SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Chapter 3

Web Interface Design

3.3 Contextual Tools

Somaiya TRUST

Introduction

Designing Web Interfaces

- Principle One: Make It Direct
- **Principle Two: Keep It Lightweight**
- Principle Three: Stay on the Page
- Principle Four: Provide an Invitation
- Principle Five: Use Transitions
- Principle Six: React Immediately

Contextual Tools

Fitts Law

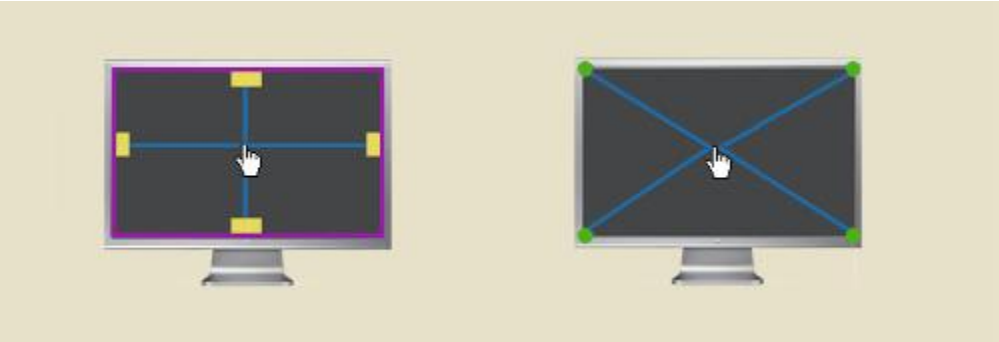
It is based on the work of psychologist Paul Fitts in 1954, which determines

“The time required to rapidly move to a target area is a function of the ratio between the distance to the target and the width of the target”.

For example, this law influenced the convention of making interactive buttons large (especially on finger-operated mobile devices) — smaller buttons are more difficult (and time-consuming) to click.

Fitts Law – Prime Pixel

The ideal fitts' law application would let us know where's the user cursor is when he lands on the application. This point would be called **Prime Pixel**, the point where the user would carry out all his tasks from.

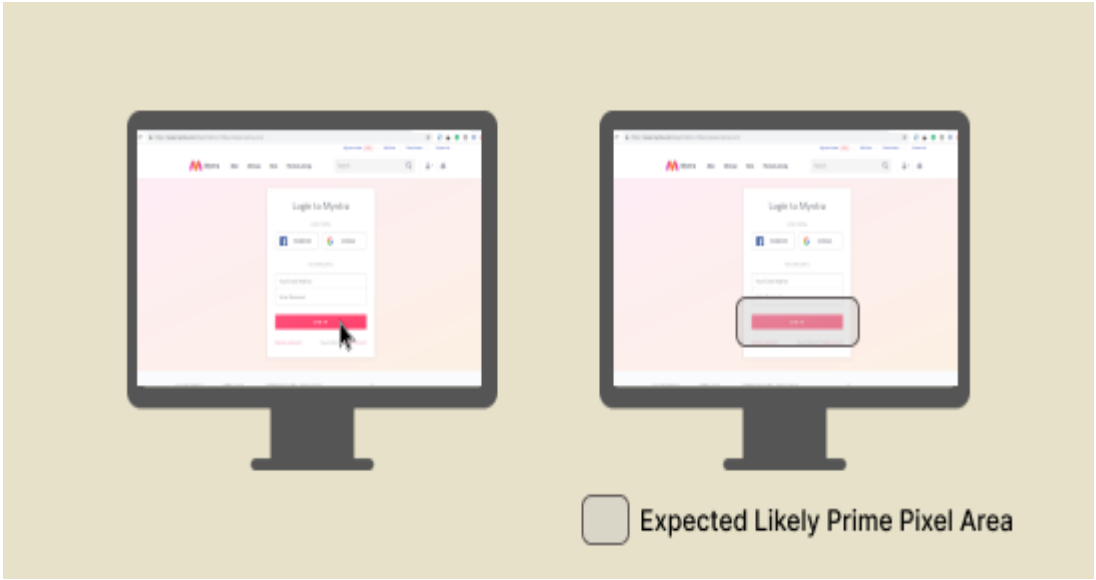


Unfortunately, while our browser and applications(example windows apps, desktop games) can utilize the prime pixel but a website cannot. Even if we determine the prime pixel, it would change every time the user moves the cursor.

