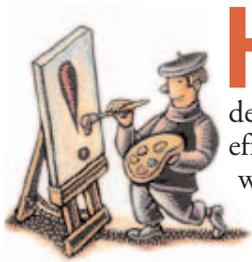


Slanty Design

This new take on usability yields desirable (and the absence of undesirable) user behavior, even as it furthers grander corporate goals.



Human-computer interface designers must move beyond user-centered design if they are to create more effective systems that integrate well with the complex business and social context of modern life. Slanty design is an approach that extends user-centered

design by focusing on the things people should (and should not) be able to do with the product(s) behind the design. It includes an evaluative phase to identify (and eliminate) unforeseen, especially unwanted, side effects.

Usability is not always king. Douglas Englebart, the original developer 40 years ago of the computer mouse, once said “If ease of use was the only valid criterion, people would stick to tricycles and never try bicycles.” We need more than usability to make things work properly. Design is (or should be) a conversation between users and design experts and between desired outcomes and unwanted side effects. We must ensure that the needs of one are not subsumed by the demands of the other. In the bad old days when the waterfall development model held sway, programmers built systems based on what they thought users wanted; it didn’t work so well, producing shelves of requirements specifications

without delivering better products. In our own more enlightened times, we have user-centered design, which gives far greater weight to user experience, desirability, and value. The classic approach to user-centered design was spelled out in [4], with a modern perspective available online in [3].

Users are involved and consulted at every stage of the process, validating or questioning the decisions of programmers and designers. As the results of involving users show, this input has had a demonstrable effect in improving systems [5]. And while any design tends to be an instantiation of trade-offs, these trade-offs are made more explicit within a user-centered approach.

Why would anyone consider moving beyond this inclusive process? Because it doesn’t always deliver the best possible system. For a start, user-centered design is grounded in the user’s own current behavior, which is often less than optimal. Unusable systems often reflect a discrepancy between the user’s approach to and model of their task and that of the system [1], but it is not necessarily the case that the system is wrong. There is also a problem in defining who the system is being built for. There are often competing (and sometimes diametrically opposed) goals, depending on whether one consults the people who use the system, focusing on everyday activities, or more senior managers, focusing on broader

User-centered design is grounded in the user's own current behavior, which is often less than optimal.

business goals. For example, users may be interrupted in their work and so may need multiple windows to allow them to multitask effectively. But as a business, it may be the case that entering incorrect information into the wrong record would cause a significant problem, and the new multitasking approach increases the chances of this happening. As Englebart implied, moving to a new level of performance and opening up new ways of doing things may well require more innovative perspectives on the wider problem than usability approaches typically deliver.

What should we do? Bring on slanty design. Slanty design is the term I've given to design that purposely reduces aspects of functionality or usability; it can be very effective.¹ It originated from an apocryphal story that some desks in the U.S. Library of Congress in Washington, D.C., are angled down toward the patron, with a glass panel over the wood, so when papers are being viewed, nothing harmful (like coffee cups, food, and ink pens) can be put on top of them. This makes them less usable (from a user-centric point of view) but much more appropriate for their overall purpose. These desks, in the Main Reading Room, are the originals installed in 1897, but food and drink are officially banned, so this story is, at best, true only sometime in the past. However, the principle holds—that slanty design is useful when the system must address wider goals than users might have, when, say, they wish to do something that in the

grander scheme of things is less than desirable.

Apple Computer's iPod shuffle is another example of slanty design. Small, elegant, and lightweight, it holds about 100 songs, piggybacks on the wave of iPod chic, and is marketed with a lack of usability as a selling point; you can either play all the tracks in order or randomly. You can also skip to the next track. Where is Apple's famous usability? It went



Slanty-designed baggage carousel. Sloping floor keeps the area clear.

when Apple's designers removed the screen, which I suspect had far more to do with manufacturing costs than a desire to create a new musical experience for its customers. You can't see what is playing, choose your songs, or organize your musical experience. It is indeed less usable. But is it also less desirable? Probably not. It brings iPod cool down in price, making it even more widely available. It also offers a different experience in listening to music, forcing randomness onto listeners; how many people do you know who

¹Originally described as an emergent property of a design in the "Noise Between Stations" blog (www.noisebetweenstations.com) run by Victor Lombardi, an entrepreneur and consultant in strategic product/service design and management.

have ever used the random button on their CD players? The fact that people can experience music in this random way doesn't mean they actually do it. Moreover, Apple was banking on the fact that just because users didn't use it randomly doesn't mean they wouldn't want to.

Discussions with users of the iPod shuffle show they like the fact that it opens up their music collection to them, helping them rediscover tracks they once enjoyed but now rarely listen to or may have forgotten, obscured as they are by too many mediocre songs beforehand. These users explain how the random playlist provides them with a different level of engagement with their music, as tracks become more like individual songs in their own right, played out of their usual context of prior and subsequent tracks, and how they often discover new meaning in the lyrics. Sometimes the contrast between tracks highlights great changes of mood in the music; at other times tunes seem to form a soundtrack to the listener's life—a serendipitous coincidence of music and personal events made more noticeable by chance.

Google's Gmail is yet another example of slanty design. Deleting email from a Gmail account is very difficult; the commands are buried in menus involving many steps. While we may feel that deleting irrelevant email is something we actively want to do, we might conclude it is less usable than it should be. So why have built-in difficulty and obstruction to usability? Because Google uses your body of email to mine for information it uses to target the ads it delivers to generate revenue; indeed, deleting it would be detrimental to the service. By taking a wider perspective, Google is asking us to question our assumptions about the way we deal with email.

