

Postures & Patterns

Jörg Cassens

Contextual Design of Interactive Systems



medieninformatik

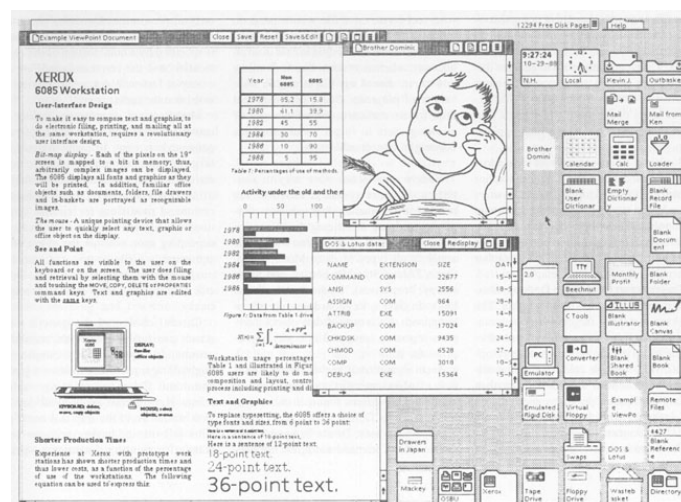
IMAI – Institut für
Mathematik und
Angewandte Informatik

1 Desktop

Xerox Star (1981)

- Schreibtisch-Metapher: vertrautes konzeptuelles Modell
- Icons: erkennen und zeigen statt erinnern und tippen
- Direkte Manipulation (Shneiderman 1983)
- WYSIWYG: what you see is what you get → PageMaker (DTP)
- Engelbarts Mouse als Zeigedepictum
- endgültige Trennung von Applikation und Interface
- Folge für Nutzer:
 - Computer zugänglicher für Nichtprogrammierer
 - Nutzer wollen immer komplexere Dinge tun
 - Selten überblickt die Nutzer alle Möglichkeiten
- Erstes System, bei dem Usability Engineering eingesetzt wird:
 - aufwendige Nutzungsanalyse und Paper Prototyping
 - Nutzertests mit potentiellen Benutzern
 - iterative Verfeinerung des Interfaces

Xerox Star Interface



Xerox Star Interface (1981)

Xerox Star (contd.)

- Kommerzieller Flop:
 - USD 15.000
 - beschränkte Funktionalität, z.B. keine Tabellenkalkulation
 - geschlossene Architektur: andere Firmen können keine Anwendungen dafür anbieten



www.thocp.net

Apple Lisa (1983)

- basiert auf den Star Interface
- etwas billiger (USD 10.000), aber auch kommerzieller Flop



www.obsoletecomputermuseum.org

Apple Macintosh (1984)

- Preis: USD 2.500
- Ideen ausgereift, Markt bereit
- ermutigt andere Anbieter, Software dafür anzubieten
- Interface Richtlinien ermöglichen Konsistenz zwischen verschiedenen Anwendungen
- exzellente Graphik und erschwinglicher Laserdrucker ermöglichen Desktop Publishing



www.at-mix.de

Posture

- Two primary types of desktop interfaces: sovereign and transient
- majority of actual work that gets done on desktop applications is done in sovereign applications
- Transients exist in supporting roles for brief, intermittent, or largely background tasks

Primary and secondary windows

- The primary window contains your application's content, typically expressed in the form of documents that can be created, edited, and shared
 - Primary windows often are divided into panes that contain content, a means of navigating between different content objects, and sets of frequently used functions for manipulating or controlling the content
 - Primary windows typically are designed to assume sovereign posture, filling most of the screen and supporting full-screen modes
- Secondary windows support the primary window, providing access to less frequently used properties and functions, typically in the form of dialogs
 - If your application allows panes located in the primary window to be detached and manipulated separately, these floating panels or palettes also take on a role as secondary windows

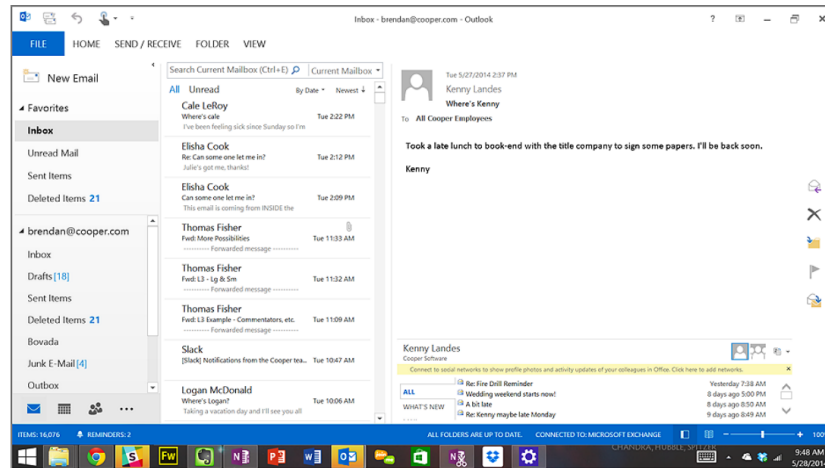
Primary window structure

- **Menus and toolbars** are collections of related actions the user can instruct the application to perform, such as "close this document" or "invert the colors of the current selection."
- **Content panes** form the primary work area within most desktop applications, whether it is the editable view of a form or document or (as in the case of a software music synthesizer, for example) a complex control panel
- **Index panes** provide navigation and access to documents or objects that ultimately appear in the content view(s) for editing or configuration
- **Tool palettes** allow the user to rapidly switch between the application's modes of operation by selecting one tool from a set of tools
- **Sidebars** most often allow object or document properties to be manipulated without the need to resort to modal or modeless dialogs
- Question: What are ribbons?

