A History & Sociology of Computer Windows.

Louis-Jean Teitelbaum.
PhD student,
Telecom ParisTech & EHESS
http://meidosem.com – @louije

Historical Snapshots

We spend so much time in front of windows. This way the computer has of cutting up our work and activities into stacks of little rectangles has grown so familiar that we seldom ask ourselves where it comes from, in the same fashion that we seldom think about where the book comes from. Windows help us deal with multiplicity—multiple documents, multiple tasks, multiple aspects of our lives. There is a history of windows, of how they came to be as we know them, as there's a grammar of windows, of how they help us represent, in space and in time, the abstractions that we deal in.
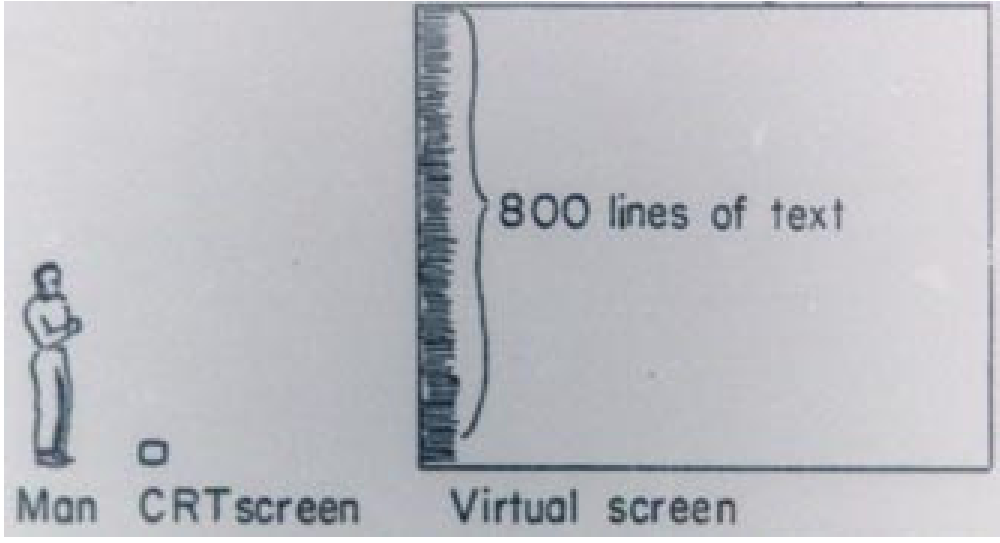
Alan Kay seems to have coined the term "window". His early work presents the computer as providing "tools in which the 'thinker' can describe his own solutions". To this purpose, windows allow the user to control the flow and organization of thought through a spatial representation. Windows are the visual representation of the process of abstraction.
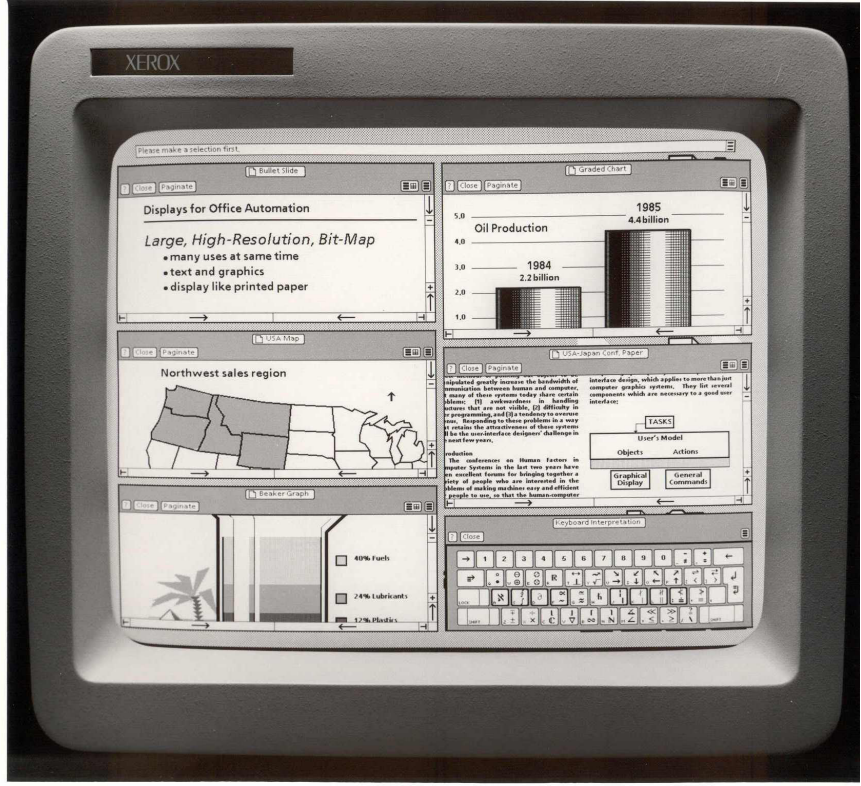
At PARC, through contact with actual users (children, workers in Xerox's printing shops), the window begins to represent actual, soon-to-be-printed-on-paper documents. It becomes a thing, only virtual. This change is known as the desktop metaphor.