Fitts Law – Prime Pixel

Though one can determine the **Likely Prime Pixel** by the actions done by the user.

So, for example, if a user clicks “login”, the box that appears for the user to do so should be as close to the “login” option as possible, and the “submit” button should be as close to that text box as possible, too.



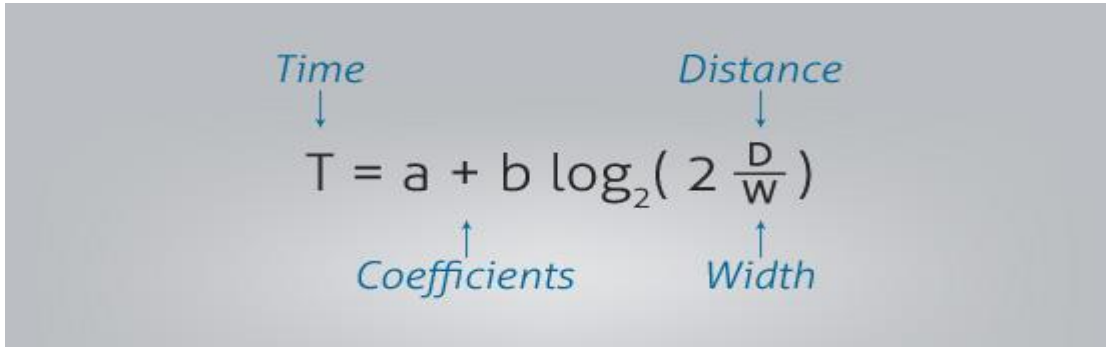
Fitts Law – Magic Pixel

- There are four other pixels that matter to Fitts' Law in web design.
- We find these in each corner of the screen. They're called "magic pixels", and they provide a boundary. User cannot go beyond the corners.
- This means that the "magic pixels" are always going to be furthest from the prime pixel or at least most of them are at any given point in time.

The magic pixels in operation systems

Fitts' law

- Fitts' law is centered around a mathematical equation that is used to illustrate the time it takes to reach a target object. The equation is as follows:



  - A target object, in the context of User Interface, can be an **interactive element such as a submit button, a hyperlink, and an input field in a web form**.
  - The idea is the quicker you can reach a target object, the more convenient and easy it is to use.

Fitts' law

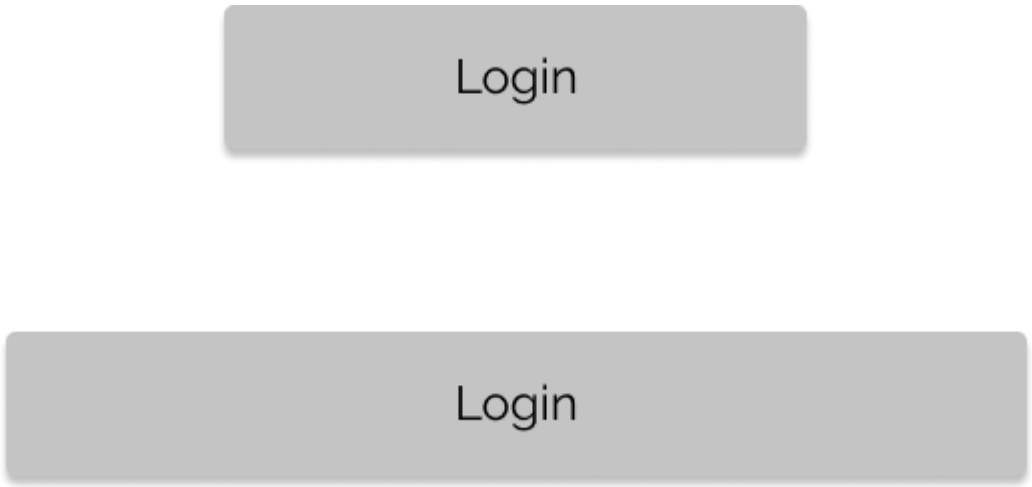
Fitts' law is a binary logarithm.

This means that the predicted results of the usability of an object run along a curve, not a straight line.

