Human Computer Interaction

Metaphors, Idioms, and Visual Interaction Design

Prof. Andrew D. Bagdanov

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze
andrew.bagdanov AT unifi.it

December 25, 2017

Outline

Overview

Overview

- Today we will see how metaphors, once thought to be the cornerstone of good interaction design, are actually not a good basis for building complex user interfaces.
- We will see how idioms are a better basis for user interfaces, since we are particularly adept at acquiring and memorizing them.
- Then we will discuss some of the building blocks of visual interface design.
- Finally, we will see some of the basic principles of visual organization for interaction design.

News

Upcoming lectures

Today	Excise, metaphors, and visual interaction design
Tomorrow	Kivy on Android (the easy way) lab
Friday	NO LESSON

Latest and Greatest

- On Friday, December 15th, we will have our final Latest and Greatest:

  Kostakos, Vassilis. "The big hole in HCI research." Interactions 22.2, 2015.

Metaphors, Idioms, and Affordances

Metaphors

- Sometimes we speak of finding the right metaphor for an interface.
- But, there is risk in using real-world objects and assuming it will aid users in learning.
- Strict adherence to metaphors ties interfaces unnecessarily to the workings of the physical world.
- An advantage of the digital world is that the model presented to users need not be bound by the messiness of real three-dimensional space.
- User interfaces based on metaphors have an array of other problems: there aren't enough good metaphors, they don't scale well, and the ability of users to recognize them is questionable, especially across cultural boundaries.

Interface paradigms

- There are three dominant paradigms in the conceptual and visual design of user interfaces:
  - implementation-centric: based on understanding how things actually work "under the hood" (difficult).
  - metaphoric: based on intuiting how things work (risky).
  - idiomatic: based on learning how things work (a natural human process).
- HCI has advanced from focusing on technology (implementation) to focus on metaphor.
- Metaphors are great tools for communication, but they are weak tools for the design of software.

Implementation-centric interfaces

- Implementation-centric user interfaces are widespread in industry, they are expressed in terms of their construction, of how they are built.
- To use them, users must understand how the software works internally.
- This is a more subtle concept than it seems at first:
  - Most programs today are implementation-centric in that they show us precisely how they are built.
  - There is one button per function, one dialog per module of code, and the commands and processes precisely echo the internal data structures and algorithms.
- The problem is that we must learn how the program works in order to successfully use the interface.

Implementation-centric interfaces

- Implementation-centric interfaces are the easiest to build: every time a programmer writes a function, we put some user interface to test that function.
- Engineers may want to understand the inner workings, but most users don't have either the time or desire.
- Most users would much rather be successful than be knowledgeable.
- Similar to the implementation-centric interface is the org-chart centric interface: where a product or website is organized not according to how users think about information, but by how the company the site represents is structured.

Metaphoric interfaces

- Metaphoric interfaces rely on intuitive connections that users make between the visual cues in an interface and its function.
- There is no need to understand the mechanics of the software, so it is a step forward from implementation-centric interfaces.
- Metaphors in the context of user interface and interaction design really mean visual metaphors: a picture used to represent the purpose or attributes of a thing.
- Metaphors can range from the tiny images (scissors) on toolbar buttons to the entire screen on some programs (full-size checkbook).
- We understand metaphors intuitively, but what does that really mean?

Metaphoric interfaces

- Here is a definition:

  in · tu · i · tion in-tu-'wi-shen n 1 : quick and ready insight 2a : immediate apprehension or cognition 2b : knowledge or conviction gained by intuition 2c : the power or faculty of attaining to direct knowledge or cognition without evident rational thought and inference

- Intuition works by inferring relationships between things, while not being distracted by their differences.
- We grasp the meaning of the metaphoric controls because we mentally connect them with things we have already learned.
- But, this depends on the idiosyncratic human minds of users, which may not have the requisite inferential power necessary to make those connections.

Metaphoric interfaces: baby steps

- The Windows XP desktop is a good example of a mixed implementation- and metaphor-based interface:

Limitations of metaphors

- The idea that metaphors are a firm foundation for user-interface design is misleading.
- Metaphors don't scale very well:
  - A metaphor that works well for a simple process in a simple program will often fail as that process grows in size or complexity.
  - Large file icons were a good idea when computers had floppies or 10 MB hard disks with only a couple of hundred files.
- Metaphors rely on associations by both the designer and user:
  - If the user doesn't have the same cultural background as the designer, it is easy for metaphors to fail.
  - Even in the same culture there can be misunderstanding: does a picture of an airplane mean "check flight arrival information" or "make airline reservations?"

Limitations of metaphors

- Though metaphors offer a small boost in learnability to first-time users, they soon become tiresome.
- Relying on metaphors implies that rational thought is not required in the process of intuiting.
- In the computer industry, the word intuitive is often used to mean easy-to-use or easy-to-understand.
- Ease-of-use is obviously important, but it doesn't promote the craft of HCI to attribute its success to metaphysics.