Early documents show Bill Atkinson struggling with terminology: what difference is there between a window, a folder, a document? Are they containers or the content itself? Finding a straightforward meaning for each term helped structure further interface work.

Focus Machines

Don Norman tells us that "windows were originally designed as explicit supports for the conduct of multiple activities," most notably because they "can serve as reminders of the existence of the activities contained within them". Tasks can be split among windows, allowing the user to visualize and organize everything he or she is currently doing. The ways windows can be set on a given system frames the user's representation of his or her activities. While some systems impose constraints on how windows can be arranged, most of the time, managing windows' size and position is up to the user.

Tiling window managers organize windows in just two dimensions. There is no depth, no stack of documents. Switching is as easy as glancing. Windows must share scarce screen real estate, and do so in a hierarchized way. How they are laid out on the screen reveals the underlying structure of the user's activity, with, for instance, the main tasks in a larger column, and references and dependences in a smaller one. Tiled windows make up a **dashboard** of sorts, a complex but purposeful machine where every tool, every resource is in the right place.

Overlapping window systems, on the other hand, evolve in a multi dimensional virtual space. Windows can be freely stacked and resized, in such ways that they can be lost, hidden, partially covered or as large as the screen. Whereas tiling brings the computer closer to the physical machine, with its fixed displays and switches, overlapping fulfills the desktop metaphor.

Windows are organized not only in space, but also in **time**, as switching to the right one often involves cycling through stacks. Splash screens and pop-up windows show how windows can be used to manage the user's time, enabling his patience or aggressively urging him to refocus on something that just appeared in the foreground. **Ultimately, windows are about multi activity as much as they are about control.**