In web design, this means that a very small object will become significantly easier to click when given a 20% size increase, while a very large object will not share the same boost in usability when given the same 20% boost in size.

Fitts' law

Creating Larger object



Fitts' law

Minimizing Movement

- If you place the links and buttons users are most likely to access on a regular basis next to each other, rather than distribute them across the interface, you will speed up interaction by reducing the number of pixels the cursor will have to travel.
- But, Arranging elements strictly according to this formula can cause conflict with other important design principles, such as the principle of grouping and separating different classes of functionality or content.

Fitts' law

- The goal of Fitts's index of performance (PDF) is to quantify the information capacity of the human motor system.
- In other words: it aims to rank input methods according to the amount of physical effort they require to execute a certain command.

Fitts' law

Input methods that are more difficult to perform can sometimes actually prevent mistakes. For example, mobile devices are often carried around in pockets, which can trigger commands by accident.

In those situations, high-precision input methods are deployed, which use a higher input difficulty to make sure that a command is not executed accidentally. However, these input methods are also a way of making users aware of the severity of the command.

Contextual Tools

We could simply isolate our functionality into islands of tools (toolbars and menus). But this would work against Fitts's Law by requiring more effort from the user.

It would also add more visual weight to the page. Instead of interacting with the functionality separately, we can bring the functionality into the content with Contextual Tools.

Contextual Tools

Contextual Tools are the Web's version of the desktop's right-click menus.

Always-Visible Tools

Place Contextual Tools directly in the content.

Hover-Reveal Tools

Show Contextual Tools on mouse hover.

Contextual Tools

Toggle-Reveal Tools

A master switch to toggle on/off Contextual Tools for the page.

Multi-Level Tools

Progressively reveal actions based on user interaction.

Secondary Menus

Show a secondary menu (usually by right-clicking on an object)

Always Visible Tool

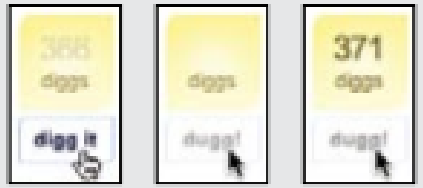
Visible tool

Beside each story is a digg scorecard. Just below is the "digg it" button. The digg button shows for all stories.

Other actions are represented less prominently.

Invitation

On mouse hover, the digg button border changes to a darker color and the text label changes to black. High-lighting is an effective way to signal interactivity.

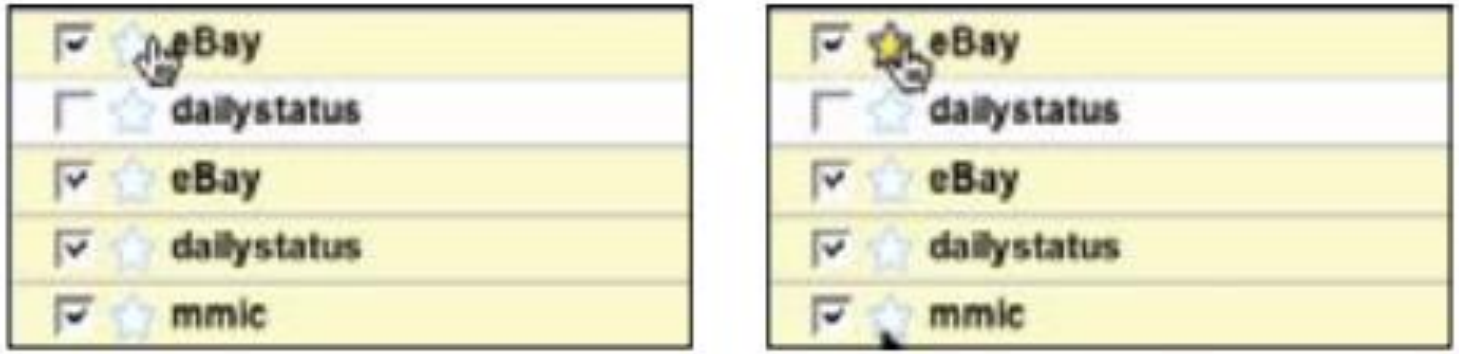


Completion

Once the user clicks the "digg it" button, the vote is counted. The current vote fades out and then the new digg count (including your vote) appears instantly. The digg button changes to "dugg" and is no longer clickable (indicated by the gray text).