Finding good metaphors

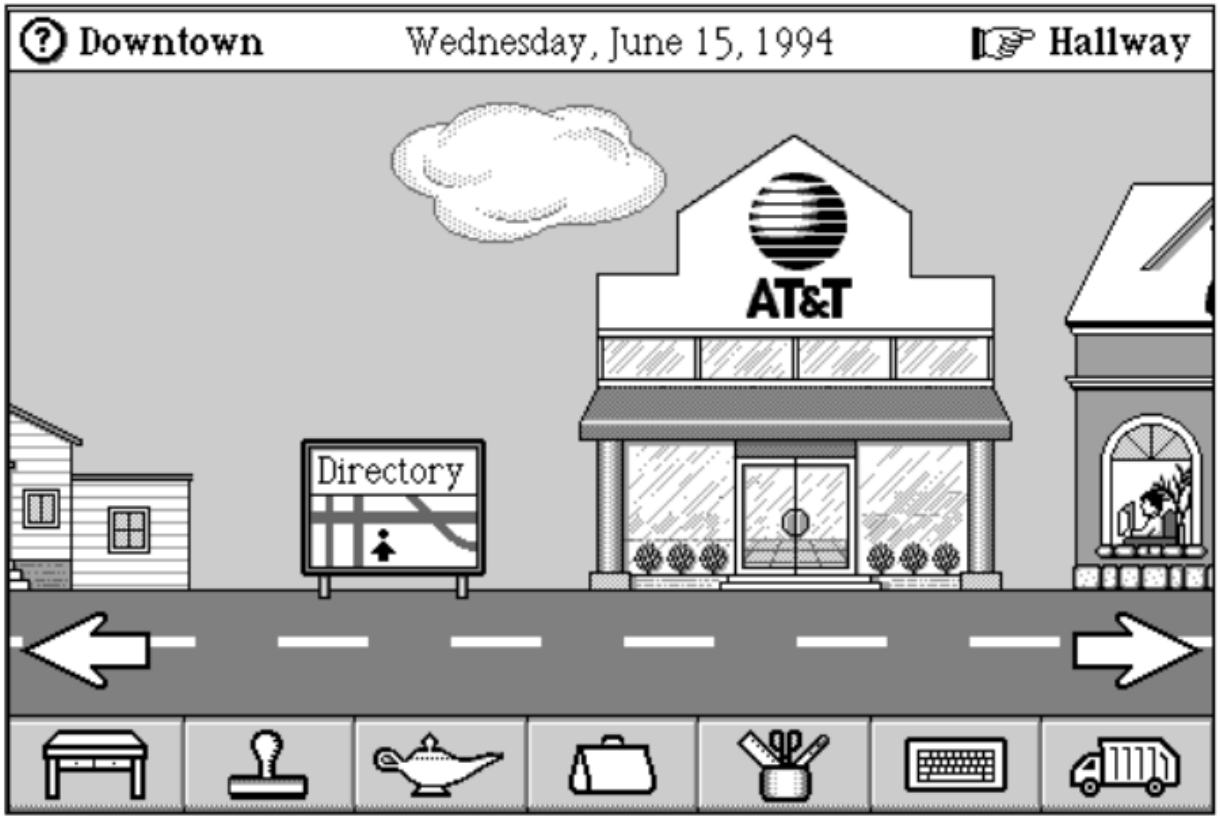
- It may be easy to discover visual metaphors for physical objects like printers and documents.
- It can be difficult or impossible to find metaphors for processes, relationships, services, and transformations – the most frequent uses of software.
- Computers are so powerful because of their ability to manage incredibly complex relationships within very large sets of data.
- Their very utility is based upon the fact that the human mind is challenged by such multidimensional problems.
- Almost by definition, these processes are not well suited to a simple, physical analog that people automatically comprehend.

Problems with global metaphors

- The most significant problem with metaphors is that they tie our interfaces to Mechanical Age artifacts.
- An extreme example of this was Magic Cap, a handheld communicator interface introduced with some fanfare by General Magic in the 1990s.
- It relies on metaphors for almost every aspect of its interface:
  - You access your messages from an inbox or a notebook on a desk.
  - You walk down a hallway that is lined with doors representing secondary functions.
  - You go outside to access third-party services, which are represented by buildings on a street.
  - You enter a building to configure a service, and so on.
- This reliance on this metaphor means that you can intuit the basic functioning of the software
- The downside is that after you understand its function, the metaphor adds significantly to the overhead of navigation.

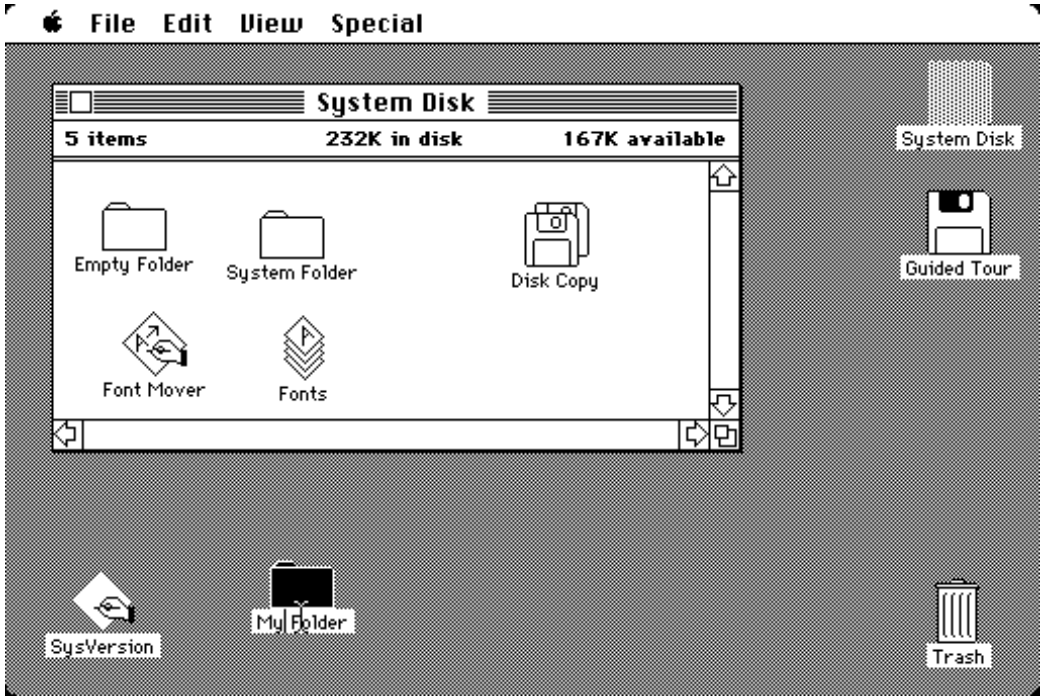
Problems with global metaphors

- General Magic's interface relies on what is called a global metaphor.
- This is a single, overarching metaphor that provides a framework for all the other metaphors in the system.
- The desktop of the original Macintosh is also a global metaphor.