Why would we view deleting email as a sensible activity? It takes time, cognitive effort, and is potentially costly if important documents and contacts disappear. By providing a large amount of storage space for free, Gmail reduces any resource pressure, and by making the deletion process difficult it tries to re-educate us to a new way of operating, which also happens to achieve Google's own wider business goals.

Focusing on wider issues than immediate usability is not itself new. Identifying what needs to be designed for all stakeholders seeks to address the wider issues (such as overall business performance) of the system in context [6] but often focuses on key users and the tasks they want to accomplish. This focus often comes at the expense of identifying things that should not occur, perhaps unwanted side effects upon the introduction of a new system. The paradox that making things easier to use doesn't necessarily make them better was discussed in [2], suggesting that a design focus should be on value and fit, rather than purely on usability [6]. This emphasis on value has some attractive features, though defining value and allowing it to constructively influence design is problematic.

Slanty design incorporates the broader message, making it difficult for users to do unwanted things, as well as easy to do wanted things. Designers need to design for user non-goals—the things users do not want to do or should not be able to do even if they want to. If usability is about making it easy for users to do what they must do, then we need to have anti-usability as well, making it difficult for them to do the things we may not want them to do. So slanty design reflects two subtly different characteristics: that we need to design for broader goals than individual users may identify, and that we need to incorporate anti-usability, as well as usability, into our systems.

One way to view slanty design is that it does away with signs and instructions associated with an interaction, situation, or system. Signs and instructions are often secondary consequences, springing up in response to unforeseen or unwanted behaviors being exhibited by a new design. Designs that cede power and control to end users can be abused or subverted or produce unintended consequences by accident. These consequences are sometimes useful, but lost control is often highly undesirable, addressed only through externally imposed restrictions. But rather than a notice saying, for example, "Do not sit here," a slanty design would make it physically uncomfortable to sit in the first place, forcing us to go elsewhere.

A pharmacological analogy helps expand this idea. A powerful drug may well do exactly what the doctor and patient want it to do, say, stop rejection of a transplanted organ. However, it may be a “dirty” drug in that it has numerous and unwanted side effects. One approach is to treat the side effects with yet other drugs, but this is a somewhat circular argument, and anyone who has had serious medical treatment knows it makes you rattle like crazy due to all the pills that must be taken. A better approach is to develop cleaner drugs with the same effect on the problem but with fewer side effects. Thus “clean usability” is another new concept we should add to the usability vocabulary, delivering usability on the important issues, but without the unforeseen consequences that allow users to create a new set of problems for themselves and possibly others.

Slanty designs result from five key design steps:

- Identify user goals;
- Identify user non-goals—the things users don’t want to be able to do easily (such as deleting all their files);
- Identify wider goals being pursued by other stakeholders, including where they conflict with individual goals;
- Follow a user-centered design process to create a system with high usability for user goals and high anti-usability for user non-goals; and
- Resolve the conflicts between wider issues and individual goals, and where the wider issues win out ensure that the design meets these needs.

Following these steps helps produce slanty design, but since our aim is to deliver a system of much greater benefit to all concerned, we should also try to develop clean usability. We should revisit our design solutions, evaluating them to ensure the side effects of the new processes do not open up new areas of difficulty. This can be trouble initially, often requiring many iterations through the design cycle (or even the product life cycle) before the system becomes cleaner.

I, for one, would like to see slanty design incorporated into airport baggage carousels. The scrum of trolleys around a typical one makes it practically impossible to grab a bag when it finally emerges. A number of approaches have been tried. Big signs were first, along the lines of “Please leave a gap between the carousel and your trolley,” which didn’t work. Next, a boundary line was drawn around the

belt; this had a slight positive effect but hardly solved the problem. The best solution I’ve seen is a wide strip of brightly colored floor tiles around the belt; people are reticent to break into that boundary area, reducing the scrum slightly. But it takes only a few people to push into it, and the effect begins to break down, exacerbated by the fact that it becomes more difficult to see the floor coloring once a mass of people is on or near it.

My slanty design would put a ramp of about 30 degrees extending two meters or so up toward the belt (see the figure here). It would be uncomfortable to stand on, and trolleys would not stay there easily, tending to roll off backward or at least be awkward to handle. I might also add a small dip that would catch the front wheels, making it even more difficult to get the trolley or any other wheeled baggage on it in the first place but not enough to trip up a person.

If I was being really slanty, I’d also incorporate 2cm-high bristles in the surface, making it a real pain for the trolleys on it and not too comfy for the passengers to stay there either. Much easier for people to remain (with their trolleys) on the flat floor than negotiate my awkward hill. We’d retain the space we need, yet we could manage the short dash forward, up the hill, to grab our bags, then return to our trolleys, clearing the way for the next baggage-hungry passenger. 

REFERENCES

1. Blandford, A. and Young, R. Specifying user knowledge for the design of interactive systems. *Software Engineering Journal* 11, 6 (Nov. 1996), 323–333.
2. Cockton, G. From quality in use to value in the world. In *CHI 2004 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, Apr. 24–29). ACM Press, New York, 2004.
3. IBM. *IBM Ease of Use: User Engineering*; www.03.ibm.com/easy/page/1996.
4. Norman, D. and Draper, S. *User-Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1986.
5. Vredenburg, K., Mao, J.-Y., Smith, P., and Carey, T. A survey of user-centered design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Minneapolis, Apr. 20–25). ACM Press, New York, 2002, 471–478.
6. Vredenburg, K., Isensee, S., and Righi, C. *User-Centered Design: An Integrated Approach*. Prentice Hall, Upper Saddle River, NJ, 2001.

RUSSELL BEALE (R.Beale@cs.bham.ac.uk) is chair of the British Computer Society’s HCI group and leads the Advanced Interaction Group in the School of Computer Science at the University of Birmingham, U.K.
