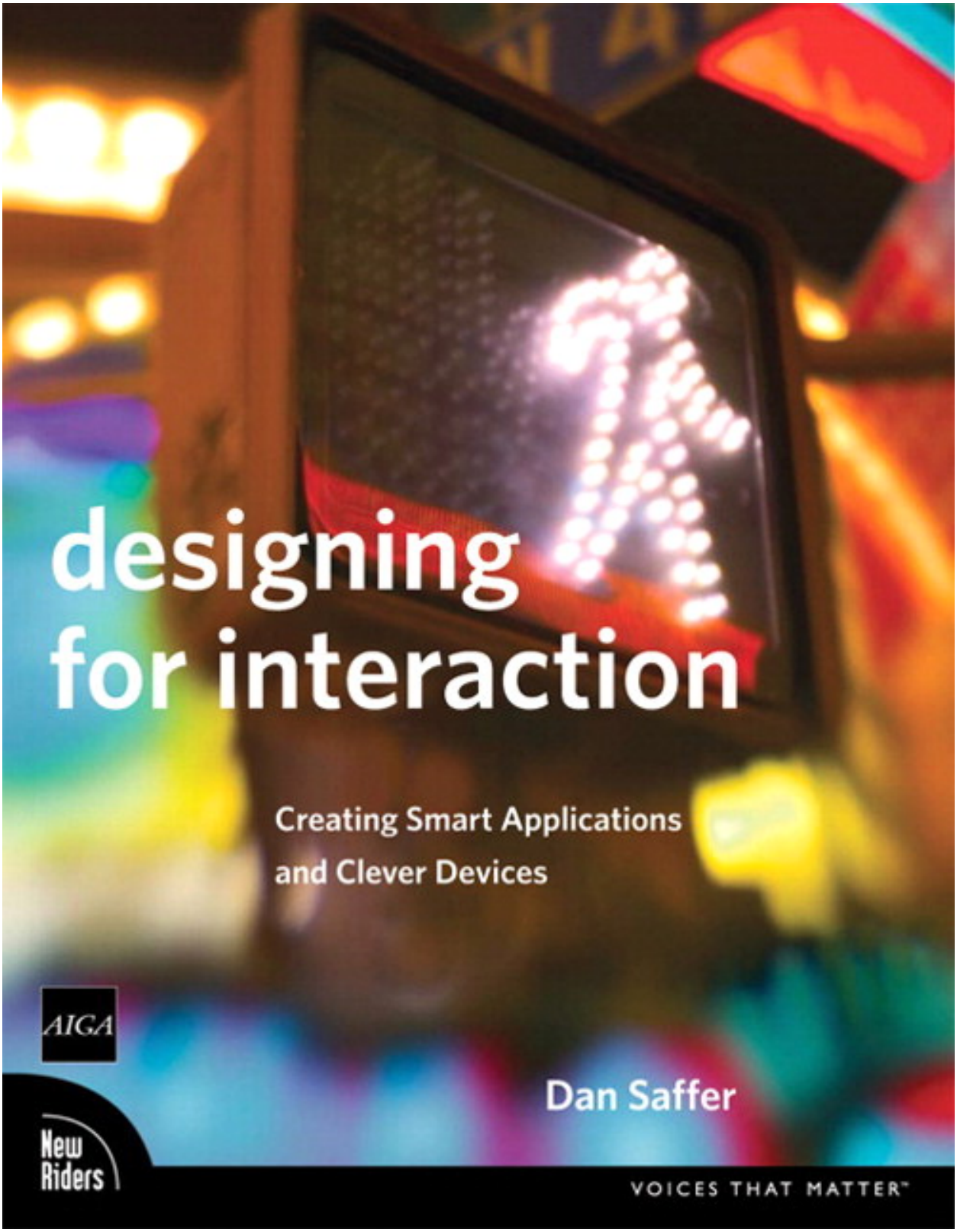
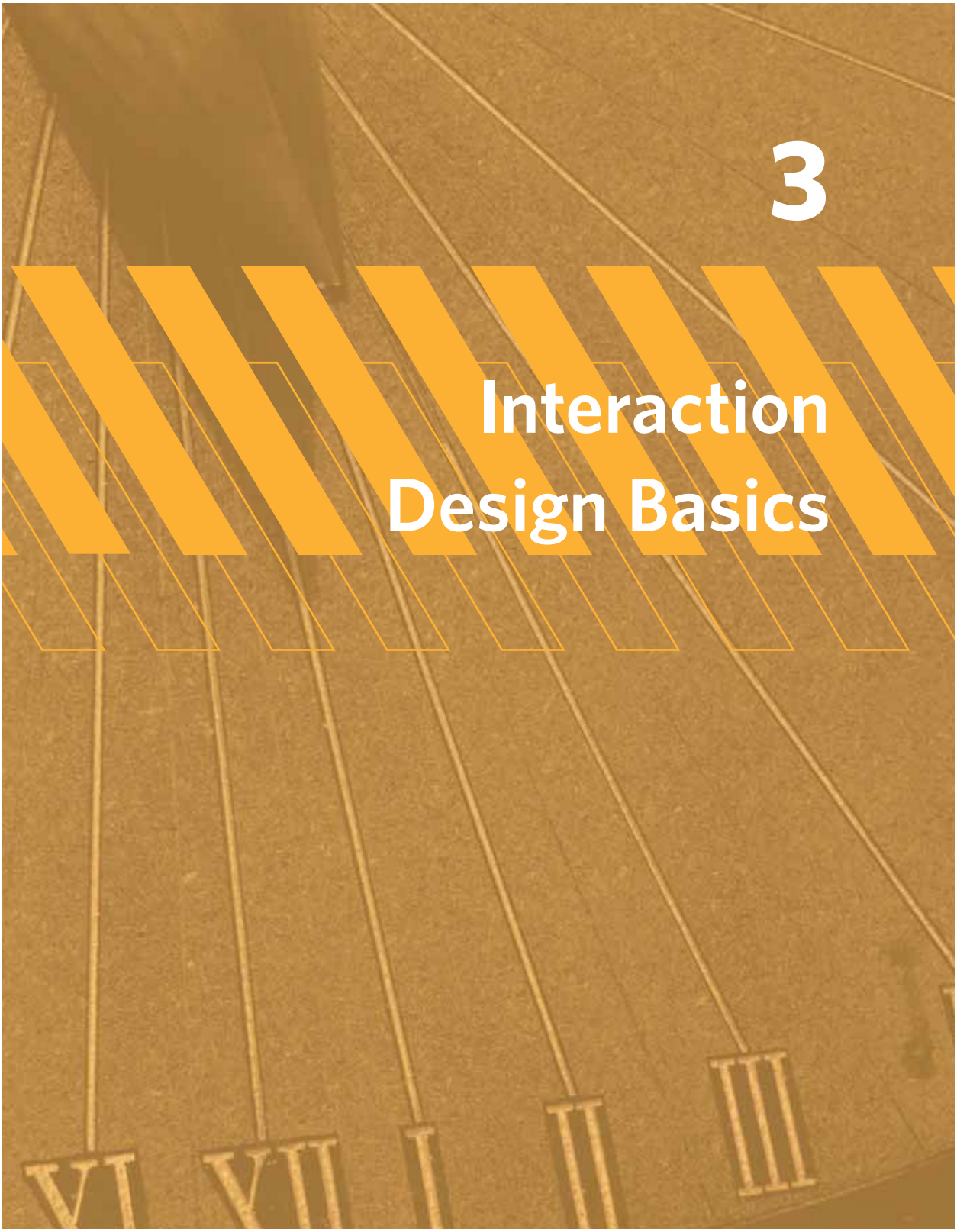