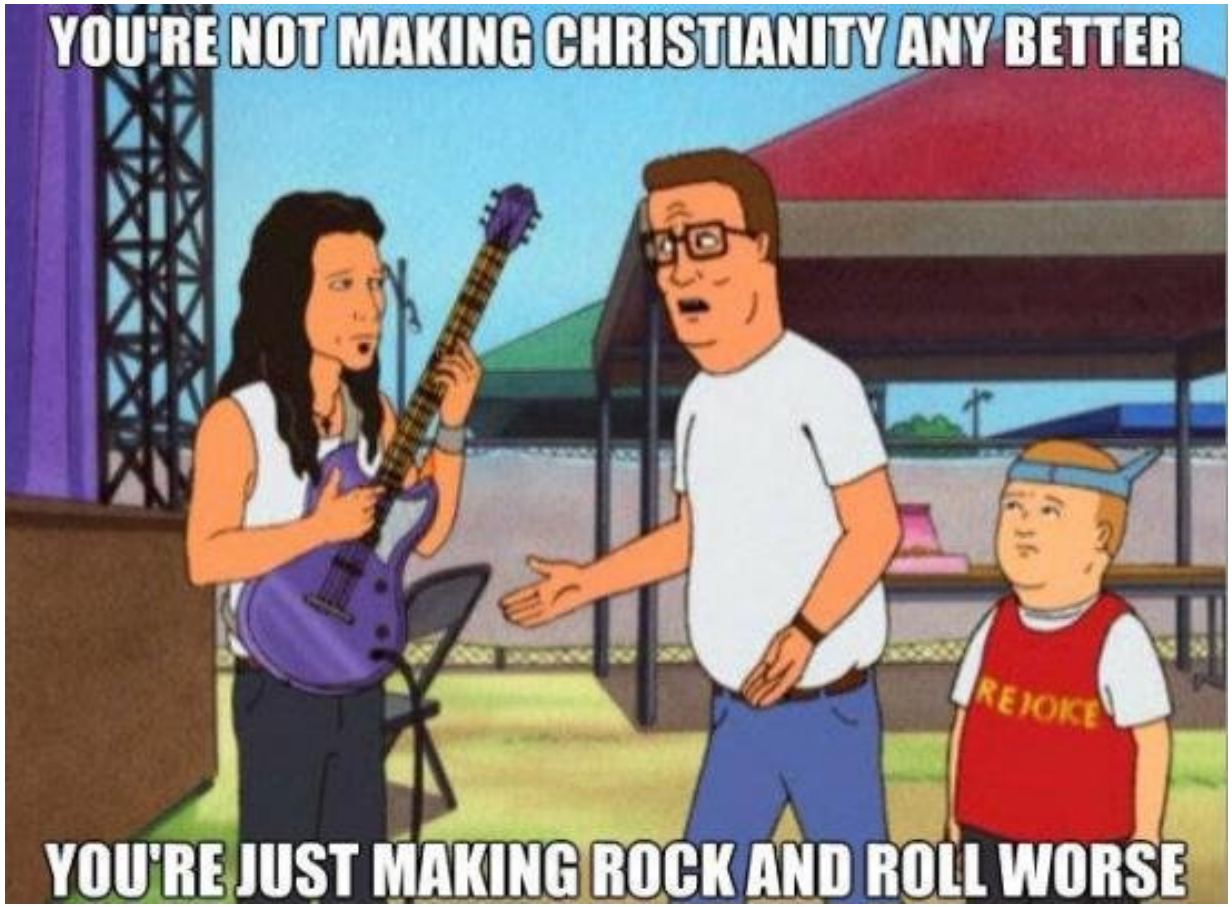
Problems with metaphors

- Using metaphors in HCI is a vicious circle:
  - after the user depends on the metaphor for recognition, he expects consistency of behavior with the real-world object;
  - and thus the designer is tempted to render more (or all) aspects of the software in terms of the metaphor's Mechanical Age referent.
- This cheapens both the metaphor and the interface.



Problems with metaphors

- Transliterating mechanical processes onto the computer usually makes them worse than they were before.
- Like Hank Hill's observation on Christian Rock music:



Problems with metaphors

- Take the example of the ubiquitous file folder in modern computer operating systems.
- It is easy to learn and understand because of its similarity to a physical file folder.
- However, it functions differently than real file folders, which can create cognitive friction on the part of users.
- No one nests physical folders 10-layers deep, which makes it difficult for novice computer users to intuit the navigational structures of an operating system.
- Use metaphors if you can find them, but don't bend your interface to fit some arbitrary metaphoric standard.

Idiomatic interfaces

- Idiomatic design is based on the way we learn and use idioms – figures of speech like “beat around the bush” or “cool.”
- Idiomatic user interfaces focus not on technical knowledge or intuition of function, but rather on the learning of simple, non-metaphorical visual and behavioral idioms to accomplish goals and tasks.
- Idiomatic expressions don’t provoke associative connections the way that metaphors do – something can be both cool and hot and be equally desirable.
- We understand the idiom because we have learned it, not because we understand it or because it makes subliminal connections in our minds.

Idiomatic interfaces

- The human mind has the capacity to remember large numbers of idioms without relying on comparisons to known situations.
- Most of the elements of intuitive graphical interfaces are actually visual idioms.
- Windows, title bars, close boxes, screen-splitters, hyperlinks, and drop-downs are things we learn idiomatically rather than intuit metaphorically.
- The use of the trashcan to unmount disks before removing it is accidentally idiomatic – despite the visual metaphor of the trash can itself.
- There is nothing about the physical appearance of the mouse that indicates its purpose or use, so learning it is not intuitive.
- However, learning to point at things with a mouse is incredibly easy.

Idiomatic interfaces

- All idioms must be learned, good idioms must be learned only once.
- We think that learning interfaces is hard because of experience with implementation-centric software.
- These interfaces are hard to learn because you need to understand how the software works internally to use them effectively.
- Most of what we know we learn without understanding: things like faces, social interactions, attitudes, melodies, brand names, the arrangement of rooms.
- We recognize these things because we have looked at them and automatically (and easily) memorized them.