Windows on the Desktop

- Overlapping windows
- Tiled windows
- Virtual desktop spaces
- Full-screen applications
- Multipaned applications

Multipaned



(Cooper et al., 2014)

Window states

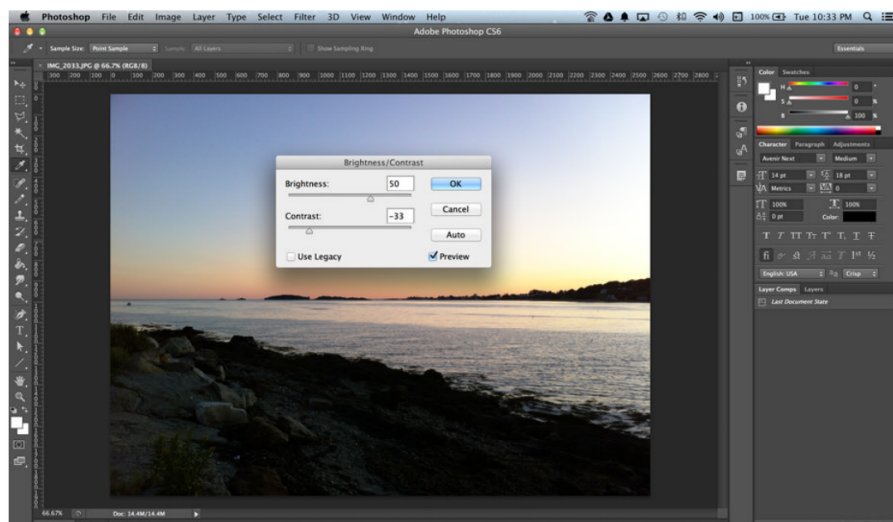
- Minimized windows get collapsed into icons on the desktop or into the taskbar (Windows) or the Dock (OS X)
- Maximized windows fill the entire screen, covering whatever is beneath them
- The pluralized state is that in-between condition where the window is neither an icon nor maximized to cover the entire screen

MDI vs SDI

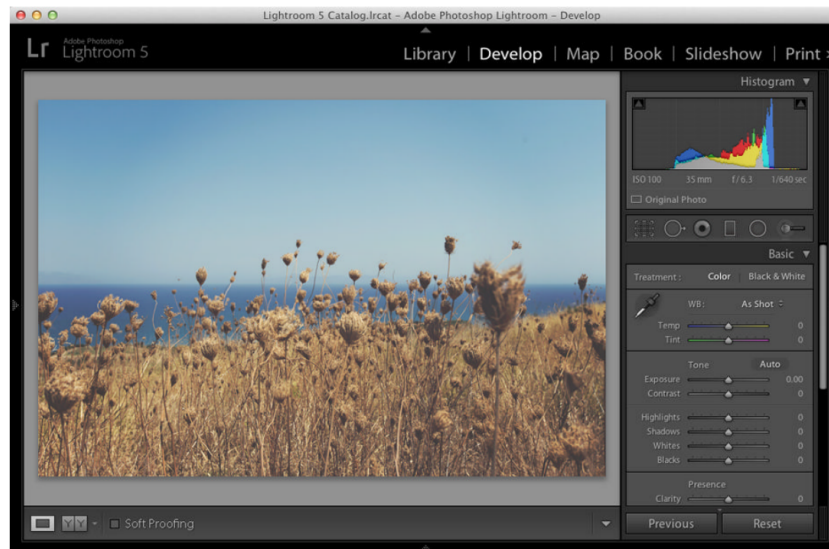
- Multiple document interface, or MDI.
 - multiple windows reside under a single parent window.
- Tabbed document interface, or TDI
 - allows multiple documents or panels to be contained within a single window.
- Single document interface, or SDI.
 - all windows are independent of each other.

Unnecessary rooms

- Secondary windows containing functions that should really be integrated into panes or other surfaces within the primary window.



(Cooper et al., 2014)



(Cooper et al., 2014)

Necessary rooms

- When users perform a function outside their normal sequence of events, it's usually desirable to provide a special place in which to perform it.
- For example, purging a database is not a normal activity.
 - It involves setting up and using features and facilities that are not part of the normal operation of the database application

Menu

- Drop-down, pop-up
- Toolbars and direct-manipulation idioms can be too inscrutable for a first-time user to understand, but the textual nature of the menus explains the functions
- For an infrequent user who is somewhat familiar with an application, the menu's main task is as an index to known tools: a place to look when he knows there is a function but he can't remember where it is or what it's called.
- For a frequent user, menus provide a stable physical location at which to access one of hundreds of possible commands, or a quick reminder about keyboard shortcuts.

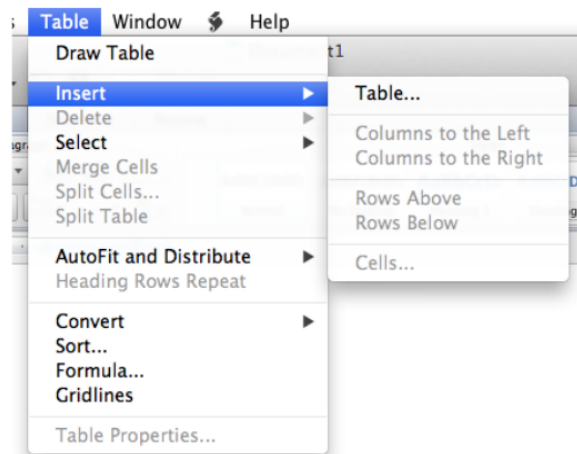
Menus II

- Disabled menu items
- Check mark menu items
- Icons on menus

Accelerators and Mnemonics

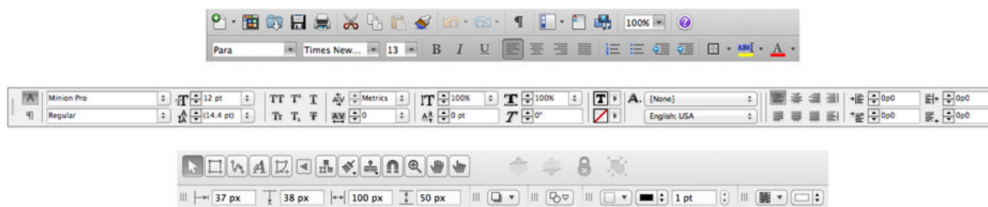
- Accelerators or keyboard shortcuts provide an easy way to invoke functions from the keyboard.
 - These are commonly function keys (such as F9) or combinations involving modifier keys (Ctrl, Alt, Option, and Command).
- Access keys or mnemonics are another Windows standard (they are also seen in some UNIX GUIs) for adding keystroke commands in parallel to the direct manipulation of menus and dialogs
 - Mnemonics are accessed using the Alt key, arrow keys, and the underlined letter in a menu item or title.