designing
for interaction
Creating Smart Applications
and Clever Devices
AIGA
Dan Saffer
New
Riders
VOICES THAT MATTER™

Table of Contents

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18



3

Interaction Design Basics

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

No matter what approach an interaction designer uses, the basic materials involved in solutions, such as motion, space, and time, remain the same. Likewise, the principles and ideas that guide and inform the design process also are the same: the "laws" such as Fitts' and Hick's, the Poka-Yoke Principle, feedback and feedforward, and direct and indirect manipulation.

We'll start with the building blocks of interaction design: the basic set of resources that interaction designers have to manipulate.

The Elements of Interaction Design

Other design disciplines use raw materials. Communication designers use basic visual elements such as the line. Industrial designers work with simple 3D shapes such as the cube, the sphere, and the cylinder. For interaction designers, who create products and services that can be digital (software) or analog (a karaoke machine) or both (a mobile phone), the design elements are more conceptual. And yet they offer a powerful set of components for interaction designers to bring to bear on their projects.

Motion

In much the same way that inert gases don't mingle with other gases, objects that don't move don't interact (**Figure 3.1**). An interaction, as noted in Chapter 1, is some sort of communication, and communication is about movement: our vocal cords vibrating as we speak, our hands and arms writing or typing as we send e-mail or instant messages, sound and data moving between two entities.

We communicate in many ways and through many different products, from mobile phones to e-mail. Those products and the people who use them generate *behavior*, and interaction designers are very concerned with behavior: the way that products behave in response to the way that people behave. And all behavior is, in fact, motion: motion colored by attitude, culture, personality, and context. There's wide variation even in such universal and seemingly simple behaviors such as walking (that's why, for instance, there's a need for both high-impact walking shoes and walkers for the elderly), and the designs we create have to understand and account for those variations. Even a simple motion like pressing a key on a keyboard can be difficult if you are elderly or infirm.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Figure 3.1

Motion. Without motion, there can be no interaction.

Motion is often a trigger for action, as when your finger clicks the button on your mouse. The triggered action (or at least the feedback for that action) is often about motion as well. You click a Web site link, and the page changes. You press a key, and an e-mail window closes. There is motion on your screen.

Without motion, there can be no interaction.

Space

Movement, even on a subatomic level, happens in some sort of space, even if the boundary of that space (as with, say, the Internet) is unclear. Interaction designers work in both 2D and 3D space, whether that space is a digital screen or the analog, physical space we all inhabit (**Figure 3.2**).