Building idioms

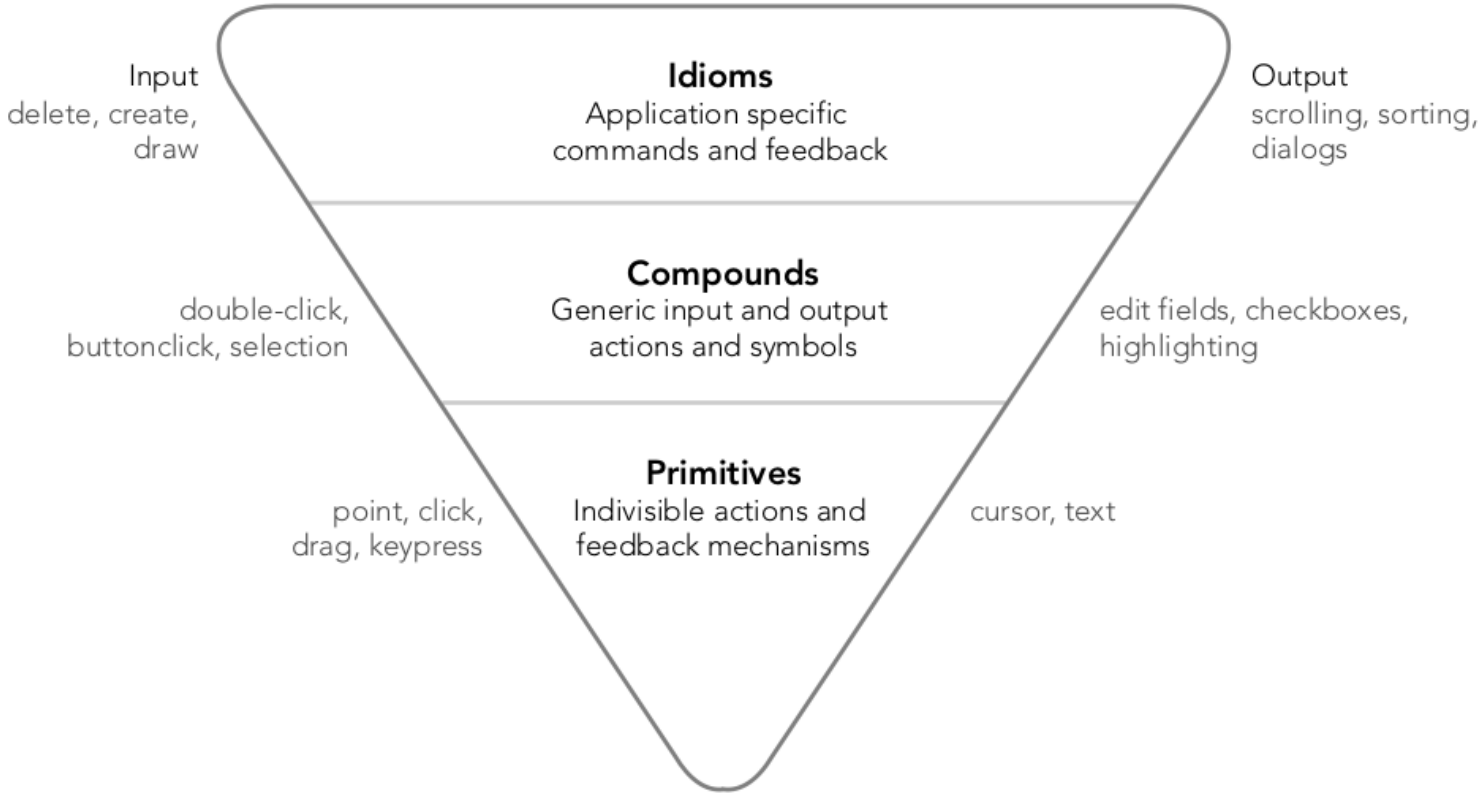
- When graphical user interfaces were first invented, they were so clearly superior that many observers credited the success to their graphical nature.
- This was a natural, but incorrect, assumption:
  - the first GUIs were better because the graphical nature of their interfaces required a restriction of the range of interaction vocabulary.
  - the input they could accept from the user went from an unrestricted command line to a tightly restricted set of mouse-based actions.
- Using the buttons on the mouse, users could click, double-click, or click and drag.
- The number of atomic elements in user input vocabulary dropped from hundreds to just three.

Building idioms: interaction vocabulary

- The more atomic elements there are in an interaction vocabulary, the more time-consuming and difficult the learning process is.
- But restricting our interaction vocabulary reduces its expressiveness at the atomic level.
- However, more complex interactions can be built from the atomic ones.
- A properly formed interaction vocabulary can be represented by an inverted pyramid.

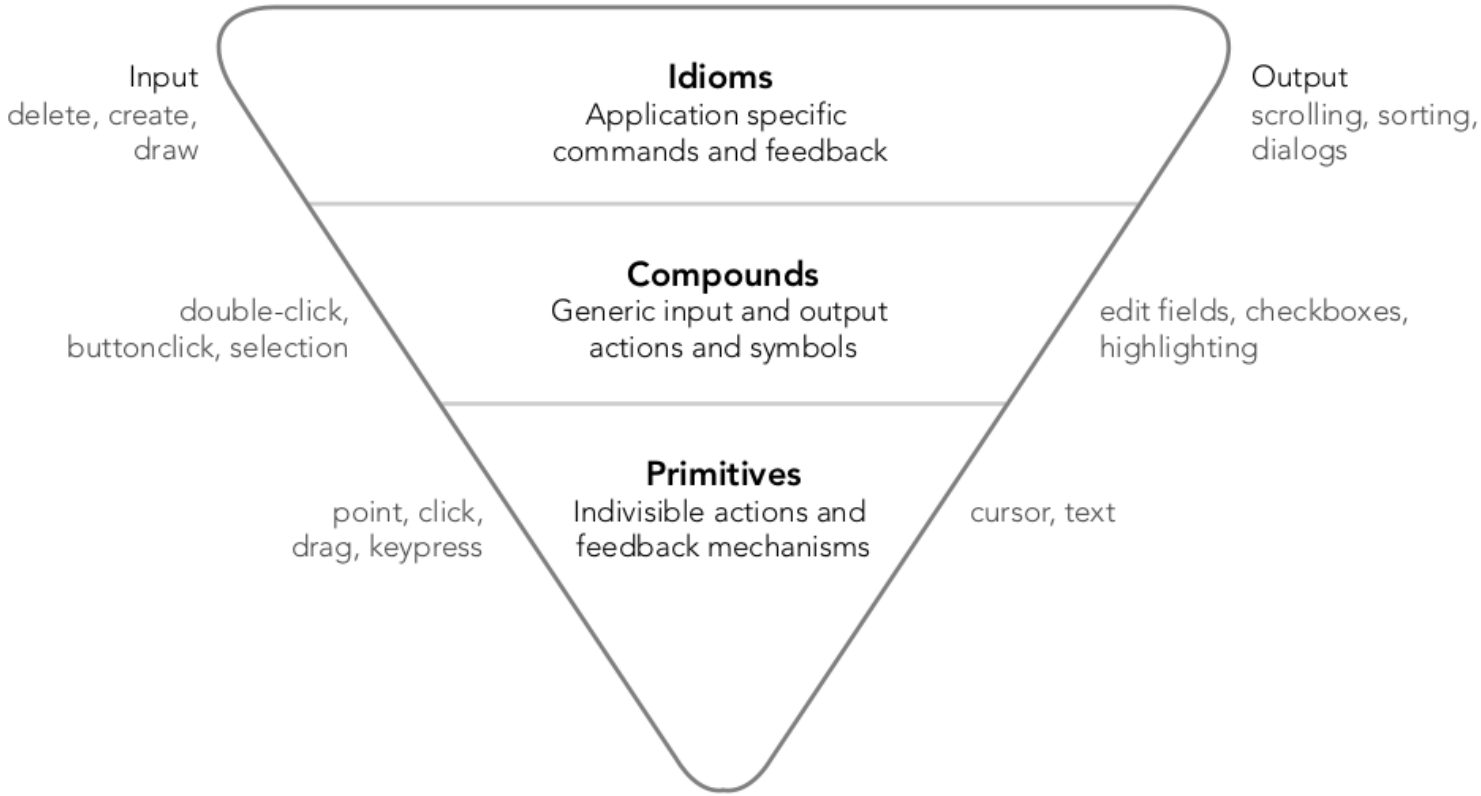
Building idioms: interaction vocabulary

- The bottom layer contains primitives, the atomic elements of which everything in the language is composed.
- In modern GUIs, these primitives consist of pointing, clicking, and dragging.



