Cooper-Hewitt National Design Museum.
- Lupton, E.** (2007). *Graphic design: The new basics*. Retrieved November 10, 2007 from <http://www.gdbasics.com>
- Lupton, E., & Miller, J. A.** (1991). *The ABCs of circle, square, and triangle: The Bauhaus and design theory*. New York: Princeton Architectural Press.
- Lupton, E., & Miller, J. A.** (1999). *Design writing research*. London: Phaidon.
- Lupton, E., & Phillips, J. C.** (2008). *Graphic design: The new basics*. New York: Princeton Architectural Press.
- Manovich, L.** (2001). *The language of new media*. Cambridge, MA: MIT Press.
- Manovich, L.** (2002). *Generation flash*. Retrieved May 31, 2006, from <http://www.manovich.net>.
- Manovich, L.** (2005). *Remixability and modularity*. Retrieved May 31, 2006, from <http://www.manovich.net>
- Manovich, L.** (2006). *Import/export*. Retrieved May 31, 2006, from <http://www.manovich.net>.
- Manovich, L.** (2007). After effects, or velvet revolution. *Artifact*, 1(2), 67–75.
- Margolin, V.** (1998). *The struggle for utopia: Rodchenko, Lissitzky, Moholy-Nagy, 1917–1946*. Chicago, IL: University of Chicago Press.
- McCoy, K.** (1990). *Cranbrook: The new design discourse*. New York: Rizzoli.
- McCoy, K.** (1997). Countering the tradition of the apolitical designer. In M. Bierut, S. Heller, W. Drenttel, & D.K. Holland (Eds.), *Looking closer 2: Critical writings on graphic design* (pp. 212–218). New York: Allworth Press.
- McCoy, K.** (1999a). Hybridity happens. *Émigré*, 67, 38–47.
- McCoy, K.** (1999b). Maximize the message: Tailoring designs for your audience. *How*, 14(4), 106–113.
- McCoy, K.** (2002). When designers create culture. *Print*, 56(3), 181–183.
- Moholy-Nagy, L.** (1947). *Vision in motion*. Chicago, IL: Paul Theobald.
- Moholy-Nagy, S.** (1950). *Moholy-Nagy: Experiment in totality*. Cambridge, MA: MIT Press.
- Rothschild, D., Lupton, E., & Goldstein, D.** (1999). *Graphic design in the mechanical age: Selections from the Merrill C. Berman Collection*. New Haven, CT: Yale University Press.
- Rowe, C., & Slutzky, R.** (1993). Transparency: Literal and phenomenal (Part 2). In J. Ockman (Ed.), *Architecture culture 1943–1968, a documentary anthology* (pp. 205–225). New York: Rizzoli.
- Ruder, E.** (1981). *Typography*. New York: Hastings House.
- Rudnick Polstein, C., Pressman, E. R., & Zinnemann T.** (Producers). Frankenheimer, J. (Director) Cooper, K. (Title Designer). (1996). *The island of Dr. Moreau* [Motion picture]. United States: New Line Cinema, 1996.
- Spitz, R.** (2002). *The Ulm School of Design: A view behind the foreground*. Stuttgart, Germany: Edition.
- Story, D.** (2000). *From darkroom to desktop: How Photoshop came to light*. Retrieved May 28, 2006, from http://www.storyphoto.com/multimedia/multimedia_photoshop.html
- The history of Photoshop.** (2005). Retrieved May 28, 2006, from http://www.computerarts.co.uk/in_depth/features/the_history_of_photoshop
- Tupitsyn, M.** (1999). *El Lissitzky: Beyond the abstract cabinet*. New Haven, CT: Yale University Press.
- Weber, N. F.** (2004). *Josef&Anni Albers: Designs for living*. London: Merrell Publishers.
- Weingart, W.** (2000). *Wolfgang Weingart: My way to typography*. Switzerland: Lars Muller.
- Wick, R. K., & Grawe, G. D.** (2000). *Teaching at the Bauhaus*. Ostfildern, Germany: Hatje Cantz Publishers.

CORRESPONDENCE

Ellen Lupton,
Maryland Institute College of Art,
1300 Mt. Royal Avenue,
Baltimore, MD 21217, USA.
E-mail: elupton@designwritingresearch.org

ISSN 1749-3463 print/ ISSN 1749-3471

DOI: 10.1080/17493460701819488

© 2007 Artifact