Full screen is not the same thing as a big window.
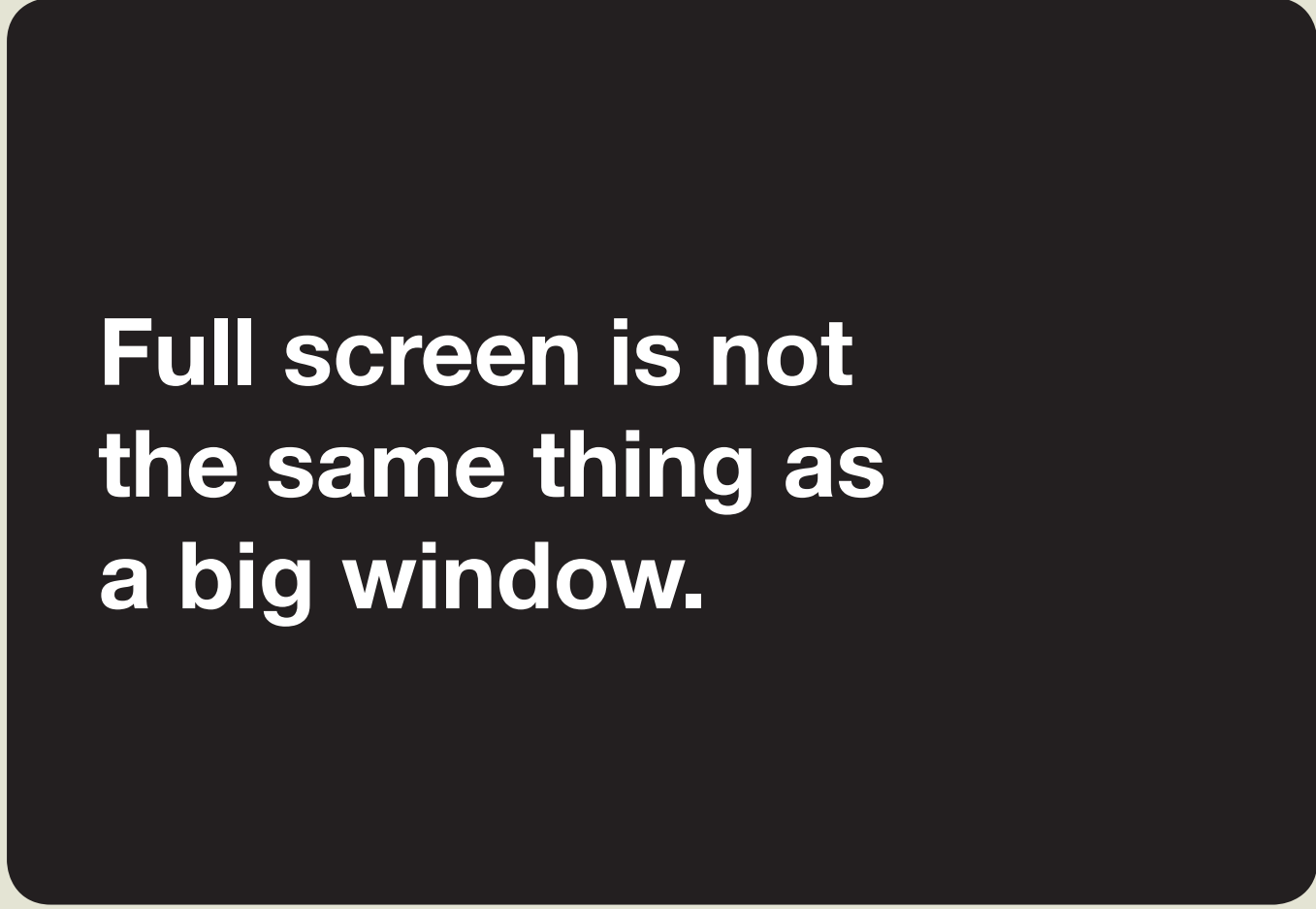
Windows are about control, and that control is obtained through framing. Windows have borders that can be "held" (or, rather, clicked), whereas the screen has no limits the mouse can attain. As with a movie theater screen, as with a projected slideshow, full screen is all about claiming the public's undivided attention.