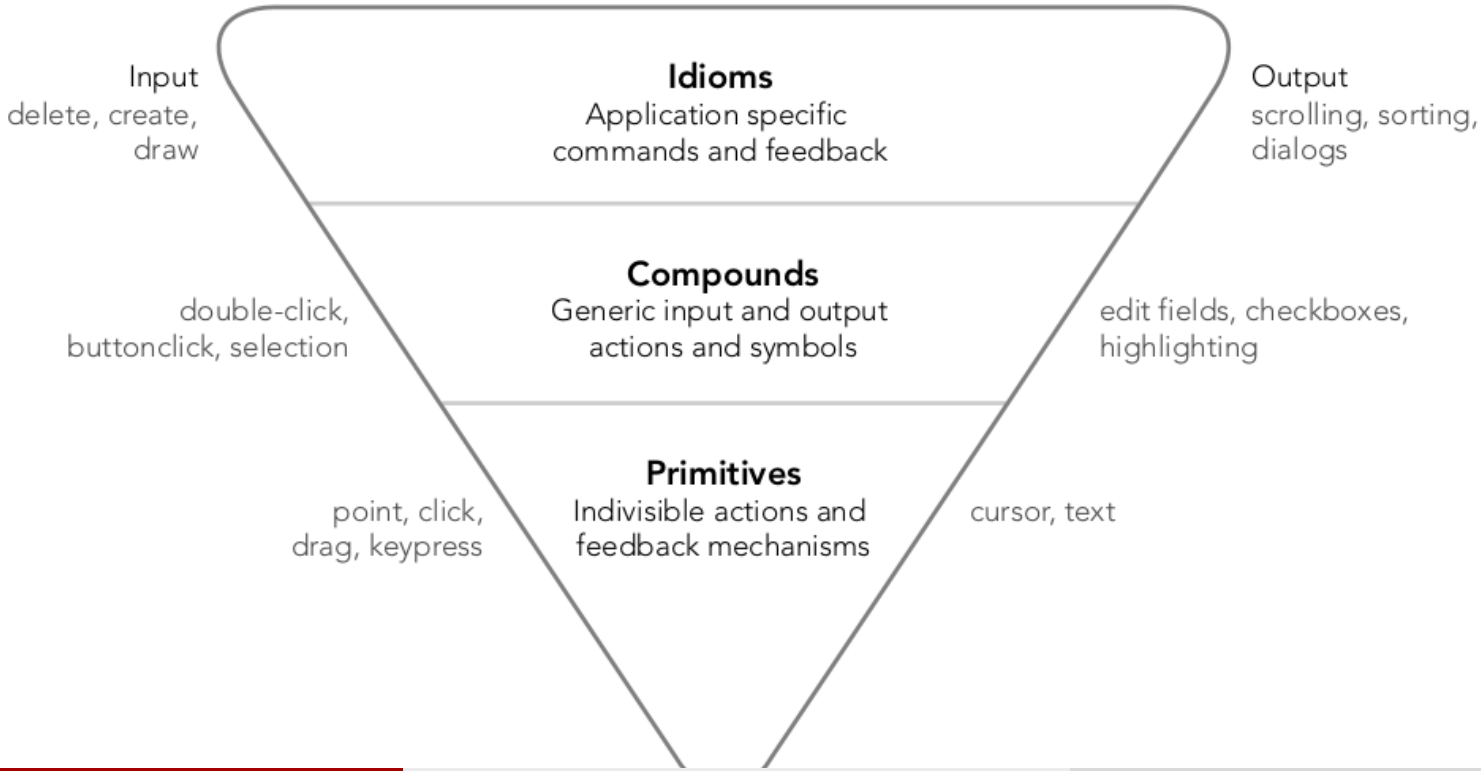
Building idioms: interaction vocabulary

- The middle layer contains compounds created by combining one or more of the primitives.
- These include visual objects such as text display, actions such as double-clicking or clicking-and-dragging, and manipulable objects like pushbuttons, check boxes, etc.



