Part 3

Four strategies for simplicity

The four strategies

Over the years I've seen many ingenious solutions to the problem of simplifying a DVD remote, but I've found that they fall into four categories:

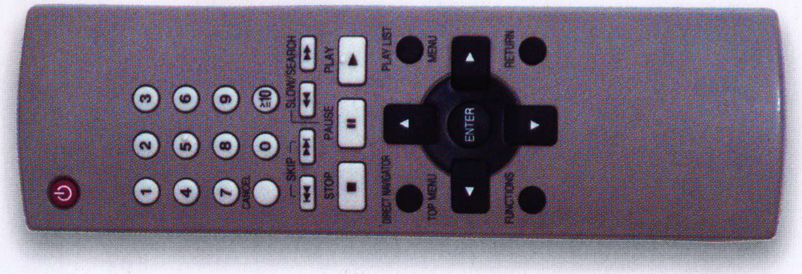
- Remove—get rid of all the unnecessary buttons until the device is stripped back to its essentials.
- Organize—arrange the buttons into groups that make more sense.
- Hide—hide all but the most important buttons behind a hatch so they don't distract users.
- Displace—create a very simple remote control with a few basic features and control the rest via a menu on the TV screen, displacing the complexity from the remote control to the TV.

Some people do a little of each, but usually they pick a primary strategy. Some use additional technology like touch-screen displays on the remote control or the ability to wave at the TV, but these are just forms of removing, organizing, hiding, or displacing.

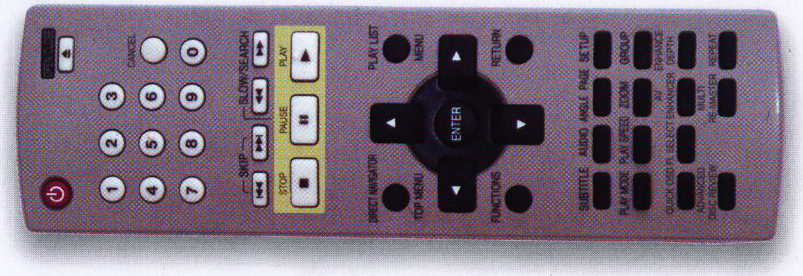
As I've tried to simplify other devices and experiences, I've found that the same four strategies keep cropping up in one form or another. The strategies apply to both functionality and content. And the strategies apply whether you're looking at something large, like an entire website, or something small, like a single page.

Each of the strategies has its strengths and weaknesses, which I'll discuss in the following chapters. A big part of success comes in choosing the right strategy for the problem at hand.

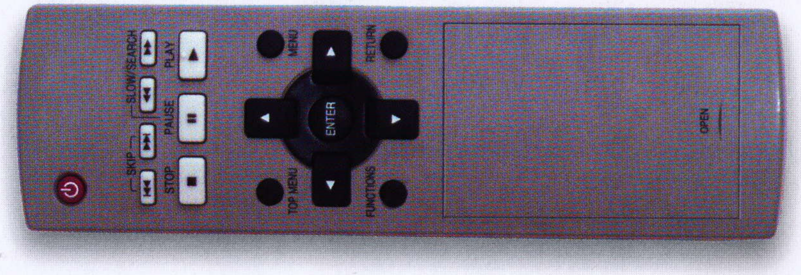
Remove



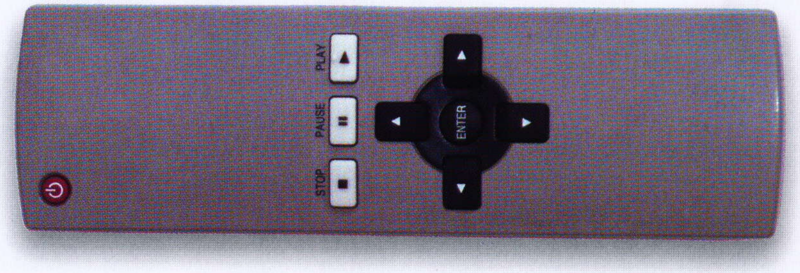
Organize



Hide



Displace



Part 4

Remove

Remove

According to a 2002 study by Standish Group, 64 percent of software features are "never or rarely used." Take a look at your DVD remote control and count the number of buttons that you've never touched. The same goes for almost any gadget or software you care to name. There are plenty of opportunities to simplify by removing.

Removing or omitting features can lead to successful products:

- Tumblr's blog service has a fraction of the functionality of sites like WordPress or Blogger, but three years after its launch, it was booming with over two million blog posts every day.
- The Lotus Elise started life as a back-to-basics sports car with no air-conditioning and a production run of eight hundred. Fifteen years later, it is still in production and tens of thousands of them have been sold.
- At launch, the iPhone had fewer features than competing phones from Nokia and RIM (makers of BlackBerry), but it was an instant hit.
- Basecamp, a project management extranet by 37signals, does a fraction of what extranet software like Microsoft SharePoint does, but *BusinessWeek* described it as "addictively easy-to-use" and it is used by millions of people worldwide.

Conventional wisdom says that more features mean more capability which, in turn, means a more useful product. But these examples choose depth of capability rather than breadth. They're useful because they do a few things far better than their rivals.

Conventional wisdom also says that products with more features will beat products with fewer. But all of these examples have competed against more fully featured rivals and won.

Removing clutter allowed designers to focus on solving a few important problems really well. It also allowed users to focus on meeting their goals without distraction.

It's often easy to understand what's essential: a DVD remote needs a play button and a stop button. The problem comes with things that might be valuable. So, when you're simplifying by removing, begin with a blank sheet of paper and ask, "What are the important problems?" Then gradually add the features and content that matter most.

The most obvious way to simplify is to remove what's unnecessary.

Focus

The "remove" strategy is about removing distractions to bring focus to your project:

- Focus on what's valuable to users. This means concentrating on features that deliver the users' core experience. It also means delivering features that eliminate users' frustrations and ease their sense of anxiety.
- Focus your resources on delivering value by removing lame features, irrelevant extras, and bribes.
- Focus on meeting users' goals. Agonizing over the process will get you bogged down in detail.
- Remove the distractions of tiny speed bumps that add to the load on the user: error messages, irrelevant text, unnecessary choices, and visual clutter.

With patience and the data to back you up, you can bring focus to most projects. If your problem is political, you can overcome it by building on small successes or by using evidence from tests. If your problem is out-of-date technology or incompatible systems, these too can change (slowly) over time. However, there are a couple of exceptions.

Sometimes there is an unavoidable legal requirement to include particular wording or information. Financial services and medical regulations often require that specific wording is used, not because it makes sense to the public, but because it makes sense to lawmakers. Laws can be changed, too. David Sless in Australia has had some success in getting lawmakers to focus on whether consumers understand labels, rather than requiring long and confusing instructions.

Sometimes you can't remove because your design is part of a larger system. That's the case with the DVD remote. For instance, there are millions of DVDs in circulation that make use of the numeric keypad on the remote. If you removed it, you would risk breaking the user experience for anyone who already owned such a DVD.

While you're waiting for the world to change, however, there are other ways of simplifying that are less radical, but quicker to implement.