WINDOWING ORGANIZES WHAT YOU SEE

IT DETERMINES ALSO WHAT YOU DON'T SEE

CONFIDENTIAL
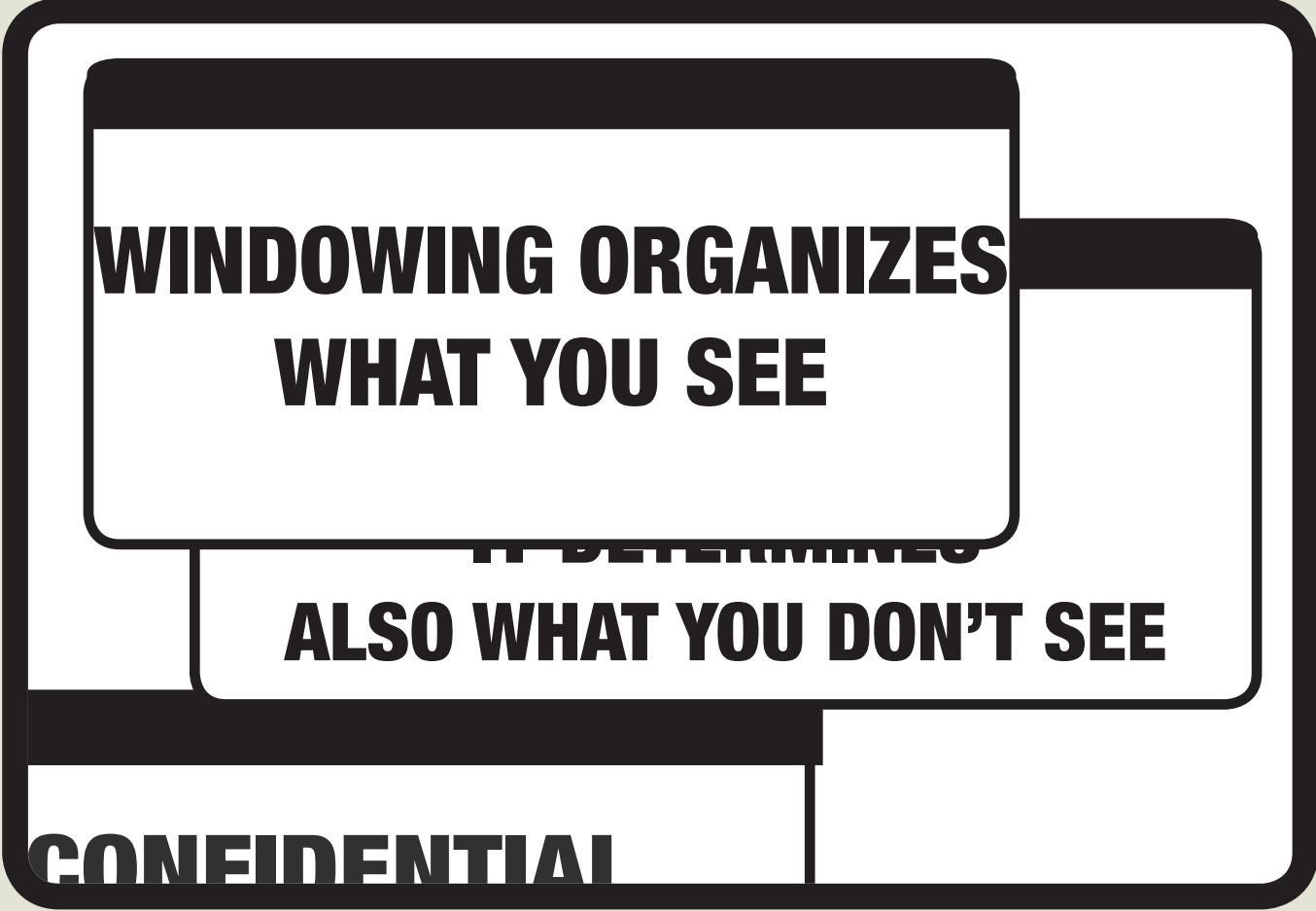
Overlapping window managers have a much larger vocabulary than constrained systems. Windows can be half hidden, near full screened, quasi tiled, and kept right at hand.
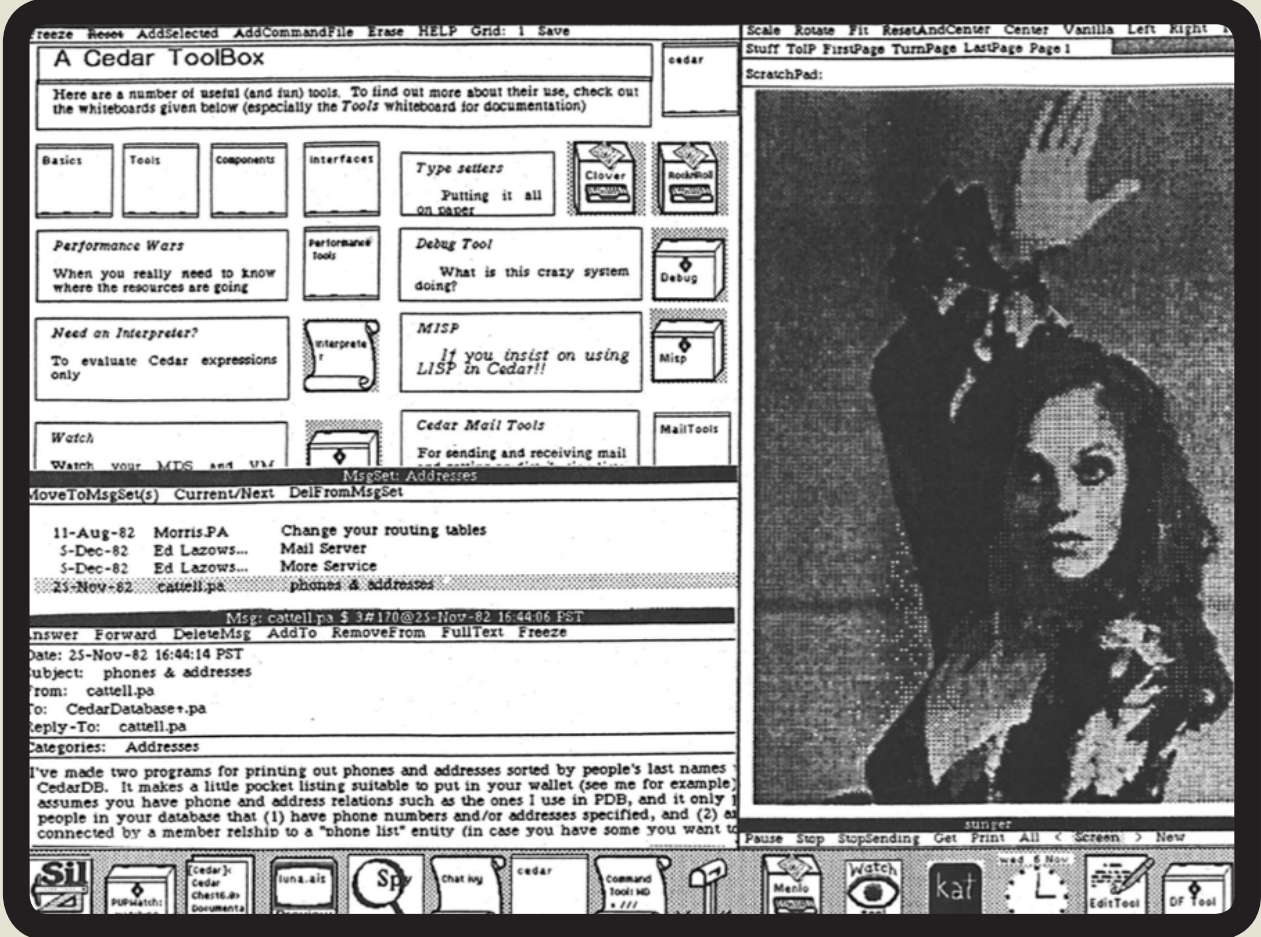
CEDAR's tiling window manager, developed by Xerox in the early 1980's. Windows are stacked up in two columns of uneven width. Layouts can be saved and restored. Tiled layouts are extremely valuable in stable, unchanging situations where clarity and structure are needed. This whole poster is a tiled layout.