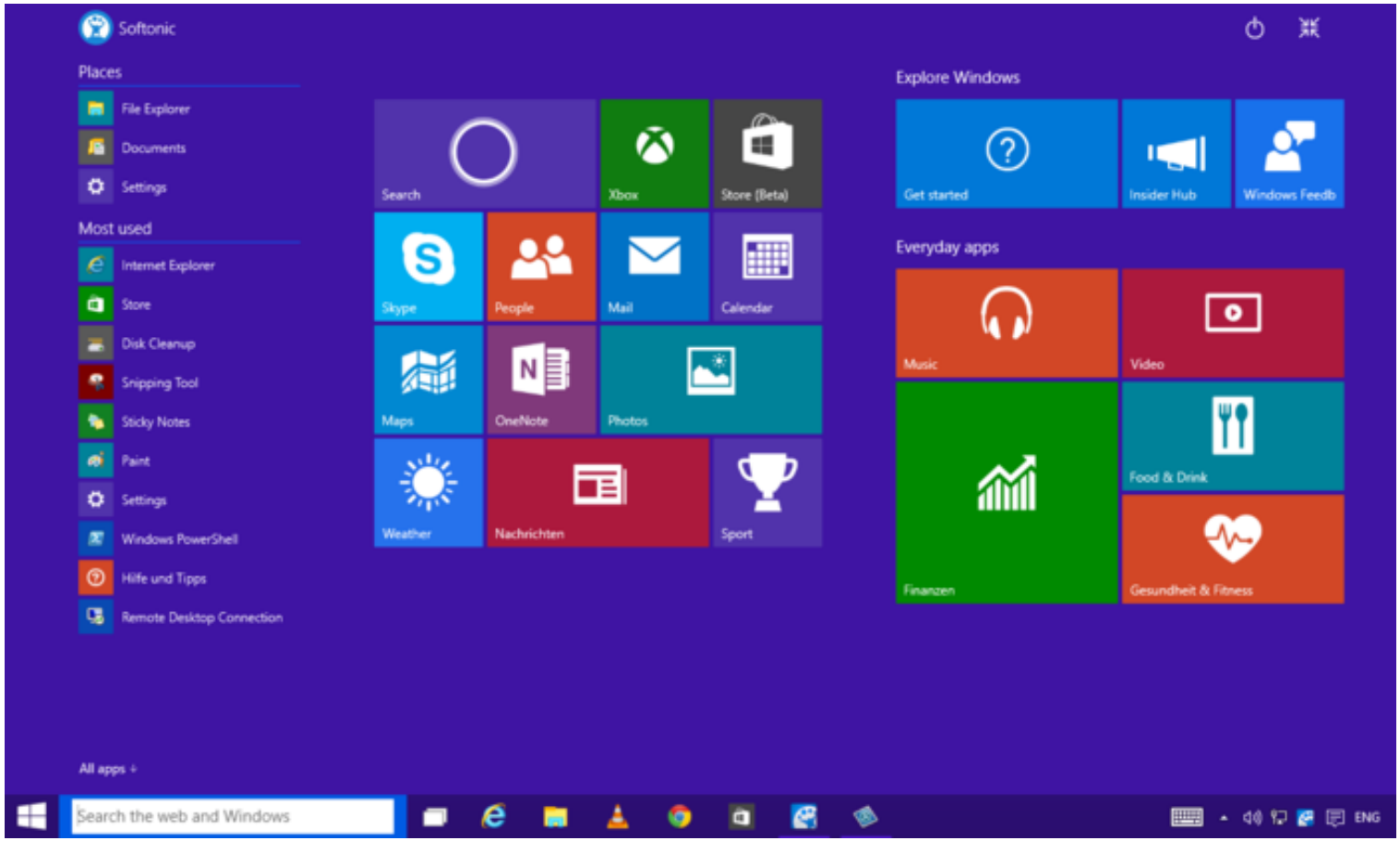
Building idioms: interaction vocabulary

- The uppermost layer contains idioms combining and structuring compounds using domain knowledge of the problem under consideration.
- The set of idioms opens the vocabulary to information about the particular problem the program is trying to address.
- In a GUI, it includes things like labeled buttons and fields, navigation bars, list boxes, icons, and even groups of fields and controls, or entire panes and dialogs.



Beyond global metaphors

- We have moved (in a sense) beyond metaphors:



User expectations of affordances

- In the real world, an object does what it can do as a result of its physical form and its connections with other physical objects.
- In the digital world, an object does what it can do because a programmer imbued it with the power to do something.
- We can discover a great deal about how objects work by physical inspection.
- On a computer screen, though, we can see a raised, three-dimensional rectangle that clearly wants to be pushed like a button, but this doesn't necessarily mean that it should be pushed.
- It could literally do almost anything.

Building idioms: interaction vocabulary

- When we render a button on the screen, we are making a contract with the user that that button will visually change when pressed.
- It will appear to be depressed when the mouse button is clicked over it.
- Further, the contract states that the button will perform some reasonable work that is accurately described by its legend.
- This is relatively rare for pushbuttons, but all too common for other controls where the lack of affordances can make it difficult to differentiate between controls, content, and ornamentation.
- Make sure that your program delivers on the expectations it sets via the use of manual affordances.

Building Blocks of Visual Interface Design

Visual Interface Design (VID)

- With interactive products, communication of idioms commonly happens visually.
- Visual interface design is a frequently misunderstood discipline, largely because of its similarities to visual art and graphic design.
- It is also commonly mischaracterized as skinning the interface.
- Visual interface design is critical and unique, and it must be conducted in concert with interaction design.

The building blocks of VID

- Interface design is concerned with the treatment and arrangement of visual elements to communicate behavior.
- Every element in a visual composition has a number of properties that work together to create meaning.
- There is rarely an inherent meaning to any one of these properties.
- Rather, the differences and similarities in the way these properties are applied to each element come together to allow users to make sense of an interface.
- When two objects share properties, users will assume they are related or similar.
- When users perceive contrast in these properties, they assume the items are not related, and the items with the greatest contrast tend to demand our attention.

VID Building Blocks: Shape

- Is it round, square, or amoeba-like? Shape is the primary way we recognize what an object is.
- However, distinguishing among different shapes takes a higher level of attention than distinguishing some other properties such as color or size.
- This means shape is not the best property to contrast when your purpose is to capture user attention.
- The weakness of shape as a factor in object recognition is apparent to anyone who has mistakenly selected iTunes instead of iDVD, or iWeb instead of iPhoto.

VID Building Blocks: Size

- How big or small is it in relation to other items on the screen? Larger items draw our attention more.
- Size is also an ordered and quantitative variable, which means that people sequence objects in terms of their size.
- If we have four sizes of text, we assume relative importance increases with size.
- This makes size a useful property in conveying information hierarchies.
- Sufficient distinction in size is also enough to draw our attention quickly.

VID Building Blocks: Value

- How light or dark is it? Of course, the idea of lightness or darkness is meaningful primarily in context of the value of the background.
- On a dark background, dark type is faint, whereas on a light background, dark type is pronounced.
- Contrasts in value are something people perceive quickly and easily, so value can be a good tool for drawing attention to elements that need to stand out.
- Value is also an ordered variable – for example, darker colors on a map are easy to interpret as deeper water or denser population.

VID Building Blocks: Color

- Is it yellow, red, or orange? Differences in hue draw our attention quickly.
- Colors also take on meaning from the social contexts in which we've grown up.
- To Westerners who've grown up with traffic signals, red means stop and sometimes even danger.
- Similarly, white is associated with purity and peace in the West, and with funerals and death in Asia.
- Unlike size or value, though, hue is not intrinsically ordered and shouldn't be used to convey quantitative data
- As we have already discussed, color must be used judiciously: the carnival effect overwhelms users and limits ability to communicate.

VID Building Blocks: Orientation

- Is it pointing up, down, or sideways? This is a useful variable to employ when you have directional information to convey (up or down, backward or forward).
- Orientation can be difficult to perceive with some shapes or at small sizes, though, so it's best used as a secondary communication vector.
- For example, if you want to show the stock market is going down, you might want to use a downward-pointing arrow that's also red.

VID Building Blocks: Position

- Where is it relative to other elements? Like size, position is both an ordered and a quantitative variable, which means it's useful for conveying information about hierarchy.
- We can leverage the reading order of a screen to locate elements sequentially, with the most important or first used in the top and left.
- Position can also be used to create spatial relationships between objects on the screen and objects in the physical world.
- This is useful for creating natural mappings.

Principles of Visual Interaction Design

Principles of VID

- The human brain is a pattern-processing computer, making sense of visual information everywhere we look.
- Our brains manage the overwhelming amount of data our visual sense provides by discerning visual patterns and establishing a system of priorities for the things we see.
- Imagine manually calculating the trajectory of a thrown baseball in order to predict where it lands.
- Our eyes and brains together do it in a split second, without conscious effort on our part.
- To most effectively communicate the behavior and functions of a program to users, visual interface designers must take advantage of this innate visual processing capability.

Principals of VID

- Visual interfaces should:
  - Use visual properties to group elements and create a clear hierarchy.
  - Provide visual structure and flow at each level of organization.
  - Use cohesive, consistent, and contextually appropriate imagery.
  - Integrate style and function comprehensively and purposefully.
  - Avoid visual noise and clutter.
- We discuss each of these principles in more detail in the following.