Removing clutter helps users focus on what's important.

Part 5

Organize

Organize

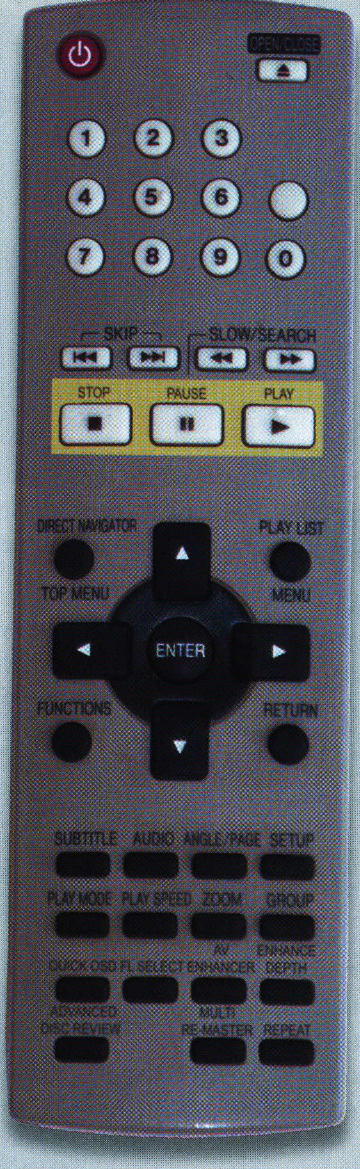
Organizing is a great strategy for simplifying. In the case of the DVD remote control, it's probably the solution I've seen most often. It's usually an inexpensive solution—changing the layout and labeling the buttons on the DVD remote control costs less and demands fewer tough decisions than, say, removing.

There are plenty of options open to you in organizing an interface—size, color, position, shape, hierarchy. But those choices need to be employed with restraint. Some of the DVD remote controls I've seen over the years have had so many colored buttons they look as though they're made from Skittles.

If you want to organize for simplicity, it's important to emphasize just one or two important things. Simple organization doesn't draw attention to itself, it lets users focus on what they're doing.

The best DVD remote control designs emphasize the starting point (the on/off switch) and the most frequently used buttons (play, pause, and stop).

The Flip is also an excellent example of this. Of its nine buttons, only one (record) is strongly emphasized. If design is like a conversation, then openings are always the most difficult part. The Flip knows just how to say, "Hello, let's start here."



Organizing is often the quickest way to make things simpler.

Chunking

One way to make the blocks of buttons on the DVD remote control more manageable is to break them into chunks.

Chunking is used throughout user interface design. Microsoft Word has hundreds of features. To make them manageable, they are divided up into around nine menus. Each of those has a couple dozen commands—still too many to take in at a glance, so they're divided into chunks again. Click on a menu item and it'll often take you into a dialog box where even more features are available. The daunting list of features is grouped into manageable chunks within a hierarchy.

The classic advice here is to break items down into groups of "seven plus or minus two." In theory, this corresponds to the number of items your brain can hold in short-term memory. If you read a list of ten items, you'll likely have forgotten one of them by the time you get to the end.

Many psychologists now believe short-term memory may be rather smaller—perhaps just four items. But the "seven plus or minus two" rule remains, because it works. It seems to be a number that people can cope with. When I ask users to divide items into chunks, they tend to come up with around half a dozen groups.

There's no reason you can't divide the user's options into fewer chunks. I would always use as few chunks as feels simple to your mainstream user—fewer chunks mean fewer choices and less load on the user.

You don't always need to chunk. If your user needs to find an item in a long alphabetical list or timeline, there's no point in breaking up the list into half a dozen bits. Marking out letters of the alphabet or months of the year can help users to quickly scroll to approximately the right place, but chunking is most effective when users have to evaluate several possibilities rather than locate an item on a continuous index or scale.

Organize into bite-size chunks.

Organizing for behavior

The first question a user will ask is, "What can I do with this?" so your first point of organization is to understand users' behavior: what they want to do and in what order they want to do it.

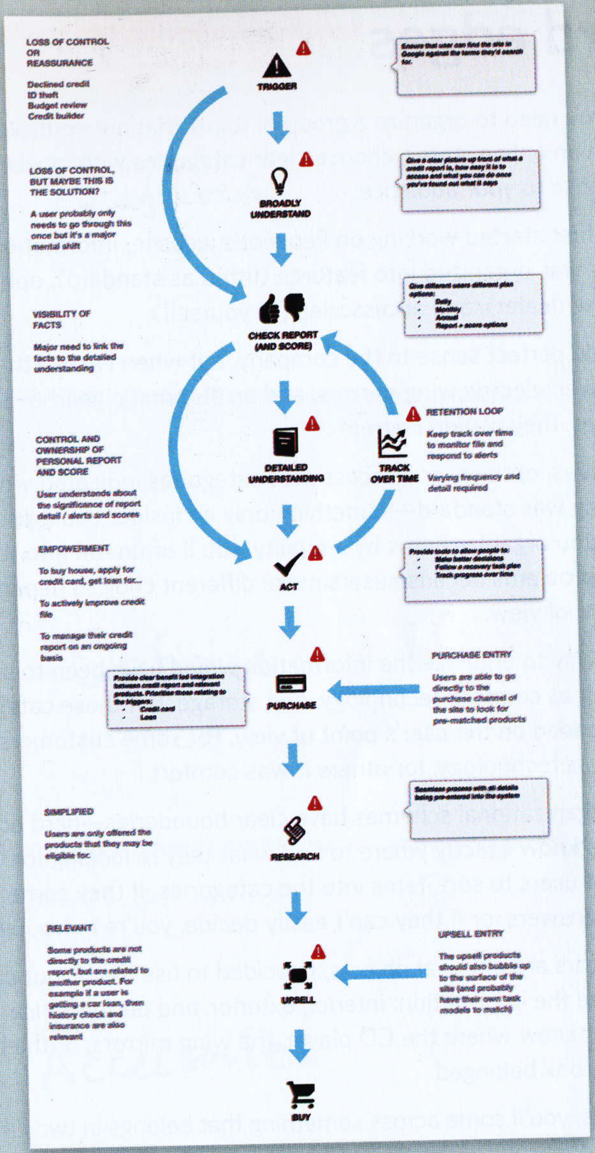
An online supermarket requires users to find the items they want to buy, add them to a shopping basket, schedule a delivery, and pay for the goods. Those are the main chunks into which the site should be divided.

People expect to begin their shopping by choosing groceries. This is also the most time-consuming part of the task, so it should be the most prominent.

People usually expect to do things in a particular sequence. It's disorienting and frustrating to break that sequence. The usual culprits here are registration processes and eligibility checks. If you can't remove them, defer them; if you can't defer them, minimize them. Find out what sequence of tasks users expect and do everything you can to stick to that pattern.

If your audience breaks down into totally separate groups who do completely different things on your website (like "doctors" and "patients"), this can be a useful first step.

The problem is that many audiences have similar or overlapping tasks. If your company provides information for journalists on its website, you'll need to give them company background information, press releases, new product information, press photographs, annual reports and staff biographies. A financial analyst wants almost the same information. If you don't have unique audiences, you probably shouldn't label by audience.