Most often, interaction design involves a combination of physical and digital spaces. You make a gesture in physical, analog space—for instance, turning a knob on your stereo—and you see the results on its digital display screen. The reverse can, of course, be true as well. You can stream music from your computer through your stereo and into physical space.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Figure 3.2

Space. All interactions take place in a space. This Italian piazza in Ferrara was designed for interaction.

Most interaction designers underutilize 3D space on screens. The physical flatness of our monitors and display screens causes us to ignore what the Renaissance painters discovered so long ago: perspective. Objects, even in a 2D space, can appear to move backward and forward in 3D space. Perspective creates, alongside X (height) and Y (width), a Z (depth) axis on which to work. Web sites are notably bad in their use of Z space.

Starbucks cafes typically make excellent use of physical space, with the ordering area separated from the fulfillment area where customers receive their beverages, and those areas separated from the area where people can customize (add milk and sugar and other condiments to) their drinks. Compare that to the typical crush around a single counter of a fast food restaurant.

Space provides a context for motion. Is the action taking place in a quiet office in front of a computer screen or in a crowded, noisy airport in front of a kiosk?

All interactions take place in a space.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Time

All interactions take place over time. Sometimes an interaction can occur instantaneously, as with a mouse click. Sometimes it can involve very long durations (**Figure 3.3**). You can still find online Usenet messages (Usenet is a sort of bulletin board system) from decades ago.

Movement through space takes time to accomplish. As every gamer will attest, it takes time to press buttons (around 8 milliseconds at the fastest). Even with broadband speeds, it takes time for packets of data to travel from distant servers through the physical wires and perhaps through the air via wireless signal to your computer.

Interaction designers need an awareness of time. Some tasks are complicated and take a long time to complete—for instance, searching for and buying a product. Many e-commerce Web sites require you to log in before making a purchase, and that login session will be active for a set time. Imagine if Amazon or other e-commerce sites timed out every few minutes and required you to log in repeatedly while shopping—it's unlikely you'd buy much from them. Some travel and concert-ticket Web sites make users race against the clock to enter their credit card information before their selected seats are lost.

Figure 3.3

Time. All interactions take place over time. Sometimes this time can be very brief, and sometimes it can be very long indeed.

Digital time is definitely not human time. Digital time is measured in milliseconds, a single one of which is considerably shorter than the blink of an eye. Changes made by the computer can be so nearly instantaneous that programmers need to program in delays so that humans can detect them.

You can feel the impact of milliseconds, however. Extra milliseconds added to every keystroke or mouse-click would probably make you think

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

your computer is slow because of the tiny delays. Several hundred milliseconds would cause frustration and anger, and a single-second delay each time you pressed a key would probably make your computer unusable.

Time creates rhythm. How fast a menu pops up on the screen or how long it takes to complete an action such as renewing your driver's license controls the rhythm of the interaction. Games are often alert to rhythm: how many aliens come at you at any given moment or how long it takes to complete a level. Rhythm is also an important component of animation: how quickly does a folder open or close on the desktop, how slowly does a drop-down menu slide open. Interaction designers control this rhythm.

Battery life (the duration of which is slowly getting better) is another element of time of which designers need to be cognizant. Some features, such as a backlight, drain more battery power than others and thus decrease the amount of time a device works. A mobile phone that worked for only 10 minutes when unplugged from a power outlet wouldn't be of much use.

Interactions happen over time.

Appearance

How something looks gives us cues as to how it behaves and how we should interact with it (**Figure 3.4**). The size, shape, and even weight of mobile devices let us know that they should be carried with us. The sleek black or silver look of digital video recorders like TiVo devices tells us that they are pieces of electronic equipment and belong alongside stereos and televisions.

Appearance is one major source (texture is the other) of what cognitive psychologist James Gibson, in 1966, called *affordances*. Gibson explored the concept more fully in his 1979 book *The Ecological Approach to Visual Perception,* but it wasn't until Don Norman's seminal book *The Psychology of Everyday Things*, in 1988, that the term spread into design. An affordance is a property, or multiple properties, of an object that provides some indication of how to interact with that object or with a feature on that object. A chair has an affordance of sitting because of its shape. A button has an affordance of pushing because of its shape and the way it moves (or seemingly moves). The empty space in a cup is an affordance that tells us we could fill the cup with liquid.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Affordances (or, technically, *perceived* affordances) are contextual and cultural. You know you can push a button because you've pushed one before. On the other hand, a person who has never seen chopsticks may be puzzled about what to do with them.

Figure 3.4

Appearance. The design of this gate latch provides affordances indicating how it should be used.

Except to the visually impaired (for whom texture often substitutes), appearance also conveys emotional content. Is this product whimsical or serious? Practical or playful? Appearance can also convey other attributes that may be meaningful: Is the object expensive or cheap? Complicated or simple? Daunting or approachable? Single use or enduring? Structured or casual?

Appearance has many variables for designers to alter:

- Proportion
- Structure
- Size
- Shape
- Weight
- Color (hue, value, saturation)

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

All of these characteristics (and more) add up to appearance, and nearly every design has some sort of appearance, even if that appearance is a simple command line.