VID: visual properties for grouping elements

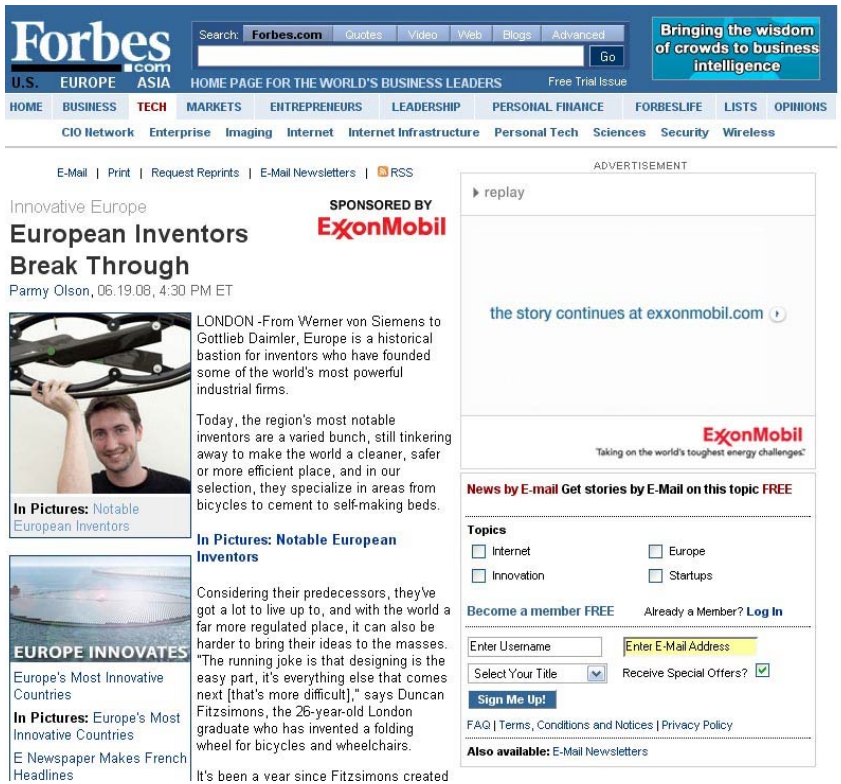
- A visual interface is based on visual patterns.
- It's a good idea to distinguish logical sets of controls or data by grouping them using visual properties such as color and dimensional rendering.
- By consistently applying these visual properties throughout an interface, you can create patterns that your users will learn to recognize.
- For example, in Windows XP, all buttons are raised with rounded corners and text fields are rectangular, slightly inset, and have a white background and blue border.
- Because of the consistent application of this pattern, there is no confusion as to what is a button and what is a text field, despite a number of similarities.

VID: creating hierarchy

- Based on scenarios, determine which controls and bits of data users need to understand instantly, which are secondary, and which are needed only by exception.
- This ranking informs the visual hierarchy.
- Next use hue, saturation, value, size, and position to distinguish levels of hierarchy.
- The most important elements should be larger, have greater contrast in hue, saturation, and value in relation to the background, and be positioned above other items.
- Items meant to stand out are best rendered in saturated colors.

VID: creating hierarchy

- Establishing a clear visual hierarchy is one of the hardest challenges in visual interface design, and takes skill and talent.
- It is the lack of visual hierarchy and accompanying confusion that jumps out at most users.



VID: creating hierarchy

- Establishing a clear visual hierarchy is one of the hardest challenges in visual interface design, and takes skill and talent.
- It is the lack of visual hierarchy and accompanying confusion that jumps out at most users.

VID: establishing relationships

- To convey which elements are related, return to your scenarios to determine not only which elements have similar functions but also which elements are used together most often.
- Elements that tend to be used together should generally be grouped spatially to minimize mouse movement.
- Spatial grouping makes it clear to users what tasks, data, and tools relate to each other, and can also imply sequence.
- Good grouping by position takes into account the order of tasks and subtasks and how the eye scans the screen: left to right (in Western languages), and generally from top to bottom.

VID: establishing relationships

- A good way to help ensure that a visual interface design employs hierarchy and relationships effectively is to use what graphic designers refer to as the squint test.
- Close one eye and squint at the screen with the other eye in order to see which elements pop out and which are fuzzy and which items seem to group together.
- Changing your perspective can often uncover previously undetected issues in layout and composition.

VID: visual structure and flow

- It's useful to think of user interfaces as being composed of visual and behavioral elements.
- These are used in groups, which are then grouped together into panes, which then may, in turn, be grouped into screens, views, or pages.
- There may be several such levels of structure in a sovereign application, and so it is critical that you maintain a clear visual structure.
- Visual structure in complex applications has a tremendous impact on flow.

VID: visual structure by alignment and grids

- Alignment of visual elements is one of the key ways that designers can help users experience a product in an organized, systematic way.
- In general, every element on the screen should be aligned with as many other elements as possible.
- The decision not to align two elements or groups of elements should be made judiciously, and always to achieve a specific differentiating effect.
- In particular, designers should take care to:
  - Align labels: labels for controls stacked vertically should be aligned with each other; unless labels differ widely in length, left-justification is easier for users to scan than right justification.
  - Align within a set of controls: a related group of check boxes, radio buttons, or text fields should be aligned according to a regular grid.
  - Align across control groups and panes: groups of controls and other screen elements should all follow the same grid wherever possible.

VID: visual structure by alignment and grids

- The Adobe Lightroom GUI is an excellent example of visual structure created by alignment:

VID: visual structure by alignment and grids

- A grid system is one of the most powerful tools available to the visual designer.
- A grid provides a uniform and consistent structure to layout, which is particularly important when designing an interface with several levels of visual or functional complexity.
- Typically, the grid divides the screen into several large horizontal and vertical regions.
- A well-designed grid employs an atomic grid unit that represents the smallest spacing between elements.
- For example, if your atomic unit is four pixels, spacing between screen elements and groups will all be in multiples of four pixels.

VID: visual structure by alignment and grids

- Most Content Management Systems (CMS) provide standard layout grids as templates for websites:

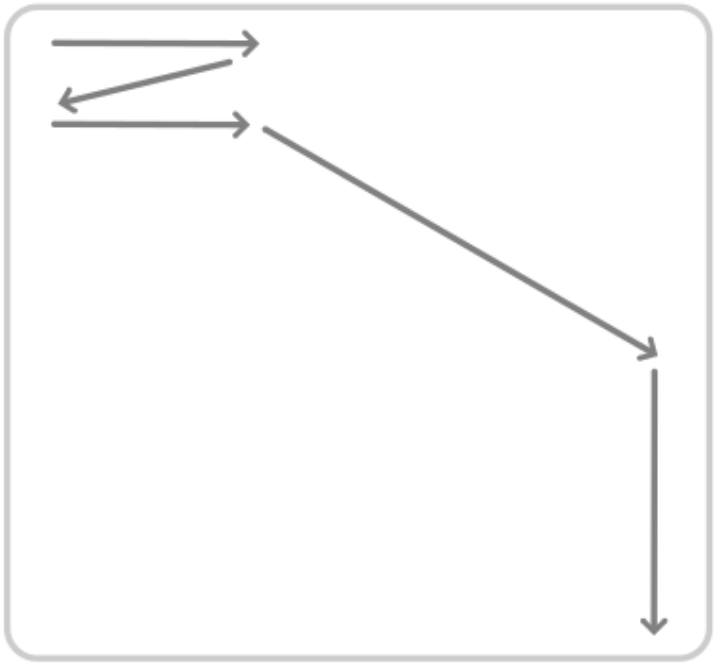
VID: visual structure by alignment and grids