LOSS OF CONTROL OR REASSURANCE

Declined credit
ID theft
Budget review
Credit builder

Ensure that user can find the site in Google against the terms they'd search for.

LOSS OF CONTROL, BUT MAYBE THIS IS THE SOLUTION?

A user probably only needs to go through this once but it's a major mental shift

Give a clear picture up front of what a credit report is, how easy it is to access and what you can do once you've found out what it says

VISIBILITY OF FACTS

Major need to link the facts to the detailed understanding

Give different users different plan options:

- Daily
- Monthly
- Annual
- Report + score options

CONTROL AND OWNERSHIP OF PERSONAL REPORT AND SCORE

User understands about the significance of report detail / alerts and scores

RETENTION LOOP

Keep track over time to monitor file and respond to alerts

Varying frequency and detail required

EMPOWERMENT

To buy house, apply for credit card, get loan for...

To actively improve credit file

To manage their credit report on an ongoing basis

Provide tools to allow people to:

- Make better decisions
- Follow a recovery task plan
- Track progress over time

Provide clear benefit led integration between credit report and relevant products. Prioritise those relating to the triggers:

- Credit card
- Loan

PURCHASE ENTRY

Users are able to go directly to the purchase channel of the site to look for pre-matched products

Seamless process with all details being pre-entered into the system

SIMPLIFIED

Users are only offered the products that they may be eligible for.

RELEVANT

Some products are not directly to the credit report, but are related to another product. For example if a user is getting a car loan, then history check and insurance are also relevant

UPSELL ENTRY

The upsell products should also bubble up to the surface of the site (and probably have their own task models to match)

Mapping users' behavior will help you see how to organize your software.

Hard edges

When you need to organize a group of items that are equivalent (like books in an online store), choose clear categories with labels that make sense to your audience.

When I first started working on Peugeot's website, information about each car was organized into features (fitted as standard), options (fitted by the dealer), and accessories (fit yourself).

This made perfect sense to the company, but when I asked them to sort a CD player, electric wing mirrors, and an automatic gearbox into those categories, they couldn't agree.

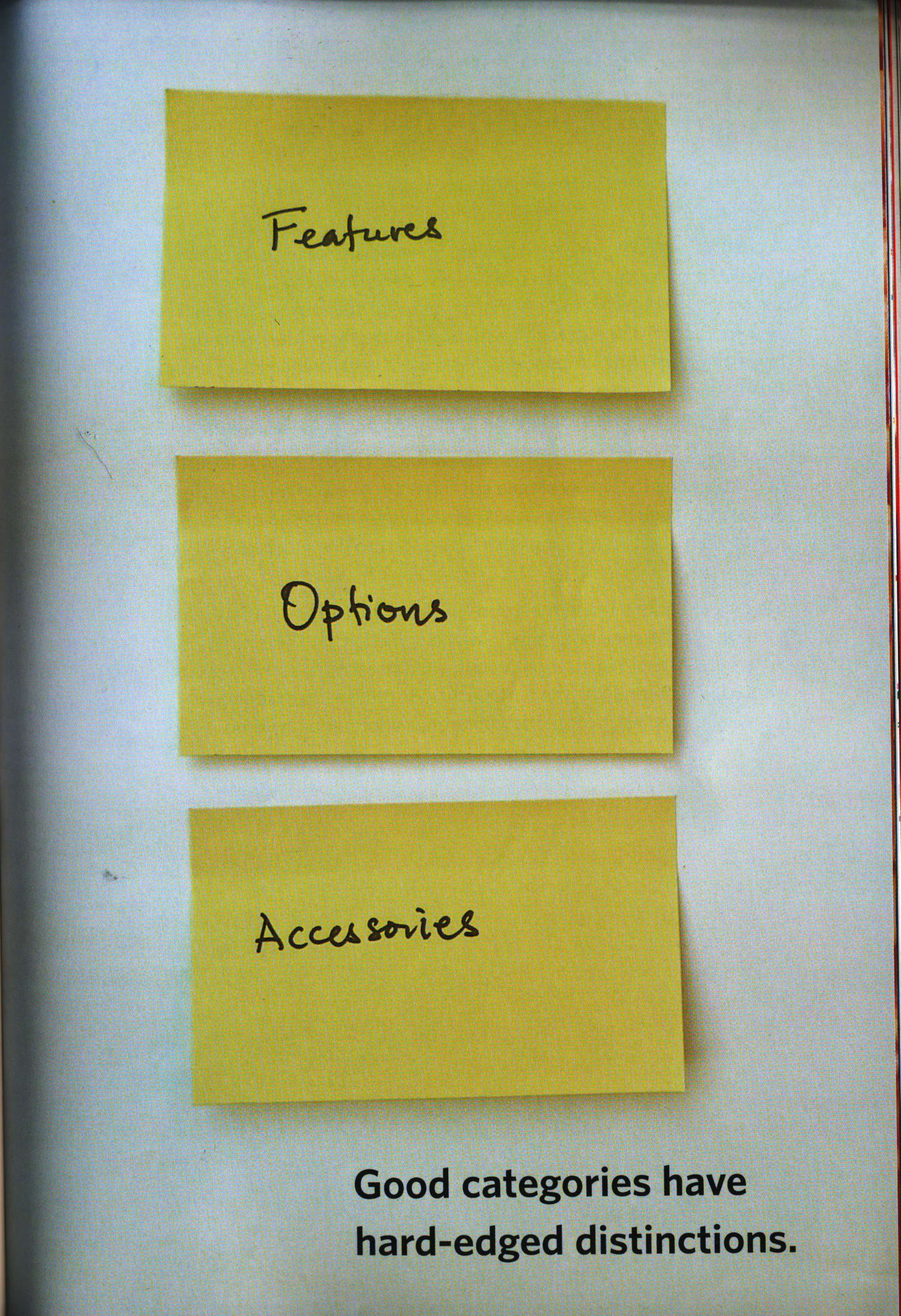
The features, options, and accessories categories indicated whether something was standard—something only an insider could possibly know. If you organize items by a quality, you'll often run into these kinds of problems because users make different choices depending on their point of view.

Another way to organize the information would have been to sort it by type, such as comfort, technology, and storage. But these categories also depended on the user's point of view. For some customers, climate control was technology, for others it was comfort.

Simple organizational schemes have clear boundaries—hard edges—so that users know exactly where to find what they're looking for. Ask a handful of users to sort items into the categories. If they come up with different answers, or if they can't easily decide, you're in trouble.

Because cars are physical objects, I decided to use the layout of the car to organize the information: interior, exterior, and performance. All the customers knew where the CD player, the wing mirrors, or the automatic gearbox belonged.

Sometimes, you'll come across something that belongs in two categories. Too much duplication leads to confusion, but sometimes it is unavoidable. Tomatoes are a fruit, but you normally find them among the vegetables at a supermarket, so they must appear in both categories. The simplest categorization is usually the one with the fewest duplicates.



Good categories have hard-edged distinctions.

Alphabets and formats

There's an old joke: where does "finish" come before "start"? In the dictionary.