Texture

While texture can also be part of the appearance, how an object *feels* in the hand can convey the same sort of information as appearance (**Figure 3.5**). Texture, too, can convey affordances. The sensation of an object can provide clues as to how it is to be used as well as when and where. Is it solid or flimsy? Is it fragile or durable? Do the knobs spin or push or do both?

Figure 3.5
Texture. How something feels in the hand can also provide affordances indicating how it could be used, as in the case of these cane lobster traps.

Texture can convey emotion as well. A fuzzy, plush object conveys a different meaning than a hard, metallic one.

Designers can also work with texture variables such as vibration and heat to signify actions. A mobile phone can vibrate when a new message arrives, and one could imagine it growing colder the longer it's been since a voice-mail message arrived.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Sound

Sound (**Figure 3.6**) is a small part of most interaction designs, but it can be an important part, especially for alerts and ambient devices (see Chapter 7). Sound possesses many variables that can convey information as well. You wouldn't want a loud screech to come out of your computer every time you received e-mail, and a soft whisper wouldn't cause traffic to move aside for an ambulance.

Figure 3.6

Sound. Sound can be manipulated as a tool of interaction design through devices such as this mixing board.

Sounds are made up of three main components, all of which can be adjusted by a designer:

- **Pitch.** How high in range a sound is. Is it high pitched like a bird's song or deep like thunder?
- **Volume.** How loud a sound is.
- **Timbre or tone quality.** What type of sound it is. Sounds played at the same volume and pitch can seem very different. Think of a middle C played on a trumpet and one played on a piano.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Sound is underutilized (some would say rightfully so) in interaction design, but even a little bit of sound can make a major difference in a product. Steve Jobs insisted that the iPod's wheel make an audible click that could be heard without headphones.

All of these elements of interaction design comprise any interaction designer's toolkit, and while interaction designers may not consciously manipulate them, they are the building blocks of interaction design. Now let's look at some principles that should guide the way that interaction designers assemble these elements into a product or service.

The Laws of Interaction Design

Interaction design, being a new field, doesn't have very many hard and fast rules, or "laws," to speak of. In a sense, interaction designers are still figuring out many of the basic principles of the work they do. However, there are a handful of laws that interaction designers have used successfully. Except for Moore's Law, which designers need only understand, not put into practice, these laws should guide the work, not dictate it.

Moore's Law

In 1965, Gordon Moore, a co-founder of microchip maker Intel, predicted that every two years, the number of transistors on integrated circuits (a rough measure of computer processing power) will double.

Amazingly, this is exactly what has occurred and is still occurring, with staggering results. There is more processing power in a modern laptop than in all of NASA's Mission Control Center when the space agency sent a man to the moon in 1969. The fulfillment of Moore's Law is the underpinning for everything from color monitors to video teleconferencing to the ability to run multiple programs at once. Designers can conceive of devices that are faster, smaller, and more powerful than could feasibly have been considered even a decade ago, much less in 1965 when Moore made his prediction. And in two more years, our devices will be faster, smaller, and more powerful still.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Fitts' Law

Published in 1954 by psychologist Paul Fitts, Fitts' (pronounced "fitzez") Law simply states that the time it takes to move from a starting position to a final target is determined by two factors: the distance to the target and the size of the target. Fitts' Law models the act of pointing, both with a finger and with a device like a mouse. The larger the target, the faster it can be pointed to. Likewise, the closer the target, the faster it can be pointed to.

Fitts' Law has three main implications for interaction designers. Since the size of the target matters, clickable objects like buttons need to be reasonable sizes. As anyone who has tried to click a tiny icon will attest, the smaller the object, the harder it is to manipulate. Second, the edges and corners of screens are excellent places to position things like menu bars and buttons. Edges and corners are huge targets because they basically have infinite height or width. You can't overshoot them with the mouse; your mouse will stop on the edge of the screen no matter how far you move it, and thus will land on top of the button or menu. The third major implication of Fitts' Law is that pop-up menus that appear next to the object that a person is working on (such as a menu that appears next to an object when the user right-clicks the mouse) can usually be opened more quickly than can pull-down menus at the top of the screen, which require travel to other parts of the screen.

Hick's Law

Hick's Law, or the Hick-Hyman Law, says that the time it takes for users to make decisions is determined by the number of possible choices they have. People don't consider a group of possible choices one by one. Instead, they subdivide the choices into categories, eliminating about half of the remaining choices with each step in the decision. Thus, Hick's Law claims that a user will more quickly make choices from one menu of 10 items than from two menus of 5 items each.

A controversial implication of this law is that it is better for products to give users many choices simultaneously instead of organizing the choices into hierarchical groups, as in drop-down menus. If followed to an extreme, this approach could create some truly frightening designs. Imagine if a content-rich site like Yahoo or Amazon presented all of its links on the home page, or if your mobile phone displayed all of its features on its main screen.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Hick's Law also states that the time it takes to make a decision is affected by two factors: familiarity with the choices, such as from repeated use, and the format of the choices—are they sounds or words, videos or buttons?