Discoverability

- Discoverability is a primary reason to choose Always-Visible Tools.
- On the flip side, it can lead to more visual clutter.



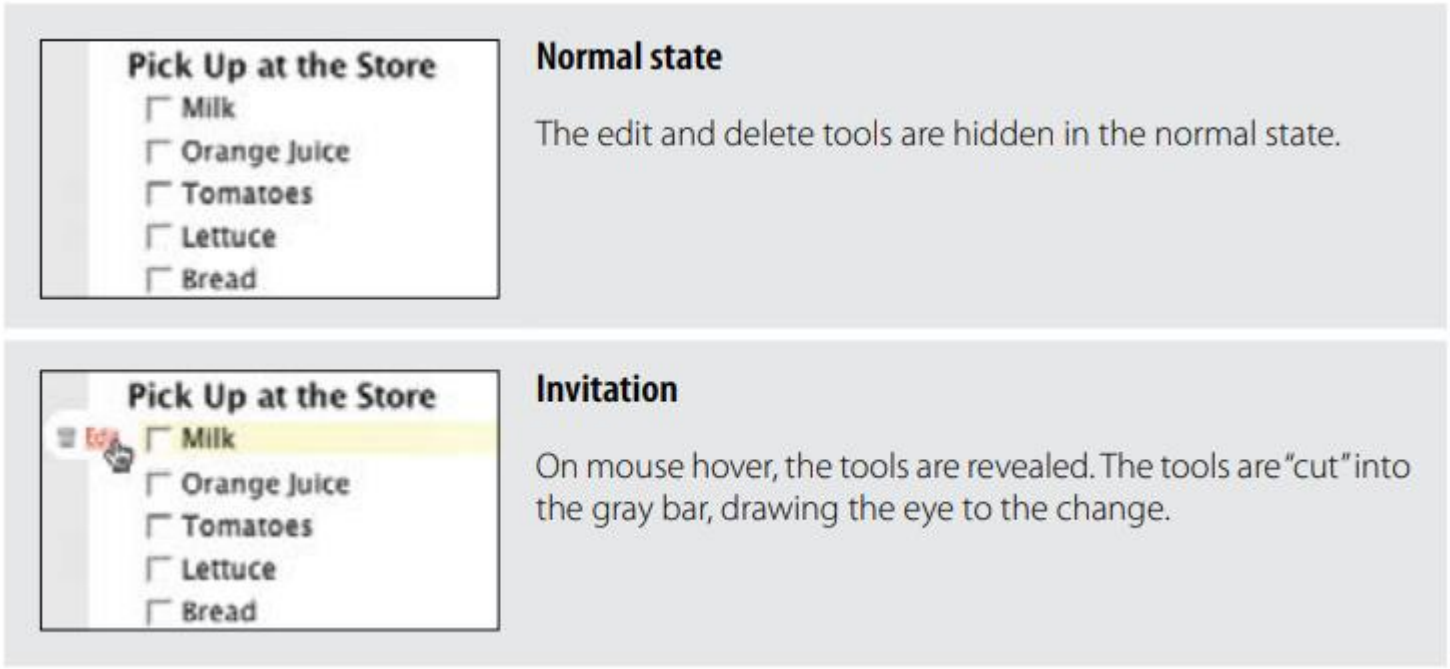
Google Mail uses Contextual tools to flag favourites

Always Visible Tool Best Practices

- Make your Contextual Tools always visible if it is important to make a prominent call to action.
- Keep visual clutter to a minimum.
- Keep the number of visual items to a minimum.

Hover-Reveal Tools

Instead of making Contextual Tools always visible, we can show them on demand. One way to do this is to reveal the tools when the user pauses the mouse over an object.



Visual noise

Hover-Reveal Tools

Yahoo! Buzz reveals additional tools for the top searches when the user hovers over each item.

Anti –Pattern :Hover-Reveal Tools

An early version of the Yahoo! for Teachers beta revealed Contextual Tools in an overlay; the overlay covered more than half of the item to its right

Hover Reveal Tools

In the redesigned version, tools were shown on hover directly surrounding the image instead of in an overlay

Anti Pattern- Mystery meat

A very early version of Zooomr revealed a number of unidentifiable icons when the mouse hovered over a photo.