Alphabetizing a list jumbles items up. So while alphabetical lists look simple, they're often hard to use. If users don't know the correct word for what they're trying to find, they're lost. Are you looking for a jacket or a sport coat? Do you want to speak to someone in Marketing or Sales and Marketing? Alphabetical lists work well for indexes of proper nouns—where there's a "correct" word to describe something—like surnames or countries. Otherwise, there are usually better alternatives.

Arranging content by format (words, pictures, videos) is another way of categorizing that looks simple but turns out to be unhelpful in the real world. If you're reading about Hawaii, you want to see photos and videos then and there. Going back to the start to find videos is just too much work.

The only situations I've come across where organizing by format makes sense are conference programs in which some formats, like tutorials, require a different registration process. In other words, some formats were used differently by the participants. But these are exceptions—it's usually simplest to organize conference information by time.

Alphabetizing often jumbles items up.

Search

There are a couple of big myths surrounding search and simplicity.

The first is that some users find searching easier than browsing—that there's a subgroup of people who always prefer to search. It's one of those myths that feels right. However, when Jared Spool tested a group of 30 users in over 120 shopping tasks, he didn't find a single individual who always preferred to search.

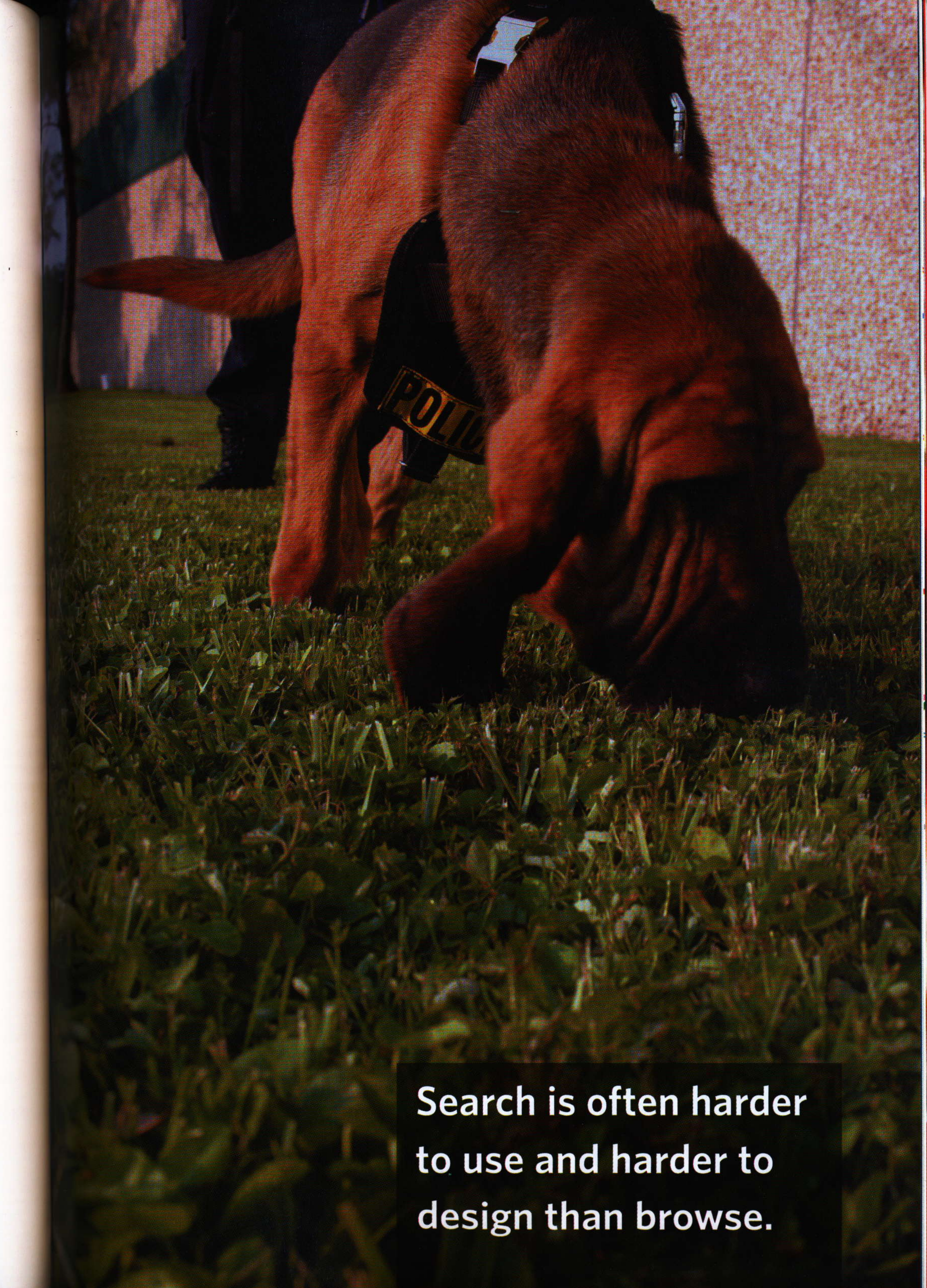
Instead, he found that when websites didn't offer links that looked like a good bet, users would search. That's not so surprising when you consider how much effort it is to think of an exact search term, type it in, and pick out a helpful search result. It's much easier to click on a link that looks like it'll carry you in the right direction. Browsing requires less mental effort up front; people will usually take the path that avoids having to think too much.

The exception is when you're asking people to pick one known item from a very large number of similar items, such as a specific track from the millions of downloadable tracks on iTunes. In this situation, yes, people will tend to search. In that case, browsing is more daunting than searching.

One of the hidden benefits of browsing is that when people look over the main links on a website, or the controls on an interface, they get an idea of what the software can do. Who needs introductory help messages when the interface speaks for itself?

The other myth is that designing a search is easier than organizing links to content. Perhaps it's because sites like Google make search look effortless that we assume it is easy to do. My experience is that it's harder to create a simple search interface. You need to take into account spelling mistakes and synonyms in users' search terms. Also, the search results themselves need to be organized. Take a look at a Google results page and you'll see a sophisticated layout that has been chosen to match the contents of your search.

If you're designing a simple user experience, it's usually best to begin with the basic organization and then move on to designing search.



Search is often harder to use and harder to design than browse.