The Magical Number Seven

Hick's Law seems to run counter to George Miller's Magical Number Seven rule. In 1956, Miller, a Princeton University psychology professor, determined that the human mind is best able to remember information in chunks of seven items, "plus or minus two." After five to nine pieces of information (for instance, navigation labels or a list of features or a set of numbers), the human mind starts making errors. It seems that we have difficulty keeping more than that amount of information in our short-term memory at any given time.

Some designers have taken the Magical Number Seven rule to an extreme, making sure that there are never any more than seven items on a screen at any given time. This is a bit excessive, because Miller was specifically talking about bits of information that humans have to remember or visualize in short-term memory. When those bits of information are displayed on a screen, users don't have to keep them in their short-term memory; they can always refer to them.

But designers should take care not to design a product that causes cognitive overload by ignoring the Magical Number Seven rule. For example, designers should never create a device that forces users to remember unfamiliar items across screens or pages. Imagine if you had to type a new phone number on three separate screens of your mobile phone. You'd scramble to do so (if you could!) before the number faded from your short-term memory.

Tesler's Law of the Conservation of Complexity

Larry Tesler, one of the pioneers of interaction design (see the interview with him later in this chapter), coined Tesler's Law of the Conservation of Complexity, which states that some complexity is inherent to every process. There is a point beyond which you can't simplify a process any further; you can only move the inherent complexity from one place to another.

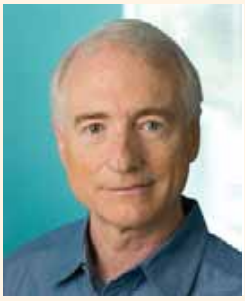
For example, for an e-mail message, two elements are required: your e-mail address and the address of the person to whom you are sending the mail. If

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

either of these items is missing, the e-mail can't be sent, and your e-mail client will tell you so. It's a necessary complexity. But some of that burden has likely been shifted to your e-mail client. You don't typically have to enter your e-mail address every time you send e-mail; the e-mail program handles that task for you. Likewise, the e-mail client probably also helps you by remembering e-mail addresses to which you've sent mail in the past, so that you don't have to remember them and type them in fully each time. The complexity isn't gone, though—instead, some of it has been shifted to the software.

Interaction designers need to be aware of Tesler's Law for two reasons. First, designers need to acknowledge that all processes have elements that cannot be made simpler, no matter how much we tinker with them. As The Who told us in their 1966 song "Substitute," the simple things you see are all complicated. Second, designers need to look for reasonable places to move this complexity into the products they make. It doesn't make sense for users to type their e-mail addresses in every e-mail they send when software can handle this task. The burden of complexity needs to be shared as much as possible by the products interaction designers make.

Larry Tesler on the Laws of Interaction Design



Larry Tesler's resume reads like the history of interaction design. He's worked at Xerox PARC, Apple, and Amazon and is now at Yahoo as vice president of the User Experience and Design group. While at Xerox PARC, he helped develop some of the language of interaction design, including pop-up menus and cut-and-paste editing. His law of the Conservation of Complexity (discussed in this chapter) is known to programmers and designers alike.

You've worked at some of the seminal places for interaction design: Xerox PARC, Apple, Amazon, and now Yahoo. What do they all have in common?

All of them place a high value on both advanced technology and customer delight.

Are there any unbreakable "laws" in interaction design?

Just one. Design for the users.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Larry Tesler on the Laws of Interaction Design *Continued*

How did you come up with Tesler's Law of the Conservation of Complexity?

In the early days of our field, when I worked at Xerox PARC, the idea of user interface consistency was new and controversial. Many of us realized that consistency would benefit not only users, but also developers, because standards could be encapsulated in shared software libraries. We made an economic argument: If we establish standards and encourage consistency, we can reduce time to market and code size.

In 1983–1985, when I was developing the Mac app object-oriented framework at Apple, I advocated a three-layer code model. In addition to the Macintosh Toolbox—a shared software library—and the application itself, I made the case for an intermediate layer that implemented what I called a generic application. A generic application was a real interactive program—with windows, menus, and commands—that did nothing at all, but did it in a standard way. You could create, open, save, and print documents, but the documents lacked form and were empty of content. You built your actual application by modifying the generic application in an object-oriented way.

To sell the idea to Apple management and independent software vendors, I came up with the Law of Conservation of Complexity. I postulated that every application must have an inherent amount of irreducible complexity. The only question is who will have to deal with it.

Because computers back then were small, slow, and expensive, programs were designed to be compact, not easy to use. The user had to deal with complexity because the programmer couldn't. But commercial software is written once and used millions of times. If a million users each waste a minute a day dealing with complexity that an engineer could have eliminated in a week by making the software a little more complex, you are penalizing the user to make the engineer's job easier.

Whose time is more important to the success of your business? For mass-market software, unless you have a sustainable monopoly position, the customer's time has to be more important to you than your own.

What personal qualities do you think make a good interaction designer?

Enough confidence to believe you can solve any design problem and enough humility to understand that most of your initial ideas are probably bad. Enough humility to listen to ideas from other people that may be better than your own and enough confidence to understand that going with other people's ideas does not diminish your value as a designer.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Larry Tesler on the Laws of Interaction Design *Continued*

True concern for the comfort and happiness of other people, including your users and your teammates. If you're not teammate friendly, your products won't be user friendly. That does not mean you should cave in under pressure on an important issue when you have data that supports your opinion. But it does mean you should judge success by the success of the product and the team, not just by the success of your own narrow contribution.