Cascading Menus



(Cooper et al., 2014)

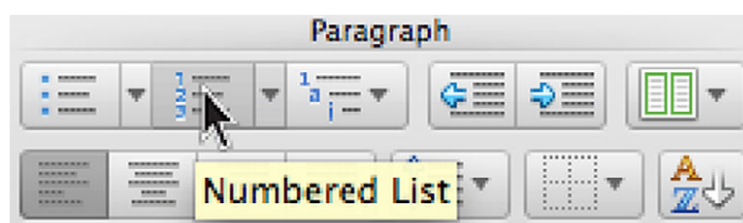
Toolbars, Palettes, and Sidebars



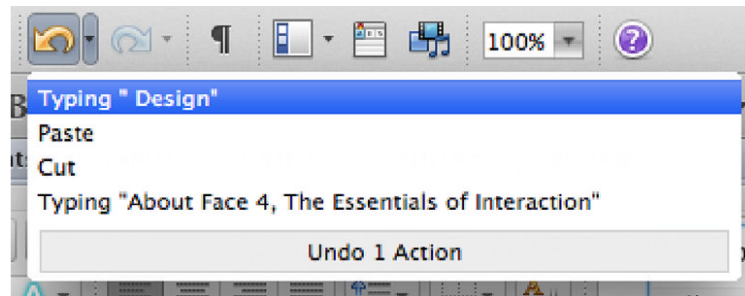
(Cooper et al., 2014)

Toolbars and menus

- Toolbars work together with menus to satisfy user needs as they mature
- Whereas menus are complete toolsets with the main purpose of teaching inexperienced users and organizing seldom-used advanced functions, toolbars are for frequently used commands and cater to perpetual intermediates
- They complement each other perfectly, addressing different user needs at different times
- Toolbars are modeless, but they don't introduce the conundrums that modeless dialogs do
- Toolbar button, or icon button
- ToolTips

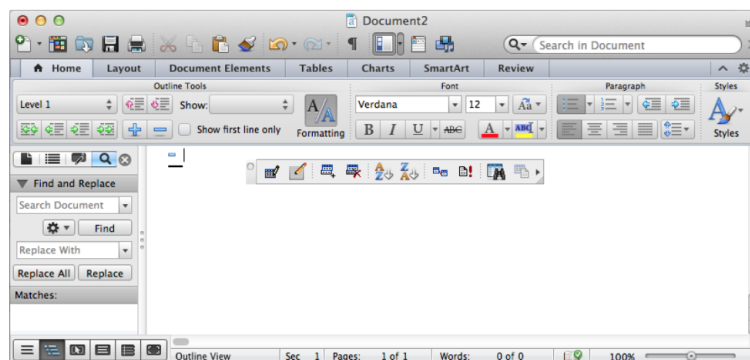


(Cooper et al., 2014)



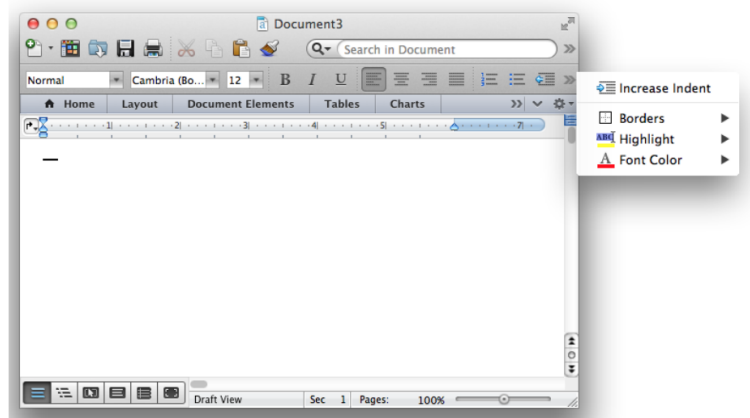
(Cooper et al., 2014)

Movable and Overflow toolbars



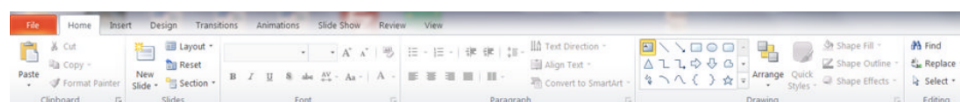
(Cooper et al., 2014)

Movable and Overflow toolbars



(Cooper et al., 2014)

Ribbons



(Cooper et al., 2014)

Docked Palettes



(Cooper et al., 2014)

Sidebars, task panes, and drawers

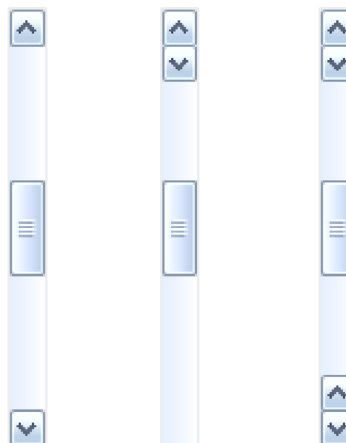


(Cooper et al., 2014)

Pointing

- The mouse is a “relative” pointing device:
- Tablets and slates are usually “absolute” pointing devices
- Touchscreens on Laptops/Desktops?