- The use of a grid system in visual interface design provides several benefits:
  - Usability: because grids attempt to regularize positioning of elements, users are able to learn quickly where to find key interface elements. A well-designed grid greatly improves the readability of the screen.
  - Aesthetic appeal: by carefully applying an atomic grid and choosing the appropriate relationships between the various areas of the screen, a design can create a sense of order that feels comfortable to users and invites them to interact with the product.
  - Efficiency: standardizing your layouts will reduce the amount of labor required to produce high-quality visual interfaces. A well-defined and communicated grid system results in designs that can be modified and extended.

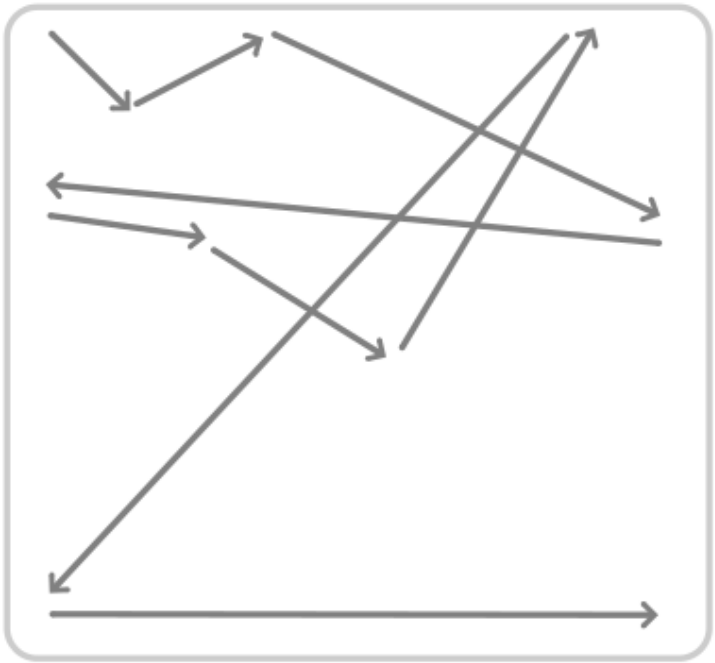
VID: grids and creating a logical path

- In addition to precisely following a grid, the layout must also properly structure an efficient logical path for users to follow through the interface.

Good logical flow
Eye movements match the path through the interface



Bad logical flow
Everything is all over the place



VID: use consistent and appropriate imagery

- Use of icons and other illustrative elements can help users understand an interface.
- A good understanding of personas and their mental models should provide a solid foundation for both the textual and visual language used in an interface.
- In addition to their functional role, icons can play a significant role in conveying the desired brand attributes.
- Bold, cartoonish icons may be great if you're designing a website for kids, whereas precise, conservatively rendered icons may be more appropriate to a productivity application.
- Whatever the style, it should be consistent – if some of your icons use bold black lines and rounded corners while others use thin, angular lines, the visual style won't hold together.

VID: consistent and appropriate imagery

- For more obviously concrete functions, some guidelines for visual design apply:
  - Represent both the action and an object acted upon to improve comprehension. Nouns and verbs are easier to comprehend together than verbs alone.
  - Beware of metaphors and representations that may not have the intended meanings for your target audience.
  - Group related functions visually to provide context, either spatially or, if this is not appropriate, using color or other common visual themes.
  - Keep icons simple and avoid excessive visual detail.
  - Reuse elements when possible, so users need to learn them only once.

VID: integrate style and function purposefully

- When designers choose to apply stylistic elements to an interface, it must be from a global perspective.
- Every aspect of the interface must be considered from a stylistic point of view, not simply as individual controls or other visual elements.
- We need to make sure that the functional aspects of the program's visual interface design are in complete harmony with the visual brand.
- You program's behavior is part of its brand, and your user's experience with your product should reflect the proper balance of form, content, and behavior.

VID: form versus function

- Visual style is a tempting diversion for many stakeholders, but the use of stylized visual elements needs to be carefully controlled within an interface.
- The basic shape, behavior, and visual affordance of controls should be driving factors in developing the visual style
- Purely aesthetic considerations should not interfere with the meaning of the interface or a user's ability to interact with it.

VID: avoid visual noise and clutter

- Visual noise within an interface is caused by superfluous visual elements that distract from the primary objective of directly communicating software function and behavior.
- Visual noise can take the form of over-embellished and unnecessarily dimensional elements, overuse of rules boxes and other visually heavy elements to separate controls.
- Cluttered interfaces attempt to provide an excess of functionality in a constrained space, resulting in controls that visually interfere with each other.
- Visually baroque, disorderly, or overcrowded screens raise the cognitive load for the user

VID: keep it simple (stupid)

- In general, interfaces should use simple geometric forms, minimal contours, and a restricted color palette.
- Typography should not vary widely in an interface: typically one or two typefaces, specified to display at just a few sizes.
- When multiple design elements are required for similar or related logical purposes, they should be rendered in a consistent fashion to take advantage of inheritance.
- Inheritance provides the opportunity for an understanding of one element to transfer to other elements that are similar.
- Elements intended to stand out should be visually contrasted with other elements through adjustment of one or more visual properties, such as size, color, and position.

VID: keep it simple (stupid)

- Unnecessary variation is the enemy of a coherent, usable design:
  - If the spacing between two elements is nearly the same, make that spacing exactly the same.
  - If two typefaces are nearly the same size, adjust them to be the same size.
  - Every visual element and every difference in color, size, or other visual property should be there for a reason.
  - If you can't articulate a good reason why it's there, get rid of it.
- Good visual interfaces, like any good visual design, are visually efficient.
- Take things away until the design breaks, then put that last thing back in.

Homework

Homework

Homework 22.1: good and bad grouping

Find some examples of good grouping of controls or other UI elements using the cues discussed in this lecture. Find some bad grouping.

Homework 22.2: Latest and Greatest

Everyone be sure to start reading our next (and last) Latest and Greatest paper:

Kostakos, Vassilis. "The big hole in HCI research." Interactions, 2015.