There are a lot of other desirable personal qualities for a designer, such as attention to detail, objectivity, appreciation of humor, appreciation of esthetics, and appreciation of data about users and usage.

The Poka-Yoke Principle

Legendary Japanese industrial engineer and quality guru Shigeo Shingo created the Poka-Yoke Principle in 1961 while working for Toyota. Poka-yoke roughly translates in English to mistake proofing: avoiding (*yokeru*) inadvertent errors (*poka*). Designers use poka-yoke when they put constraints on products to prevent errors, forcing users to adjust their behavior and correctly execute an operation.

Simple examples of the application of poka-yoke are the cords (USB, FireWire, power, and others) that fit into electronic devices only in a particular way and in a particular place, and thus prevent someone from, say, plugging the power cord into the hole where the headphones go (**Figure 3.7**). In this way, poka-yoke ensures that proper conditions exist *before* a process begins, preventing problems from occurring in the first place. Poka-yoke can be implemented in lots of forms: by signs (Do not touch the third rail!), procedures (Step 1: Unplug toaster), humans (police directing traffic around an accident), or any other entity that prevents incorrect execution of a process step. Where prevention is not possible, poka-yoke mandates that problems be stopped as early as possible in the process. Interaction designers should look for opportunities to use the Poka-Yoke Principle.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Figure 3.7

An illustration of the Poka-Yoke Principle. The USB cord will fit into only a particular slot on this laptop computer.

Direct and Indirect Manipulation

Digital objects can be manipulated in two ways: directly and indirectly. Although technically digital objects can be manipulated only indirectly (you can't touch something that's made of bits and bytes, after all), direct and indirect manipulation represent two ways of thinking about how to work with digital objects.

Direct manipulation is a term coined by University of Maryland professor Ben Shneiderman in the early 1980s. It refers to the process in which, by selecting a digital object with a finger or with a mouse or some other extension of the hand (which is really direct manipulation in and of itself), we can then do something to the object: move it, turn it, drag it to the trash, change its color, and so on. We can mimic an action that we might perform on a similar object in the physical world. For example, we can scale an object by dragging a corner of it as though we were stretching it. Direct manipulation, because it more closely maps to our physical experiences, is supposedly more easily learned and used, especially for manipulating 3D objects in digital space.

In indirect manipulation, we use a command or menu or other means that isn't directly a part of the digital object to alter that object. Choosing the Select All command in the application interface and pressing the Delete key

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

on the keyboard are examples of indirect manipulation. In the past, especially during the years before the Macintosh popularized the GUI, nearly all computer commands were indirect.

Interaction designers need to decide how digital objects in their products can be manipulated: directly, indirectly, or (more and more frequently) in both ways.

Feedback and Feedforward

Feedback, as it is commonly used, is some indication that something has happened. Feedback should occur like crooked voting does: early and often. Every action by a person who engages with the product or service, no matter how slightly, should be accompanied by some acknowledgment of the action: Moving the mouse should move the cursor. Pressing a key on your mobile phone should display a number.

Proceeding otherwise is to court errors, some of them potentially serious. Frequently, if there is no immediate or obvious feedback, users will repeat the action they just did—for instance, pushing a button twice. Needless to say, this can cause problems, such as accidentally buying an item twice or transferring money multiple times. If the button is connected to dangerous machinery, it could result in injury or death. People need feedback.

We know that feedback is essential; designing the *appropriate* feedback is the designer's task. The designer has to determine how quickly the product or service will respond and in what manner. Should the response be something simple such as the appearance of a letter on a screen (the feedback in word processing for pressing a key), or should it be a complex indicator such as a pattern of blinking LED lights on a device that monitors your stock portfolio?

Related to feedback (and also to affordances) is what designer Tom Djajadiningrat calls *feedforward*: knowing what will happen *before* you perform an action. Feedforward can be a straightforward message ("Pushing this button will submit your order") or simple cues such as hypertext links with descriptive names instead of "Here."

Feedforward allows users to perform an action with confidence because it gives them an idea of what will happen next. Feedforward is harder to design into products and services than feedback, but designers should keep an eye out for opportunities to use it.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Characteristics of Good Interaction Design

In almost all designs, no matter for what, digital or analog, interaction designers should strive to reflect the following set of adjectives. Certainly in some circumstances, some of these characteristics may not be appropriate—you may not want a mission control panel to be playful, for example—but in general, if a product or service can be described with these adjectives, it is likely an example of good interaction design.

Trustworthy

Before we'll use a tool, we have to trust that it can do the job. You wouldn't pick up a rusty hammer with a loose handle to drive in nails—you wouldn't trust it to not smash your thumb. The same holds true for digital products and for services. Given a choice, you wouldn't eat in a filthy restaurant or use a mobile phone that only occasionally rang when someone called.

Humans likely make decisions about the trustworthiness of a product or service within seconds of engaging with it. Indeed, recent research has suggested that humans make snap judgments in less than a second. Products and services have to display their trustworthiness quickly. They need to appear like they aren't going to rip us off, injure us, sell our personal data, break immediately, or otherwise betray our trust in them.