Scrollbars



(Cooper et al., 2014)

Mouse buttons and controls

- In general, the left mouse button is used for all the primary direct-manipulation functions, such as triggering controls, making selections, drawing, and so on
- The right mouse button enables direct access to properties and other context-specific actions on objects and functions via the ubiquitous context menu
- Rolling the wheel forward scrolls the window up, and rolling it backwards scrolls the window down, Pressing it acts like a third mouse button
- Using modifier keys in conjunction with the mouse can extend direct-manipulation idioms
- Metakeys include Ctrl, Alt, Command (on Apple computers), and Shift

Point & Click

- Clicking and dragging
 - selecting, reshaping, repositioning, drawing, and dragging and dropping
- Double-clicking
 - Double-clicking means single-clicking plus action
- Chord-clicking
 - ???
- Double-clicking and dragging
 - ???

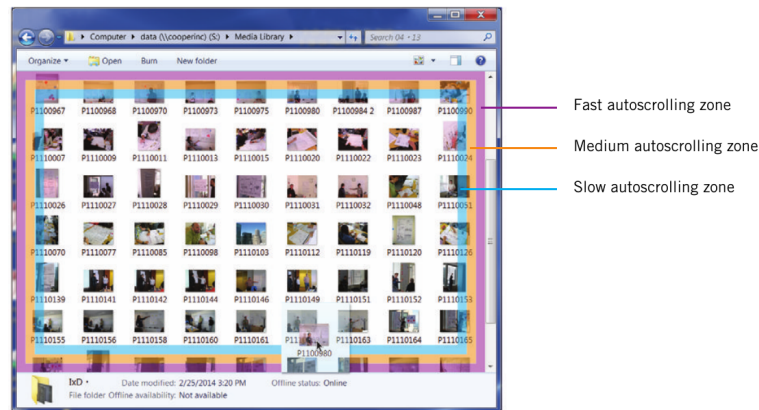
Selection

- Command ordering and selection
- Verb-object ordering is consistent with how commands are formed in English
 - “rm file” in Unix
 - Problem: termination
- With an object-verb command order, we don’t need to worry about termination
 - Select object, do something

Drag and drop

- drag-and-drop operation: clicking and holding the button while moving an object across the screen and releasing it in a meaningful location
- Surprisingly, drag and drop isn’t used as widely as we’d like to think, and it certainly hasn’t lived up to its full potential.
- Drop candidates must visually indicate their receptivity.
- The drag cursor must visually identify the source object.

Autoscoll



(Cooper et al., 2014)

2D object manipulation

- Repositioning
 - demands the click-and-drag action, making it unavailable for other purposes
- Resizing and reshaping
 - resize handles
- Connecting
 - clicks and drags from one object to another

2D object manipulation

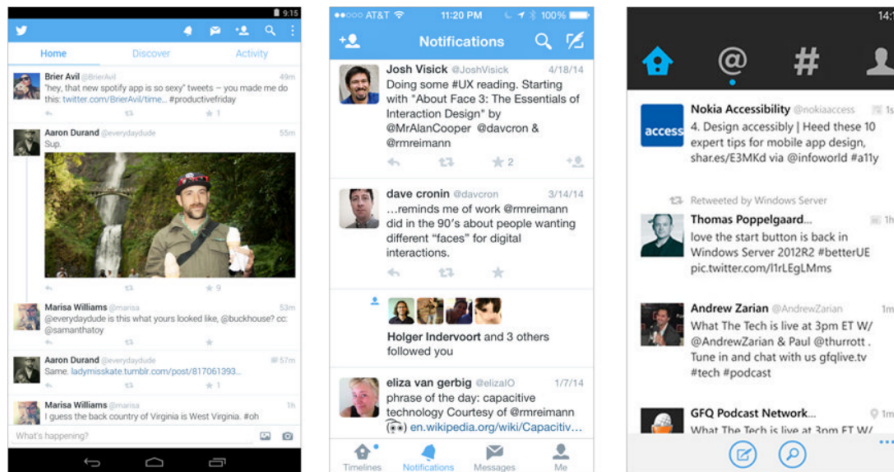
- Display issues and idioms
 - Multiple viewpoints
 - Baseline grids, depthcueing, shadows, and poles
 - Wireframes and bounding boxes
- Drag thresholds
- The picking problem
- Object rotation, camera movement, rotation, and zoom

2 Handheld

Anatomy of a Mobile App

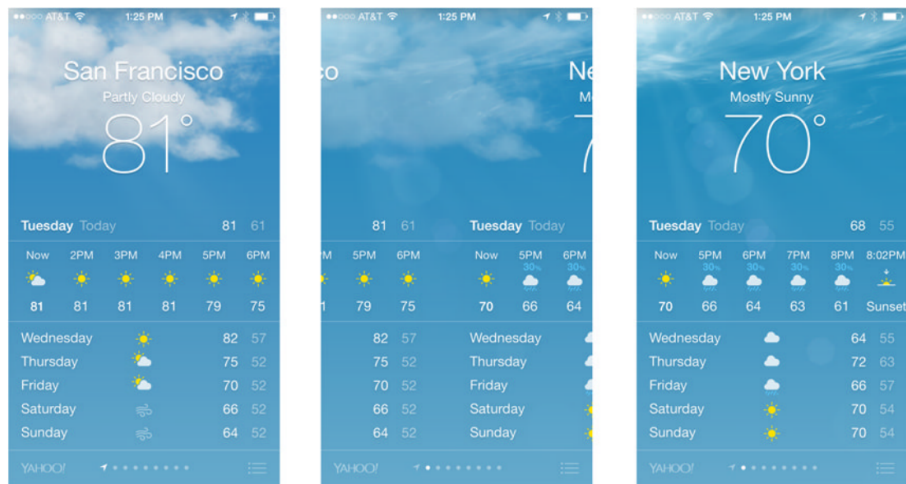
- While the posture of desktop applications is most often sovereign, mobile apps, by contrast, are by their very nature transient
- context-driven nature of the majority of mobile apps (games perhaps being the exception, but the interaction design of games in general is a unique topic in itself) dictates a transient stance, especially on handheld mobile devices
- The fact that these transient apps take up their host device's entire screen makes them no less transient