HAPPEN BUT MESSES SOMETIMES THEY SENSE MAKE
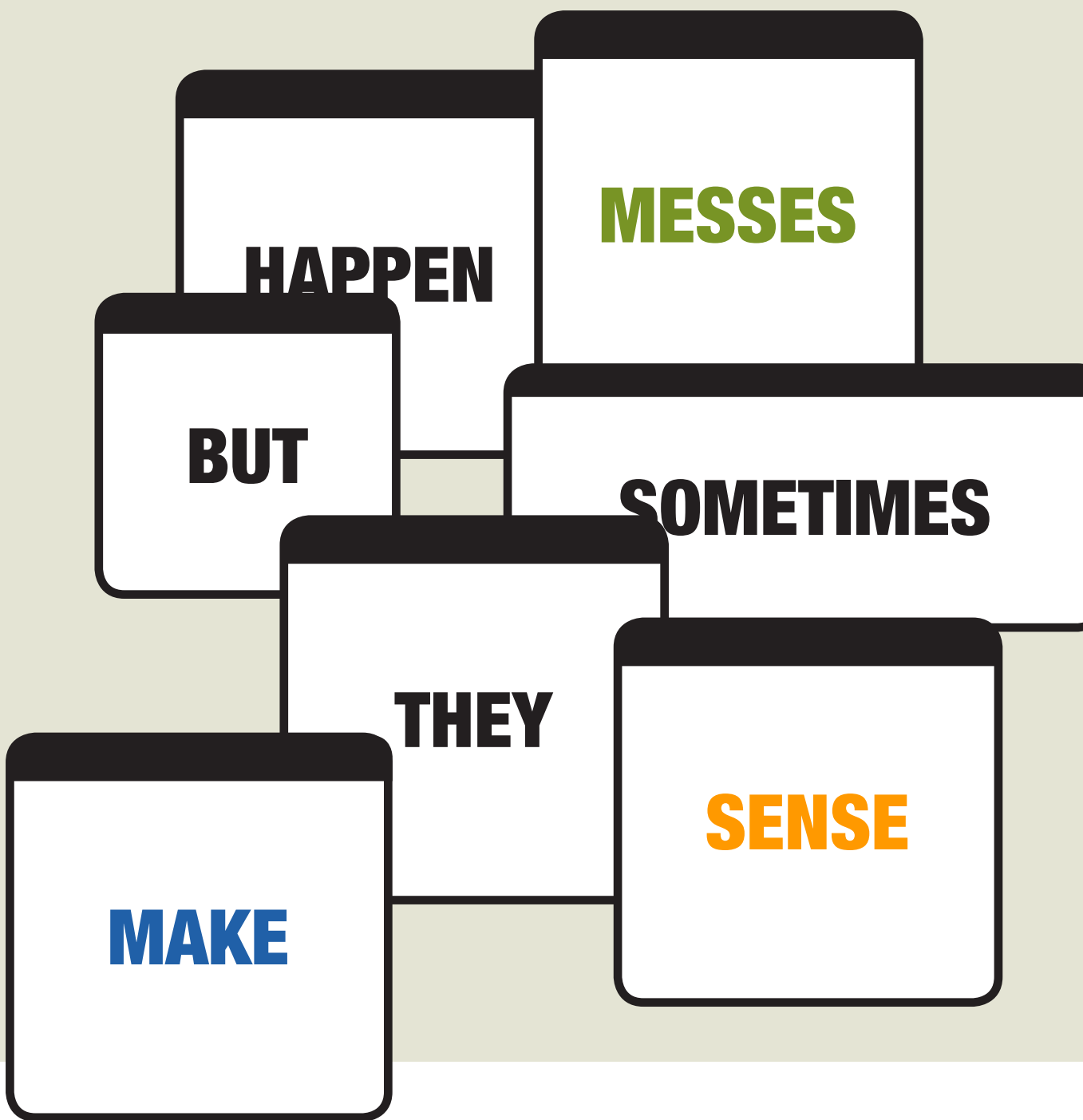
That windows are messy, and that valuable time is lost by cycling and switching to the right one, has been a leitmotiv for researchers concerned about the speed of interactions. But studies have shown that productivity is not solely a factor of speed: spatial memory and control by the user are overlapping systems' greatest strengths.