Time and space

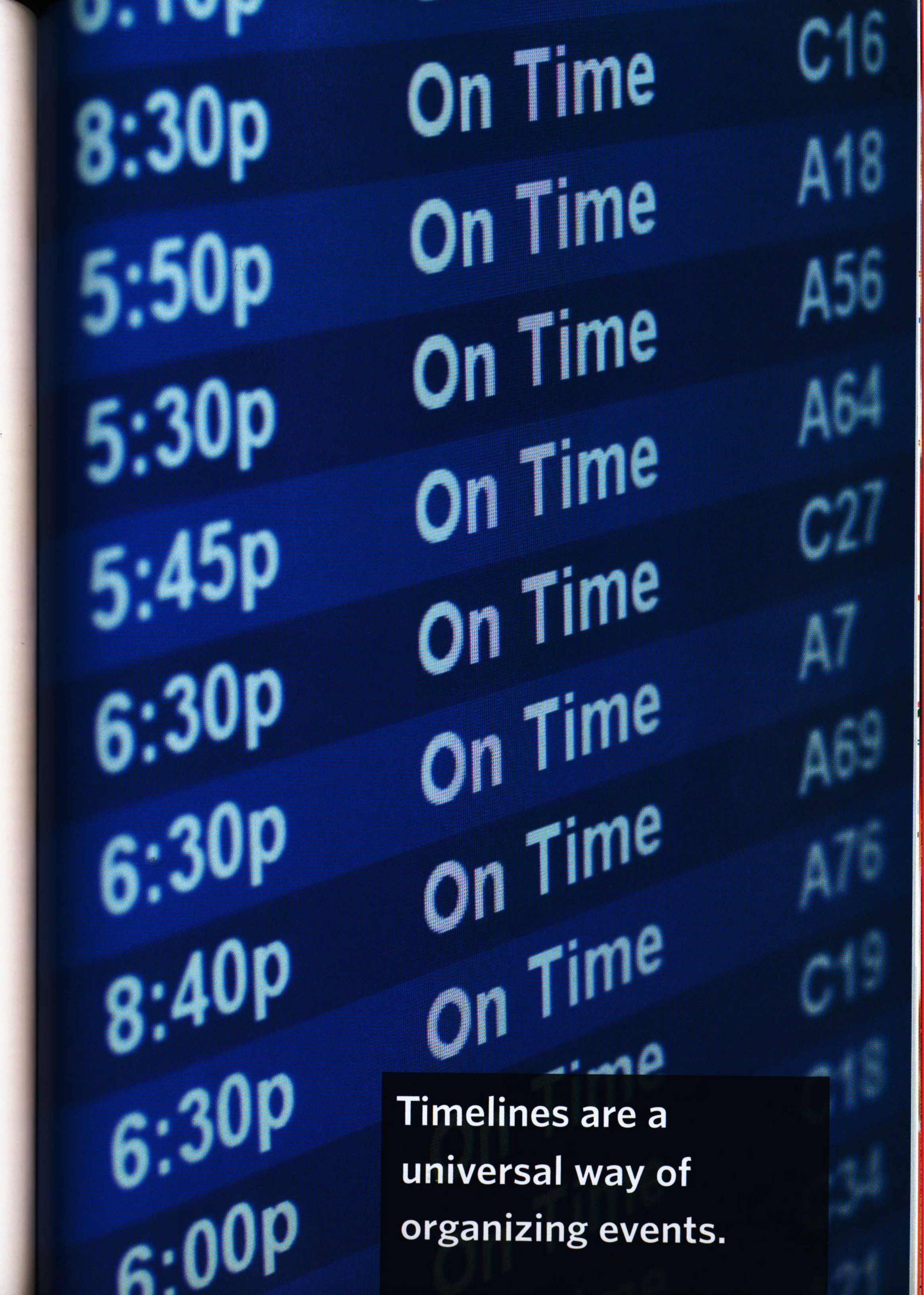
Setting events on a timeline is simple and powerful. It works best if the events are of similar duration so that users don't find themselves zooming in and out of a timeline or calendar very often. Although there may be other ways to organize the same content (such as conference themes), organizing events by time gives your audience a clear way to make sense of things.

Physical objects like hotels and countries can all be organized by space, as long as the users are familiar with the layout. For instance, you can organize a hotel website by an imaginary walkthrough of the hotel: concierge, front desk, dining, meetings and events, gym, rooms, suites. People have reasonably good memory for spaces, so this is often a good choice.

Visualizing time and space in diagrams can create some problems.

If you're plotting company offices or holiday destinations, you have to cope with the fact that some areas, like Europe, will be very crowded while others, like the Pacific Ocean, will be almost empty. The same goes for plotting events on a day planner (not much happens between 1:00 a.m. and 5:00 a.m.).

Sometimes it's useful to see variations in density, such as seeing that there's a concentration of bus services around rush hour. Other times, it can make information hard to pick out. I can set my computer's clock by clicking on a map of the world—but Paris and London are just a few pixels apart, even though they're in different time zones.



Timelines are a universal way of organizing events.

Grids

It's remarkable how far a tidy layout can go in making a design feel simple.

The form on the opposite page (top) is an interface for searching for train tickets that my company designed. It worked fine in user testing, but people hesitated over it. We revisited the layout and decided we could simplify it. We looked at the number of imaginary horizontal gridlines that were used to line up the field and simplified them. We also got rid of the heavy blocks of color that marked out the areas of the field and let the white space and alignment to the imaginary grid do the job.

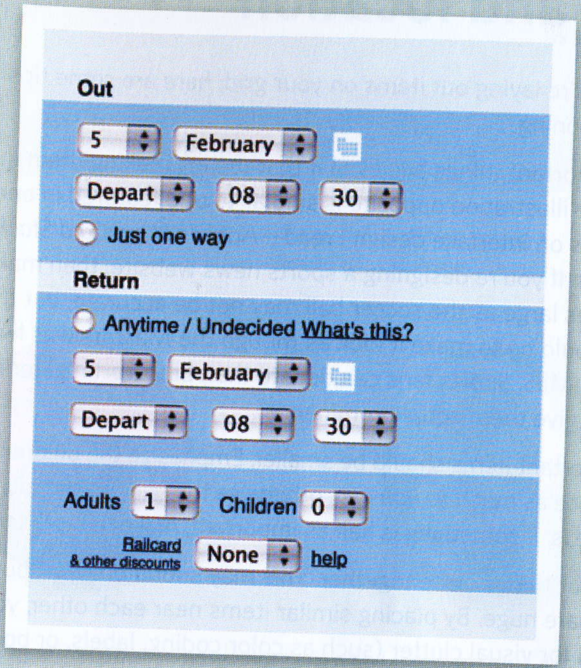
The result was a layout that felt simpler to use, even though we hadn't changed the labels or programming of the form at all.

Lining items up using an invisible grid like this is an effective way of drawing the user's attention across the screen. It says, "Here's where to look next," without relying on bright colors or flashing images. The simpler the grid, the more powerful the effect.

Having even a few elements out of position can spoil a grid. In the example opposite, only three of the seventeen controls were out of position, but this was enough to disrupt the layout.

Grid layouts can feel regimented and constricting. One way around this is to make the layout asymmetric—for instance, by having an odd number of columns. Another is to have a few elements that straddle several columns. Take a look at websites and magazines like Wired or the *Guardian* online and you'll see they're really designed around a regular, asymmetric grid.

Before



Out

5 February

Depart 08 30

○ Just one way

Return

○ Anytime / Undecided What's this?