2.1 Mobile Stacks



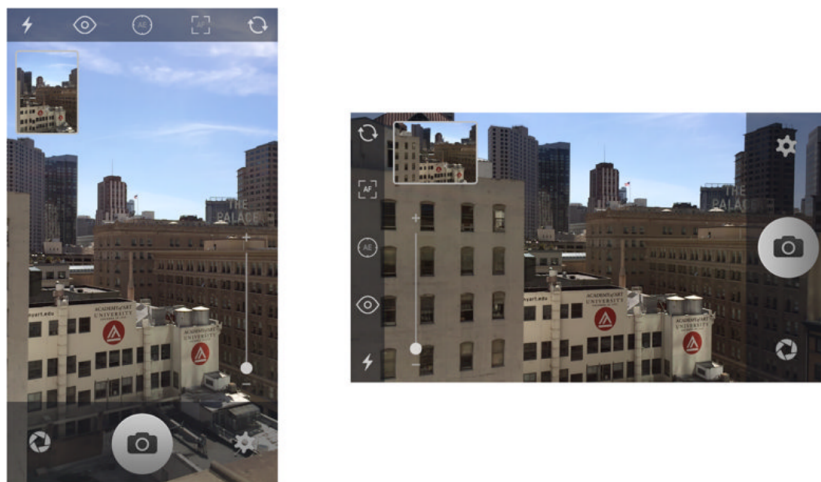
(Cooper et al., 2014)

Carousel



(Cooper et al., 2014)

Orientation and layout



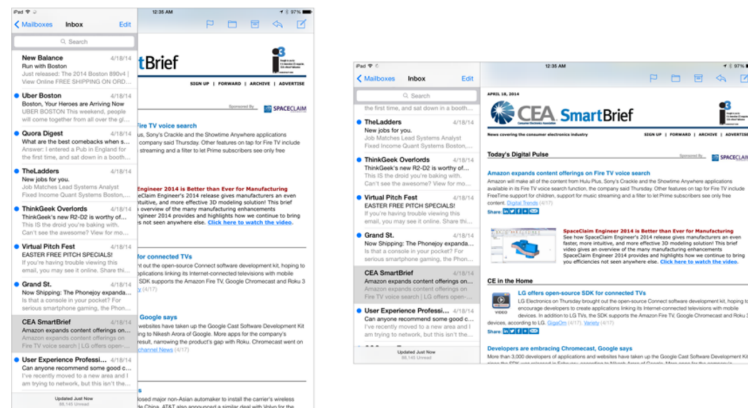
(Cooper et al., 2014)

2.2 Tablet

Tablet format apps

- Tablet format apps have considerably more breathing room than handheld-format apps as far as screen real estate is concerned
- The iPad's 4:3 aspect ratio and large screen size ensures plenty of room for navigational and functional controls, but Windows and Android tablets also manage quite serviceably with the movie-like 16:9 aspect ratios

Stacks and index panes



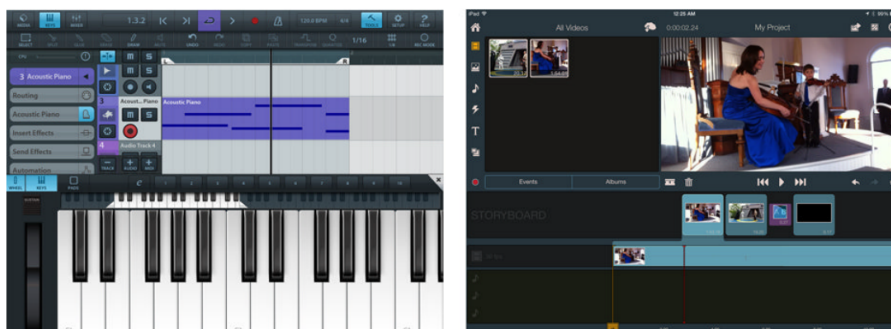
(Cooper et al., 2014)

Pop-up control panels



(Cooper et al., 2014)

Mobile versus desktop-like layout



(Cooper et al., 2014)

Hardware-like control layout



(Cooper et al., 2014)

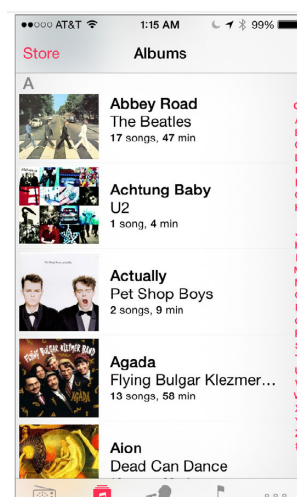
2.3 Mini-tablet

Mini-tablet format apps

- Adjacent panes
 - Generally not a good idea on full-sized tablets in portrait orientation, adjacent panes are usually far too cramped to consider on mini-tablets
 - In landscape, at most two adjacent panes can be supported
- Tool bars
 - In portrait view, these can feel distant from the action due to the tall, narrow form factor and increased screen size over handhelds
 - In landscape orientation, tool bars stacked with navigation bars leave little vertical space for content
- Lists
 - Single-column lists tend to look out of proportion on mini-tablets, even in portrait
 - Grid, swimlane, and card approaches tend to work better
- Pop-up versus full-screen dialogs
 - Mini-tablets are big enough that using phone-style full-screen idioms for menus and dialogs won't work; these should be implemented as pop-up dialogs