If we trust something, we are much more likely to deeply engage with it and perhaps even take measured risks with it. Would you rather walk across a sturdy steel bridge or a rotting, swaying rope one? A trustworthy product or service is one that users will take the time to examine and learn, discovering and using more features because they aren't afraid that something bad will happen to them if they do.

Appropriate

The solutions that interaction designers come up with need to be appropriate to the culture, situation, and context that they live in. All of these factors can drastically affect the type of product or service that a designer makes (**Figure 3.8**).

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Figure 3.8

The henna decoration of this girl's hand could provide valuable insights for designers of products for India.

The Japanese and Koreans are famously advanced mobile phone users and can use expert features that would boggle Americans. In France, cooking and eating are social events best enjoyed slowly, and even simple meals are prepared with care; American-style fast food services have only grudgingly gained acceptance there after many years of struggle. Certain colors have different meanings in different cultures—for instance, white in China and Japan denotes mourning, whereas in Europe and America it denotes purity and cleanliness.

Dutch cultural anthropologist Geert Hofstede, in his book *Software of the Mind*, identified five major dimensions that can be used to help understand different cultures and adjust designs accordingly:

- **Power distance.** To what extent will members of a culture accept inequities in power among members of that culture?
- **Individualism versus collectivism.** Do members of a culture have loose ties to their families and others, or are they members of strong groups (especially families)?
- **Masculinity versus femininity.** How strong are the gender roles in a culture? In strong masculine cultures, traditional distinctions between the sexes are maintained.

- **Uncertainty avoidance.** How tolerant is a culture of ambiguity and uncertainty?
- **Long-term versus short-term orientation.** How much does a culture value the future over the past and present?

Where a culture places on these continuums can greatly affect design. For example, a culture that places a high value on uncertainty avoidance may prefer simple, limited choices in a product rather than many, complex ones.

An understanding of the specific situation that a product or service will work in, and also the emotional context of that situation, are essential to good design. Airport check-in kiosks (**Figure 3.9**) can't be complicated or require any significant learning to use them—they are used infrequently by harried people. Likewise, you wouldn't design a toy for young children the same way you would design a military communication device to be used in battlefield conditions. The situation dictates a set of constraints—technical, cultural, and emotional—that designers ignore at their peril.

Figure 3.9

An airport check-in kiosk like this one cannot be too complicated. It needs to take into account its context (a busy place) and the user's state of mind at the time of use (probably harried).

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Smart

The products and services we use need to be smarter than we are. They have to prevent us from making mistakes or from working harder than we need to. They need to do for us humans the things we have trouble doing—rapidly perform computations, infallibly remember things over both the long and short term, and detect complicated patterns. They need to do the work we can't easily do alone. They have to be smart.

As noted earlier, Tesler's Law tells us that all processes have a core of complexity that cannot be overcome, only moved to different places. While taking care to not remove important skills from people, a good design moves as much complexity as possible into the services we use and the products we own, where it can be handled more effectively. Your mobile phone can hold far more phone numbers than you could ever memorize. Calling 911 will get the proper emergency services to your house more quickly than will looking up and then dialing the individual phone numbers. A Google search (**Figure 3.10**) will find obscure pieces of data more efficiently than will manually browsing because the Web service has taken on the complexity of browsing millions of pages.

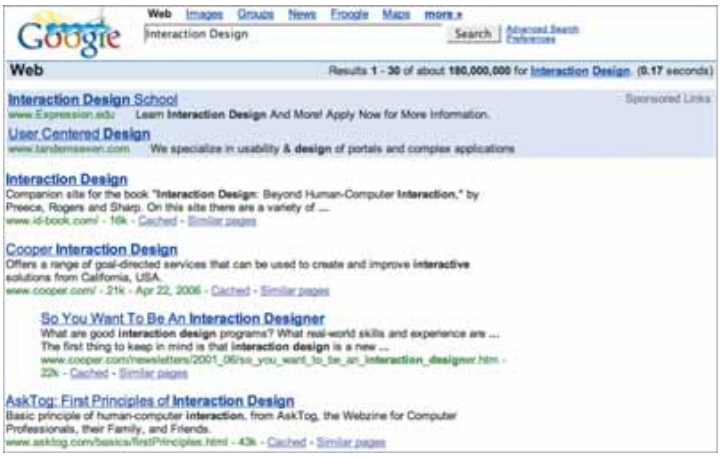


Figure 3.10
Google search results do what we cannot do ourselves: search millions of Web sites for specific words.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Responsive

As my wife will attest, there is little more annoying than talking to someone who doesn't respond. The same holds true for "conversations" with products and services. We need to know that the product "heard" what we told it to do and that it is working on that task. We also want to know what the product or service is doing. A spinning circle or a tiny hourglass icon doesn't give users much transparency into what is happening behind the scenes.