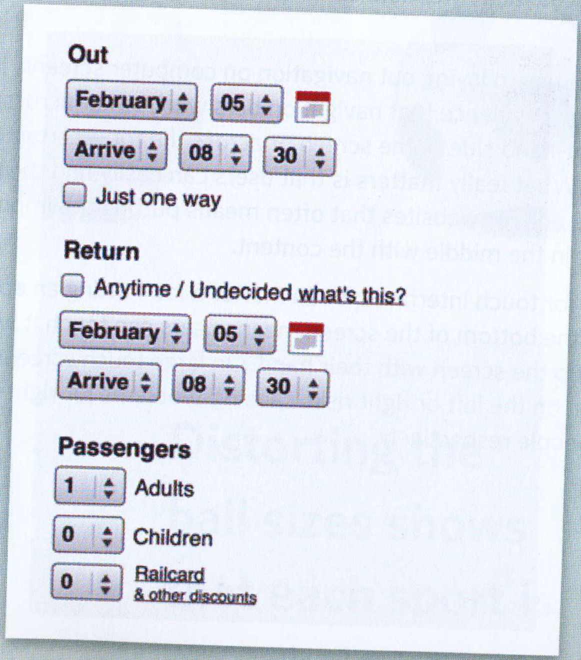
5 February

Depart 08 30

Adults 1 Children 0

Railcard & other discounts None help

After



Out

February 05

Arrive 08 30

□ Just one way

Return

□ Anytime / Undecided what's this?

February 05

Arrive 08 30

Passengers

1 Adults

0 Children

0 Railcard & other discounts

Size and location

When you're laying out items on your grid, here are some tips for size and positioning.

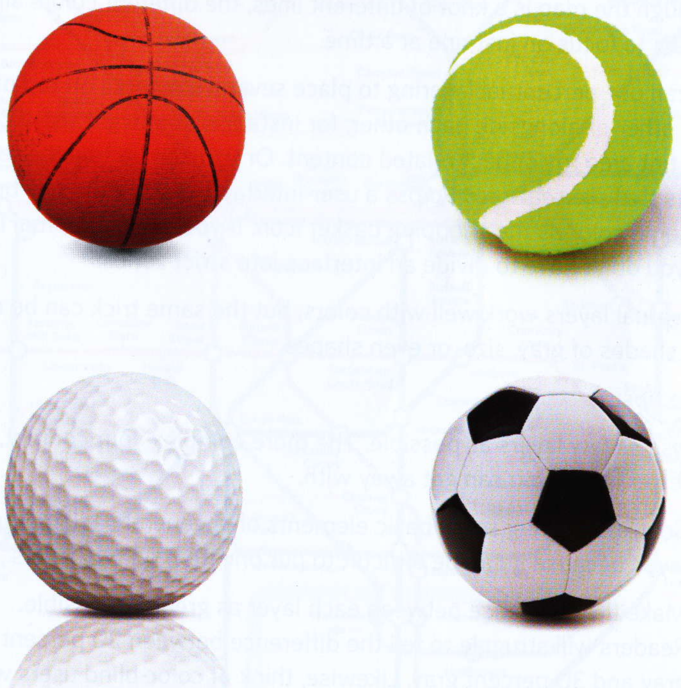
Make important things big, even if that means making them out of scale. The illustration opposite is similar to one featured in one of the first books on interface design I read—Apple's *HyperCard Stack Design Guidelines.* If you're designing a sports news website, then making the golf ball as large as the soccer ball may not be accurate, but the alternative would be to make it look as though the Masters was less important than MLS. Sports fans can debate that, but sports editors would prefer to give them equal prominence.

Less important items should be smaller. Emphasize the difference in importance as much as you can, otherwise the user will get distracted. A good rule is: if something is half as important, make it a quarter as big.

Put similar things close together. This may sound obvious, but the benefits are huge. By placing similar items near each other, you reduce the need for visual clutter (such as color coding, labels, or boundary boxes) to explain how they are related. You also make it easier for users to focus their attention, because they don't have to look all over the screen.

When it comes to laying out navigation on computer screens, I've never seen any real evidence that navigation bars work better across the left- or the right-hand side of the screen or across the top—certainly not for websites. What really matters is that users can easily find the buttons they want, and for websites that often means putting the important links right in the middle with the content.

However, for touch interfaces it can matter a lot. Putting an app's navigation at the bottom of the screen means users can touch it without covering up the screen with their hand. On large touch screens, putting navigation on the left or right risks causing problems for right- or left-handed people respectively.



Distorting the ball sizes shows that each sport is equally important.

Layers

The London Tube map crams a lot of information into a very small space. Over three hundred stations on thirteen lines are squeezed onto a pocket-sized map. One way the map stops all this information getting jumbled up is by using a technique called perceptual layering.

Each tube line has a distinct color and so seems to sit on its own layer. Without noticing, readers tune in to the color of the line they're interested in and exclude other lines from their conscious thought. Although the map is a knot of different lines, the different colors allow readers to focus on just one at a time.

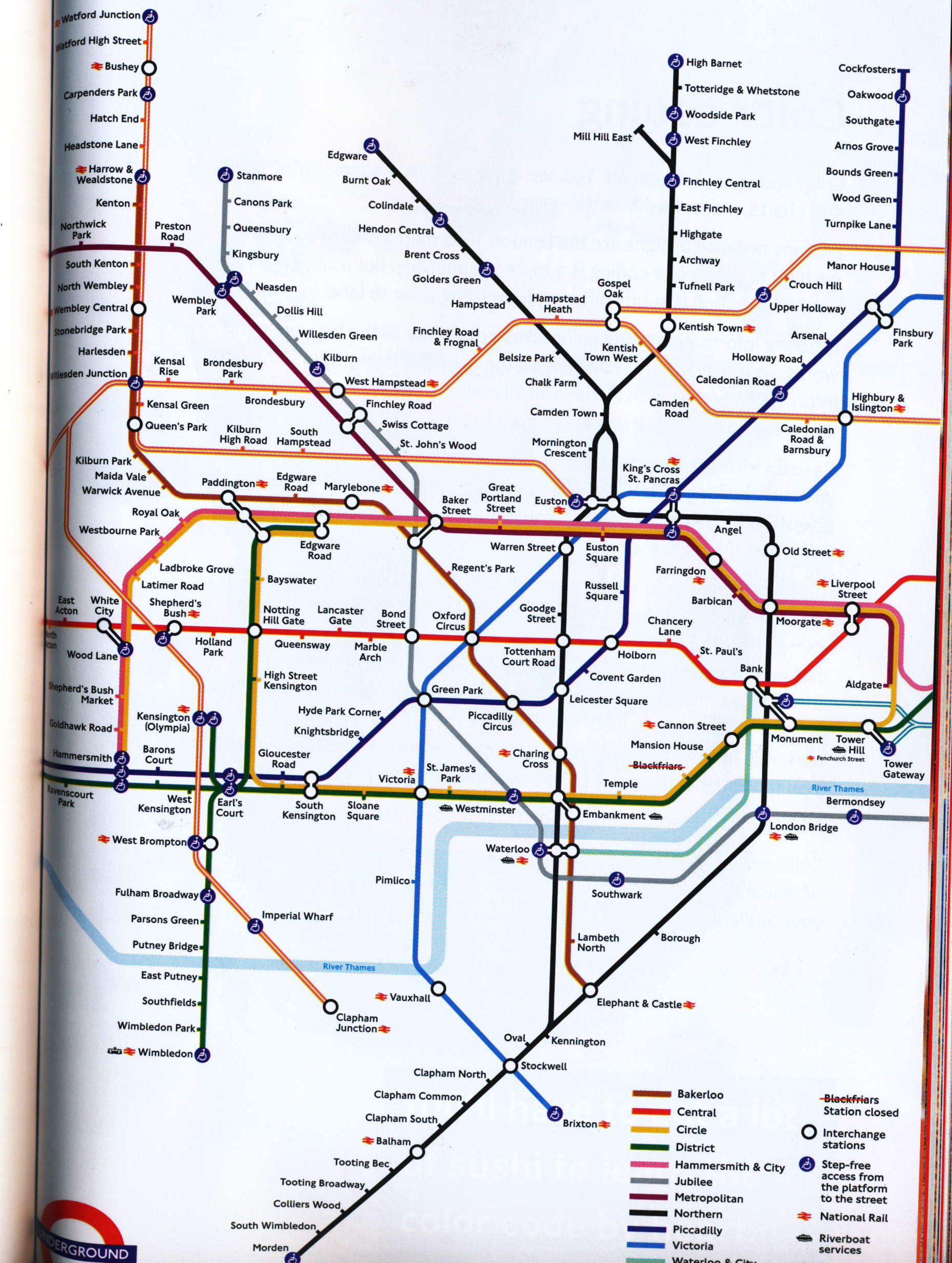
You can use perceptual layering to place several elements on top of each other or alongside each other; for instance, you might use a colored tint area to connect related content. Or you can tie together elements that are scattered across a user interface, making the buy button the same color as the shopping basket icon. If you use perceptual layers, you don't have to divide an interface into strict zones.