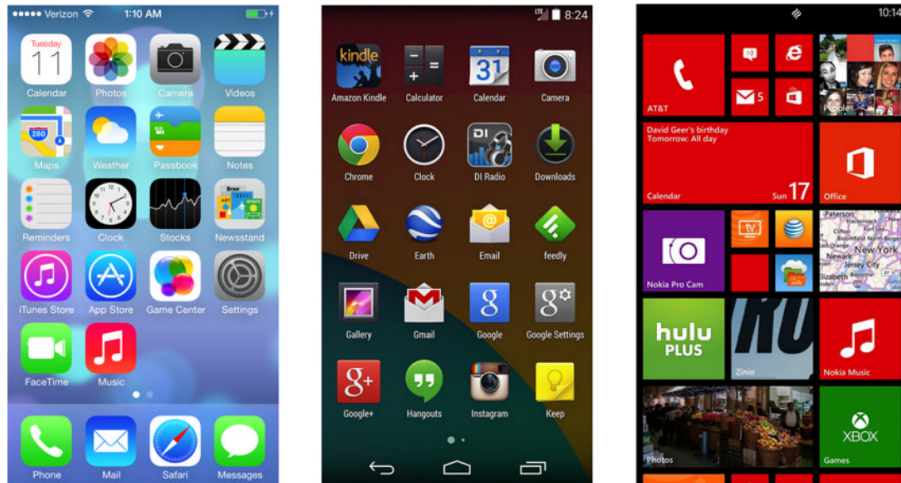
2.4 Navigation, Content, Control

Lists



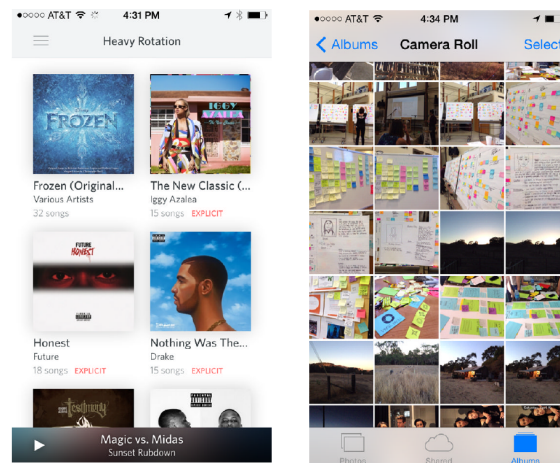
(Cooper et al., 2014)

Grids



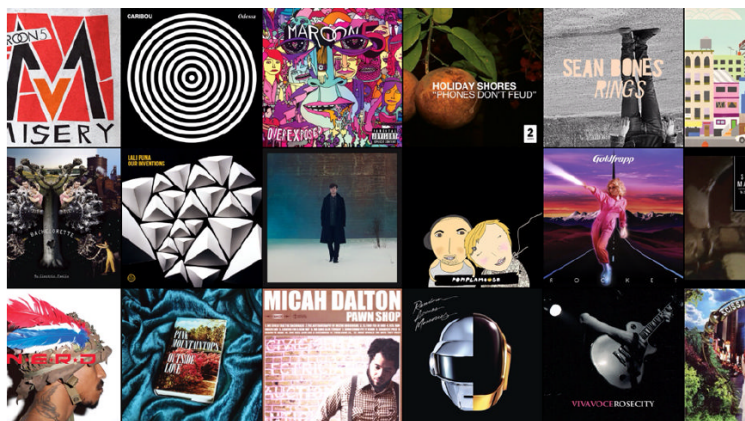
(Cooper et al., 2014)

Grids



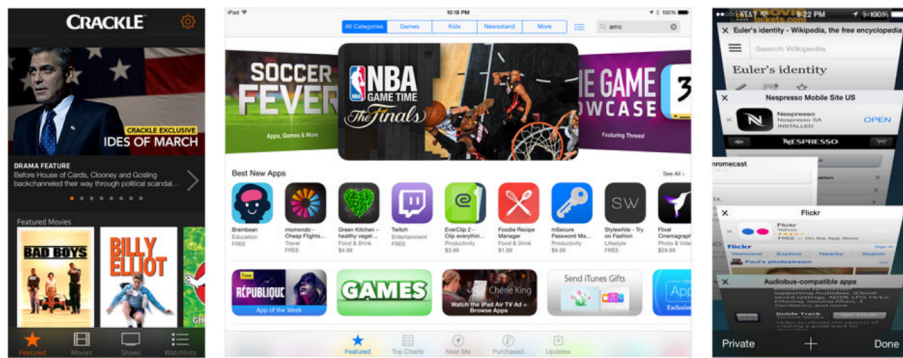
(Cooper et al., 2014)

Grids



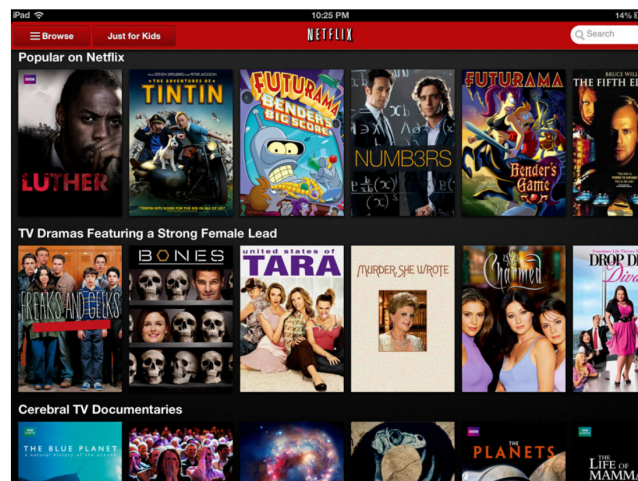
(Cooper et al., 2014)