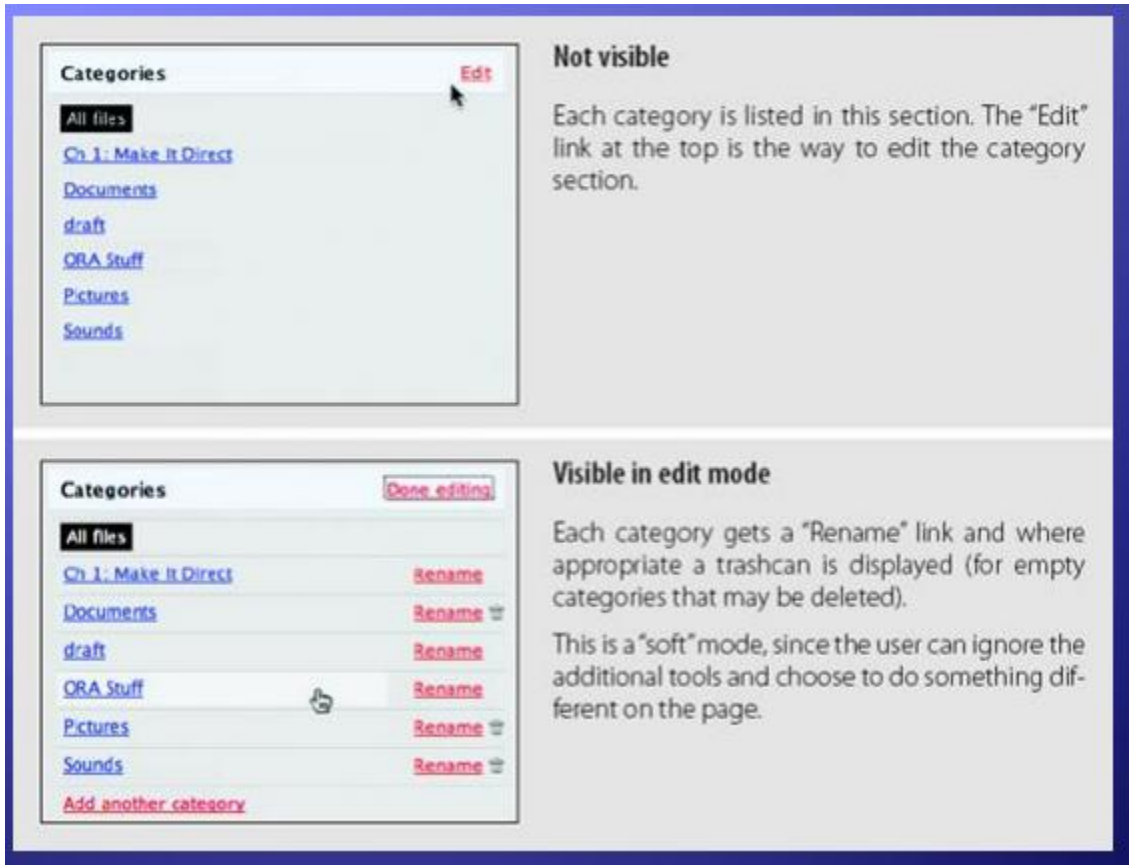
Best practices for Hover Reveal Tool

- Hide Contextual Tools behind a mouse hover if the actions are secondary and you want to make the content more important (or other visible commands more important).
- Avoid using overlays when revealing additional tools. They will lead to the Hover and Cover anti-pattern, as well as require the user to perform mouse gymnastics to accomplish the most basic tasks.
- When additional tools are revealed, make sure that all parts of the page remain stable.

Best practices for Hover Reveal Tool

- Avoid any element shifting by a few pixels or page elements moving around. These cause the user to be directed away from what is really happening.
- Make sure revealed icons are clear and understandable. When possible, just use text labels.
- Activate tool overlays instantly. Unlike informational overlays, the user needs the additional tools to be available for immediate interaction.