Perceptual layers work well with colors, but the same trick can be used with shades of gray, size, or even shapes.

Some tips:

- Use as few layers as possible. The more complex your content, the fewer layers you can get away with.
- Consider putting some basic elements on a general background layer, because it can be difficult to put one item on two layers.
- Make the difference between each layer as great as possible. Readers will struggle to tell the difference between 20 percent gray and 30 percent gray. Likewise, think of color-blind users when you're choosing colors.
- For categories that are more important than others, use bright, saturated colors to make them pop off the page.
- For categories that are equally important, use perceptual layers with the same brightness and size but vary the hue (like the lines on the London Tube map).

A quick way to figure out if your design is working is to squint at the screen and see if the layers are distinct.



Color coding

Color coding is widespread. You see it in hospitals, folders, traffic lights, size charts, maps, dashboards—everywhere.

Perhaps because designs like the London Tube map are so successful, we tend to think color coding is a route to simplicity. But using colors to layer information is subtly different from using color to label information.

Layering information using color takes advantage of the way the mind works, so it places very little load on the user. But using colors to label information comes with a cost: like all codes, it takes time to learn and decode, so it requires extra effort from the user.

Casual visitors may not have time to learn your code. The more colors you use, the longer it will take to learn. And if you are not rigorously consistent in using the colors throughout your design, users won't be sure what the code means.

Another problem is in taking a system that's well known in one context and using it elsewhere. For instance, some British food labels include traffic light colors to suggest whether they contain items like salt or fat that people need to limit. While the traffic light colors are familiar to drivers, their meaning needs to be explained all over again to food shoppers, so not much is gained. And because the red and green colors don't work well for many color-blind people, they're not a universal solution (real traffic lights use position as well as color as part of their signal).

Adding color when it is not needed creates confusion.

Color coding works best when you are sure people will spend a long time learning and reusing your design, or when you're using a code your audience has already learned.

You'll have to eat a lot of sushi to learn the color code by heart.

Desire paths

The next time you're in a park or a stretch of grass that's visited by a lot of people, keep an eye out for two things. First, look for the footpaths that a planner or architect has laid down through the park. These paths probably show how a designer, from an aerial view, thought people should move through the space—often in straight lines or a tidy, geometric pattern. Then look for the tracks that people have made as they wander across the grass. These well-worn "desire paths" are often quite different from the paved routes.

Looking down on his plans, the architect thought he'd designed the perfect layout. But when you walk through the park, you can see exactly why people have created the desire paths—taking a shortcut to a gate, avoiding a poorly lit corner, linking up two parallel routes. Walking the desire paths always feels simpler than sticking to the "proper" footpaths.

If you're plotting the user's path through your software, it's important not to fall in love with the neat lines and tidy organization you see in your plans.

Walk through the software repeatedly, and see what catches your eye (squint at your screen layouts!). Watch other people doing the same thing.

Simple organization is about what feels good as you're using the software, not what looks logical in a plan.

People won't always follow the path you set.

Part 6

Hide

Hide

Hiding features behind a hatch or sliding panel is a popular solution to the problem of simplifying a DVD remote. I own several remote controls that take this approach.

Another way to hide buttons is to use a touch-screen remote control. In those designs, the most frequently used features are on display and the rarely used ones are hidden in menus deeper within the device.

You can buy those kinds of programmable touch-screen remote controls—they're sold on ease of use and they cost about twice as much as a typical DVD player. That shows just how far some people are prepared to go for simplicity.

Whether you go down the expensive high-tech route or add a couple of cents to the cost of your remote control by hiding features behind a plastic hatch, hiding has a big advantage over organizing: users aren't distracted by unwanted details.

For some people, hiding is a first step to removing an unloved feature: hide it, let it wither in the dark, then kill it. I'm dubious of that approach. Terminating any feature means you'll need to go through the arguments I discussed in the *Remove* section, whether or not you've hidden it first. It's usually better to end it quickly.

Hiding anything means putting a barrier between the user and the feature, whether it's a plastic door on a remote control or a sequence of clicks on a website. You must carefully choose what to hide so as not to inconvenience the user.