Content carousels



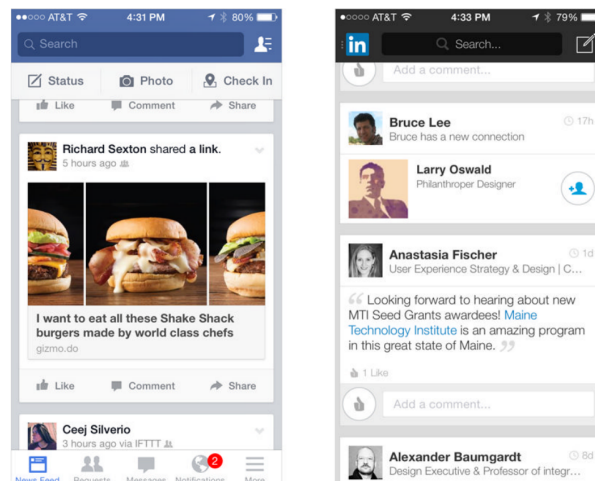
(Cooper et al., 2014)

Swimlanes



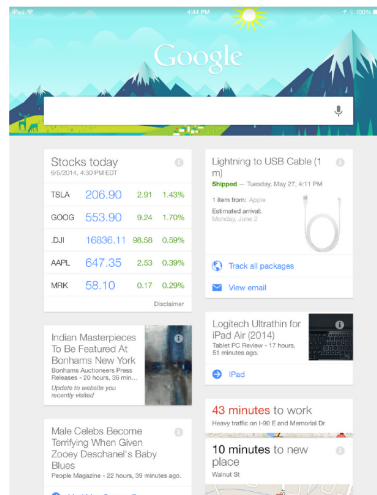
(Cooper et al., 2014)

Cards



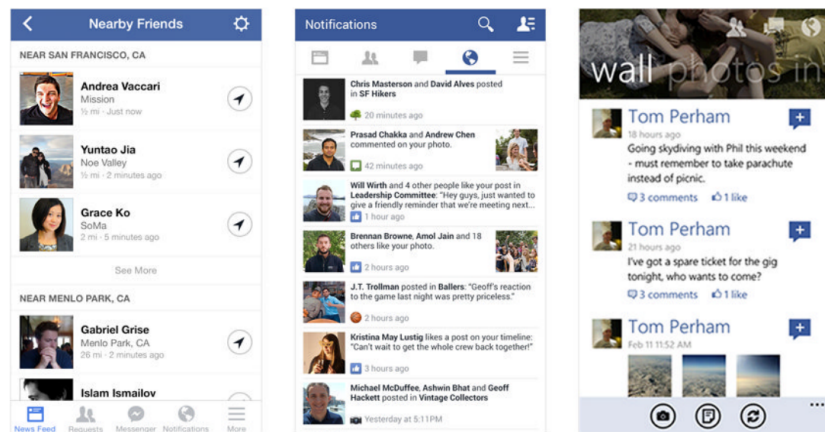
(Cooper et al., 2014)

Cards



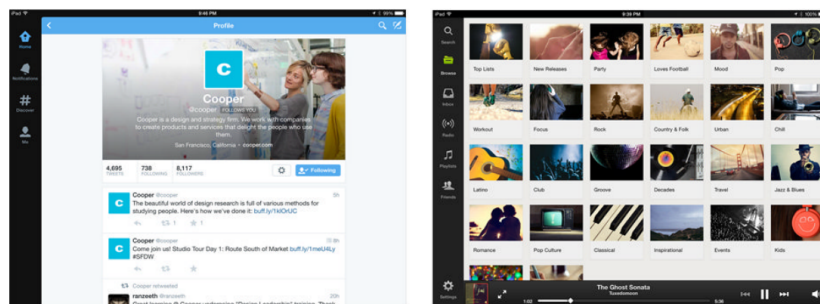
(Cooper et al., 2014)

Tab Bars



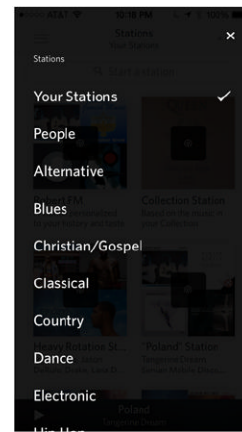
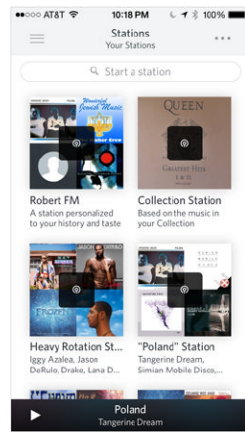
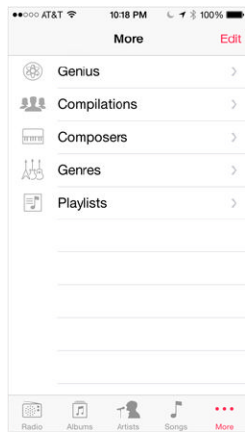
(Cooper et al., 2014)

Tab Bars



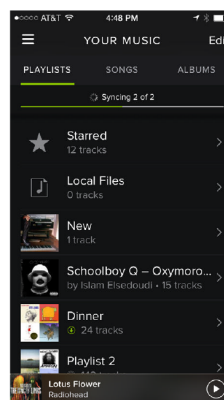
(Cooper et al., 2014)

More ... Controls



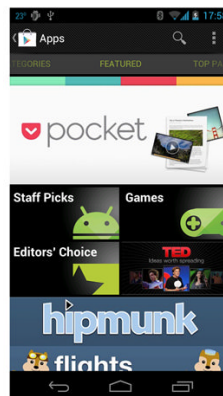
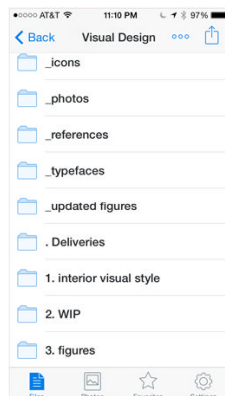
(Cooper et al., 2014)