Single Again

A new category of computing devices, exemplified by "home computers" and tablet devices such as the litl webbook or the Apple iPad, and operating systems such as Google Chrome OS, has recently appeared. One of their most distinctive traits, UI-wise, is that they do not make use of windows. Every program uses the **full screen**. As the makers of those systems put it, the user should interact only with his goals, and not with the technological intermediaries: "By eliminating window clutter and computer administrative debris, you will be able to focus on your stuff" (litl); "No pixel-level window positioning" (Chrome OS).

New values are at play: **usability** no longer means "productive", or "easy to learn and use". It means that the interaction must not feel technical, that the computer stays invisible. Together with that comes a new understanding—and a new representation—of the users by the designers. The users are seen as interacting with their work or with other people, rather than with the computer. "Being at the computer" ceases to be an activity in itself.

These systems feel different, for they do not afford as much control and tailoring of one's activity. But if, precisely, one can have too much control, researchers and practitioners trying to build mature and usable software interfaces must strive to find the right balance between as much control and as little management as possible.

Apple iPad. Each application takes over the entire screen, providing its own experience.
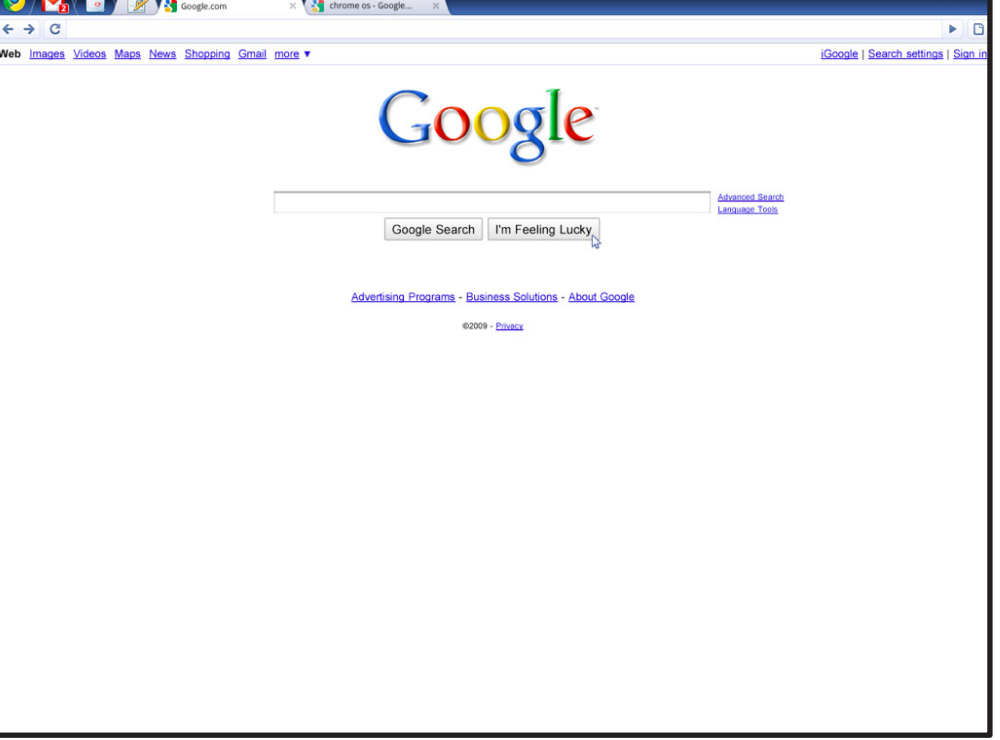
Google Chrome OS. The forthcoming system has tabs and virtual desktops, but no resizable and movable windows.
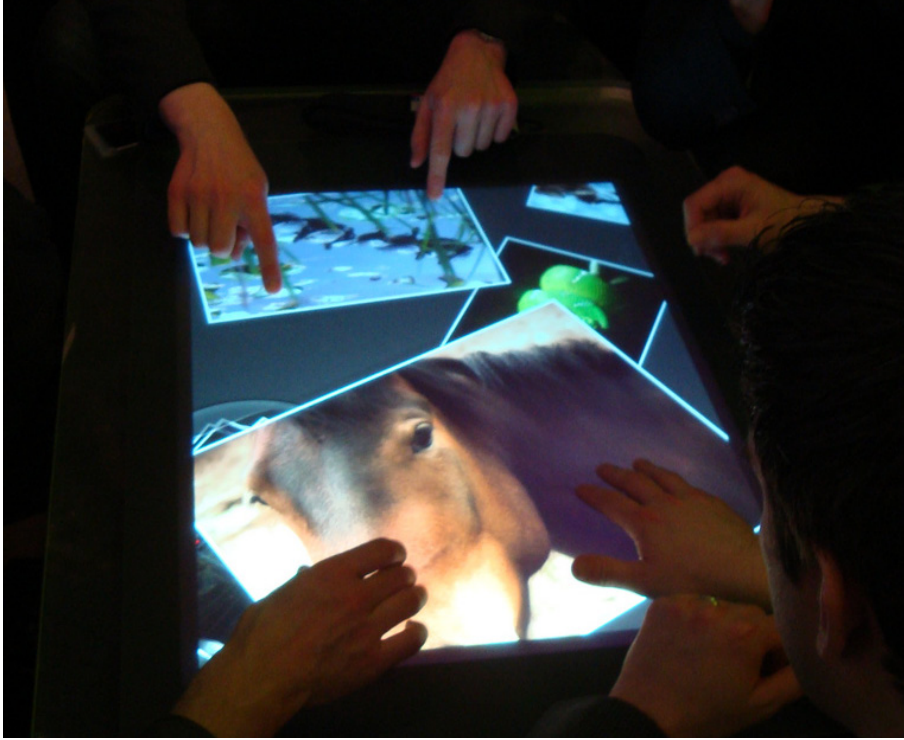
Microsoft Surface. One of the goals of Surface's design is to make the content its own interface.

Window Dressing

If, as Steven Johnson writes, there's such a thing as an interface culture, then there must be an interface fashion gallery as well.

Here are different ways in which windows can be used to express, and to organize. Each one operates through a different metaphor. All are "windows", in the technical sense. It remains to be seen whether they can be said to be the same kind of things.

The document window is a self contained representation.

The window *is* the content itself, a virtual echo of its paper counterpart.

THE VIEWER WINDOW IS JUST A FRAME FOR CONTENT TO APPEAR INSIDE. ITS PLACE, SIZE AND SHAPE ARE INDEPENDENT OF WHAT IT DISPLAYS.

The cyborg window is part document, part machine. It retains the spatial characteristics of the document window, yet it is equipped with many tools and buttons that do not belong to the document itself. Those additions blur the separation between content and interface, between data and command.

The appliance window establishes a living room metaphor, where a single device regroups every task related to its domain. What matters is not the document anymore, but the activity, the collection.