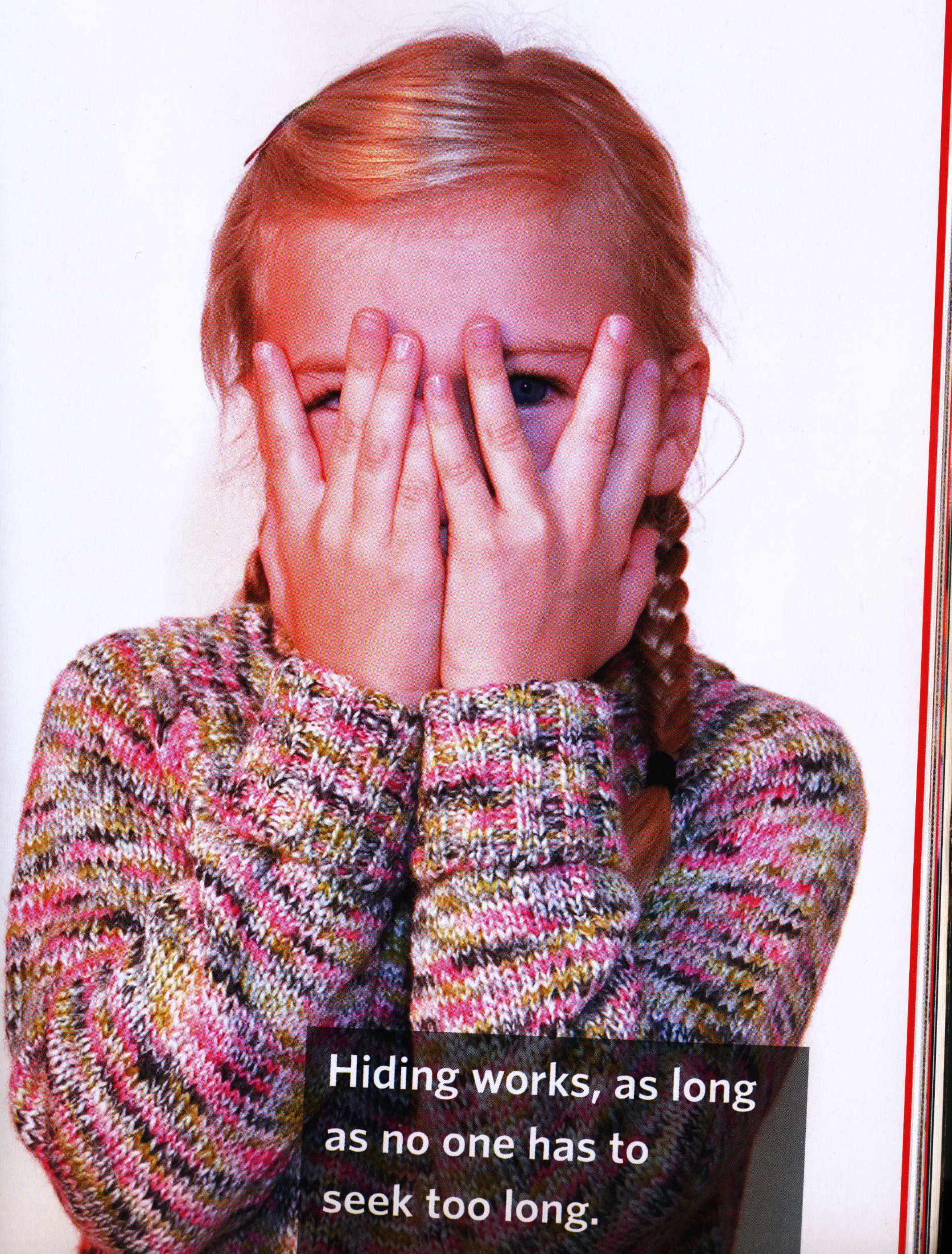
Hiding some features is a low-cost solution. But which features should you hide?

After you hide

Hiding, then, depends on four things:

- Hide one-time settings and options.
- Hide precision controls, but let expert users choose to keep them revealed.
- Don't force or expect mainstreamers to customize, but offer this option for experts.
- Hide elegantly; that is, hide completely and reveal just in time.

The three strategies so far—remove, organize, and hide—fit together neatly: remove what you don't need, organize what you do, hide what you can. But the final strategy, displace, is really about rethinking the interface entirely.



Hiding works, as long as no one has to seek too long.

Part 7

Displace

Displace

The fourth strategy for simplifying the DVD remote control is to cheat.

Designers who take this approach strip the remote control down to a few basic actions, like play and pause, and manage all the other features via a menu on the TV screen. The remote control itself is approachable, easy to understand, and simple to use.

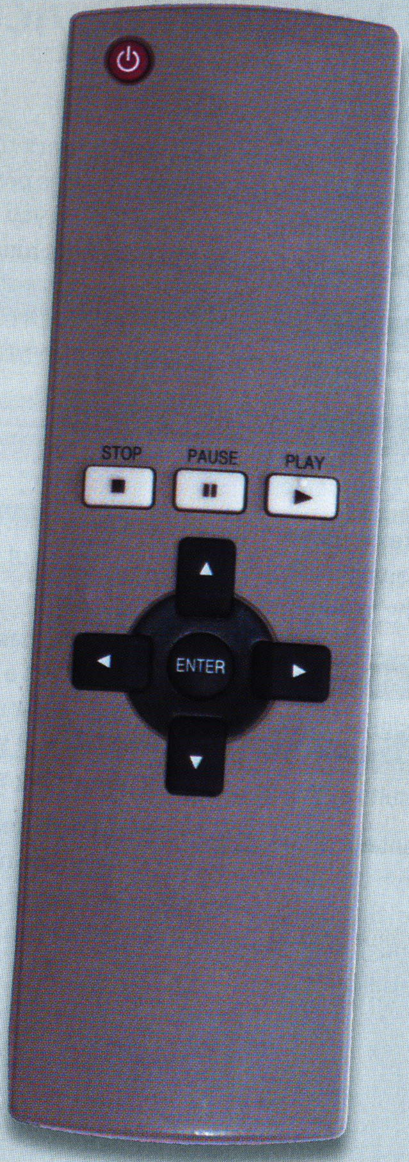
Another advantage of this strategy is that it makes good use of the remote control. Users only have a few buttons to learn and they can easily be distinguished by touch—so it's easy to use in the dark while you're watching a DVD.

It's also far cheaper to make use of the existing TV screen than to add an expensive display to the remote control. The TV screen is well suited to this. It can display an infinite number of different menus and it's bound to be in a location where the user can see it clearly.

The disadvantage of this approach is that if you displace all the features, then it's hard to guess what the remote control can do. If you had to find and access the play function by navigating into a menu, that might seem obscure and tedious. That's why most people end up leaving a few basic controls on the remote.

Also, though you've simplified the remote control, you still have the problem of designing a simple on-screen menu system (using the strategies of remove, organize, and hide).

But if you understand the trade-offs, displacing the right roles to the right devices works well. One of the secrets of creating simple experiences is putting the right functionality on the right platform or part of the system.



Why not take some buttons off the remote and use an on-screen menu instead?

Trust

Displacing tasks is easiest when you're dividing them between two devices that have to be used together in a specific way. The DVD remote control has to be used with the TV display, so it's fairly easy to see what each should do.

When you're not sure how the devices will be used together, displacing becomes harder.

You can't be sure how the RunKeeper mobile app and website will be used. Some people may not have mobile phones and will just want to use the website. Some people may stick exclusively to using their mobile phone. Some people may do a bit of both.

When that uncertainty creeps in, you find yourself duplicating functionality between platforms. So it is with RunKeeper, where only a little of the functionality is displaced between website and mobile.

You need a sense of certainty to be able to displace tasks effectively.

If you're going to displace tasks to be the responsibility of the user, you have to trust that the user will take on those tasks.

Trusting the audience is hard. Designers are used to watching them fail in user tests. Programmers are used to thinking of all the ways a system could go wrong so that they can design for error. Product managers want to provide users with interactive tools that take on all the hard work. And sometimes the unspoken purpose of software is to make users behave in ways that are convenient for the designer.

In other words, we often treat users like children. But in protecting users from making errors or finding their own solutions, we often deny them the chance to make their own decisions. No wonder users often feel rebellious or resentful toward computers.

The only way to build that trust is to try out prototypes and mock-ups with users. When you get the balance of tasks right, letting users focus on choosing and directing, and having the computer focus on remembering and calculating, you'll create experiences that are simple and surprising because of the creativity users can bring to them.

Computers often make users uncomfortable because they control and direct users' behavior. Simple experiences require trust.