Tab Carousel



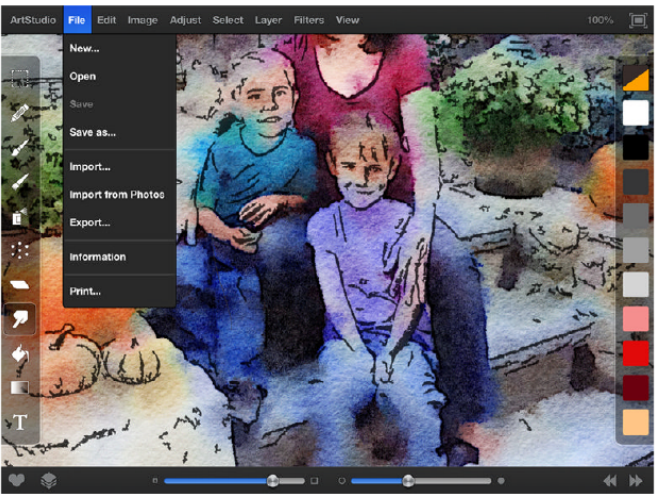
(Cooper et al., 2014)

Nav Bars & Action Bars



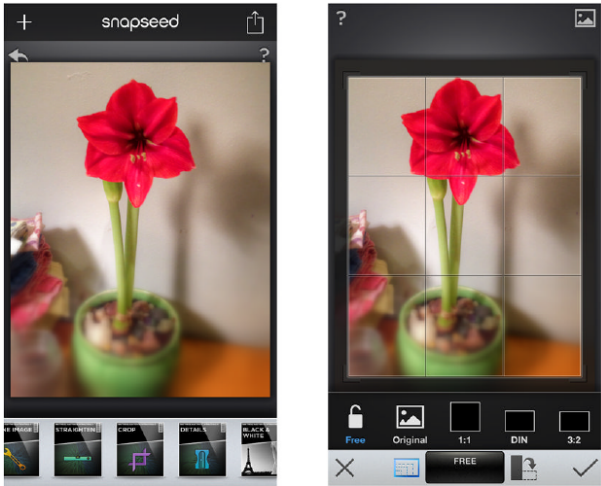
(Cooper et al., 2014)

Vertical tool bars and palettes



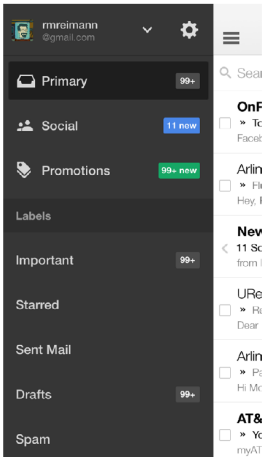
(Cooper et al., 2014)

Tool carousels



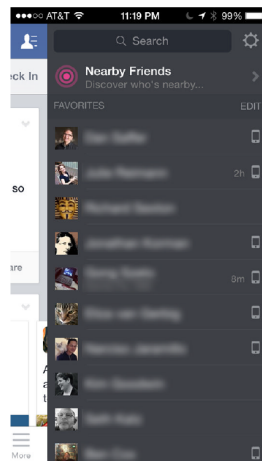
(Cooper et al., 2014)

Drawers



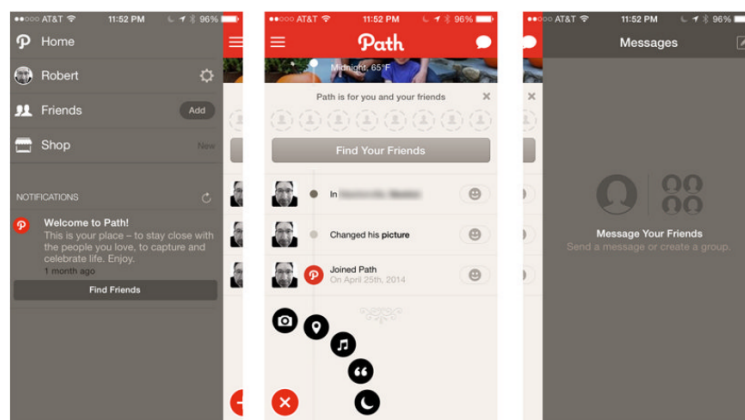
(Cooper et al., 2014)

Drawers



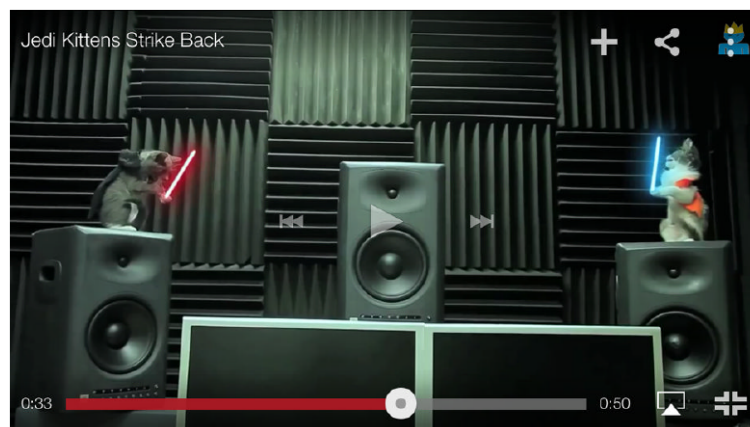
(Cooper et al., 2014)

Drawers



(Cooper et al., 2014)

Tap-to-reveal



(Cooper et al., 2014)

Direct Manipulation



(Cooper et al., 2014)

References

Literatur

Cooper, A., Reimann, R., Cronin, D., and Noessel, C. (2014). *About Face (fourth edition): the essentials of interaction design*. John Wiley & Sons.