If a response to an action is going to take significant time (more than 1 second, which, believe it or not, can seem like a long wait), a good design provides some mechanism that lets the user know that the system has heard the request and is doing something (**Figure 3.11**). This doesn't make the waiting time less, but it makes it seem less. Excellent examples of such a mechanism are the indicators that tell you how long a process such as a software installation will take. These indicators also assure the user that the process hasn't gone into an endless cycle from which there is no return.

Figure 3.11

The Orbitz searching screen won't make the wait less for search results, but it will make it seem less because of its responsiveness.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

The responsiveness of digital products can be characterized by these four basic levels, determined by the time between an action and the product's response:

- **Immediate.** When a product or service responds in 0.1 second or less, the user considers the response immediate and continues the task with no perceived interruption. When you push a key on your keyboard and a letter instantly appears, that is an immediate response.
- **Stammer.** If a product or service takes 0.1 second to 1 second to respond, users will notice a delay. If such a delay is not frequently repeated, users will probably overlook it. Repeated, it will make the product or service feel sluggish. For instance, if you press a key on your keyboard and it takes a second for the letter to appear on your screen, you'll notice the delay but likely will keep typing. If this happens with every key press, you will quickly become frustrated with your word processor.
- **Interruption.** After a second of no response, users will feel that the task they were doing was interrupted, and their focus will shift from the task at hand to the product or service itself. If you click a Submit button to execute a stock trade and nothing happens for several seconds, you will worry about the trade and wonder if the Web site is broken. Multiple interruptions can lead to a disruption.
- **Disruption.** If a delay of more than 10 seconds occurs, users will consider the task at hand completely disrupted. Feedback such as a progress bar or a timer that indicates how long a process will take will allay users' concerns and also allow the user to decide whether to continue the process. Marquees in the London Underground indicating when the next trains will arrive are excellent examples of responsiveness that addresses this level of delay.

Responsiveness makes people feel appreciated and understood.

Clever

Clever implies intelligence without smugness or condescension. It suggests humor and slyness without being obnoxious. And it also implies delight. Using a clever product or service, especially for the first time, leads to

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Figure 3.12

TiVo's Instant Replay (or eight-second rewind) button is clever in that it anticipates users' needs (to hear or see the last bit of a show again).

moments of delight when you discover how clever, how thoughtful, it is. And delight is one of the most sublime emotions that one can experience, leading to long-lasting good feelings.

Clever products and services predict the needs of their users and then fulfill those needs in unexpectedly pleasing ways. TiVo is a particularly clever service. Its designers anticipated how users would enjoy the service and designed it with cleverness in mind. An excellent example of this sort of design is its eight-second rewind button. Often while watching TV, viewers will miss a line of dialogue ("What did he just say?"). Rewinding eight seconds is typically enough to hear the line again (**Figure 3.12**). The first time viewers use the clever eight-second rewind button, they typically are hooked on the feature.

Ludic

As children, we learn how to be adults through play. We play house. We role play adult authority figures. We learn teamwork through sports and group activities. Through play, we explore and experiment with life (**Figure 3.13**).

Figure 3.13

Two little girls playing doctor. Through play, we explore the unknown.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18

Ludic (pronounced "loo-dik") means playful. Designing products or services that allow play doesn't mean designing everything to be a toy or game, but rather providing the environment and means for users to play with a product or service. Through serious play, we seek out new products, services, and features and then try them to see how they work. How many times have you pushed a button just to see what it did?

It is hard to play when you are uncomfortable, so users need to be made to feel relaxed enough so they can engage in play. A well-designed product or service makes making errors difficult instead of simply providing lots of warning messages, which make people nervous and uncomfortable.

Play also requires a lack of serious consequences. While in a larger sense this feature may be difficult to design into medical devices, banking and emergency systems, and military applications, for instance, the ability to undo mistakes is crucial to fostering an environment for play. If the user feels trapped or powerless, the play is effectively over.

Pleasurable

Unless a product or service is pleasing to use, we likely won't use it often unless we have to (say, for our work or well being). In design terms, products and services can be pleasing in two ways: aesthetically and functionally (**Figure 3.14**).

As cognitive psychologist Don Norman pointed out in his book *Emotional Design*, beautiful products work better. Humans are more forgiving of mistakes in things that are more aesthetically pleasing. Objects that are agreeable to the eye or that fit well in the hand engender good feelings in their users, as do services in pleasing environments. The experience of eating at a fine restaurant is as often about the space and the other patrons as it is about the food.

Figure 3.14

A sauna is an enjoyable experience, not only because of the heat, but also because of the functionally and aesthetically pleasing wood that typically panels the room. If the room were tiled, the experience would be very different.

But beauty isn't everything. For a product or service to be appealing, it has to work well. Have you been to a beautiful Web site where you can't find anything? Products and services need to be functionally pleasing as well. We need to feel that time spent using the product or service is time well spent, that it is effective in accomplishing what we want it to do.

Summary

All of these characteristics, along with the laws and principles of interaction design, guide the products and services that interaction designers create. But we should never forget who these products and services are for: the users. So in the next chapter, we'll discuss how to talk to them, observe them, and make things with them.

Designing for Interaction: Creating Smart Applications and Clever Devices By Dan Saffer
ISBN: 0321447123 Publisher: Peachpit Press
Print Publication Date: 2006/07/18