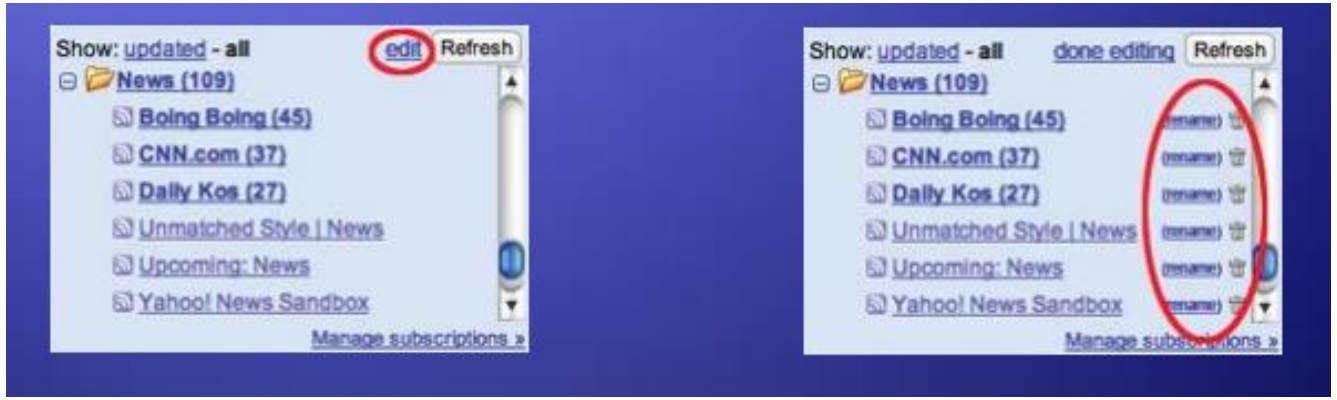
Toggle Reveal Tools



Toggle Reveal Tools

Soft mode

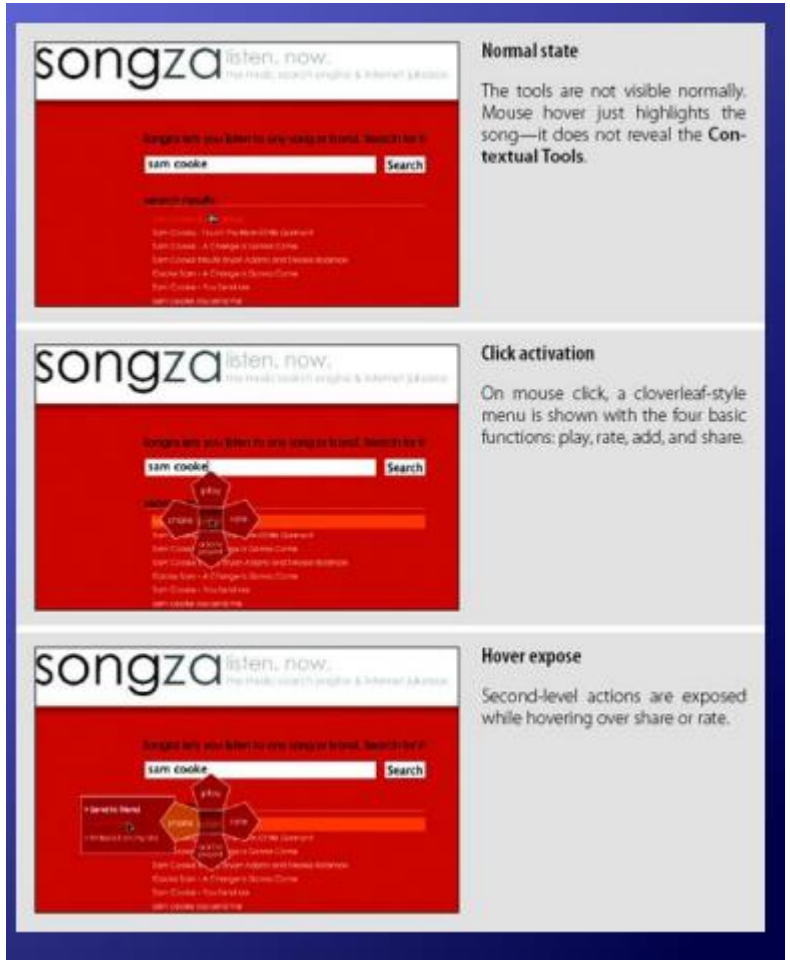
- It is a good thing to avoid specific modes in an interface
- The user is not trapped in the mode



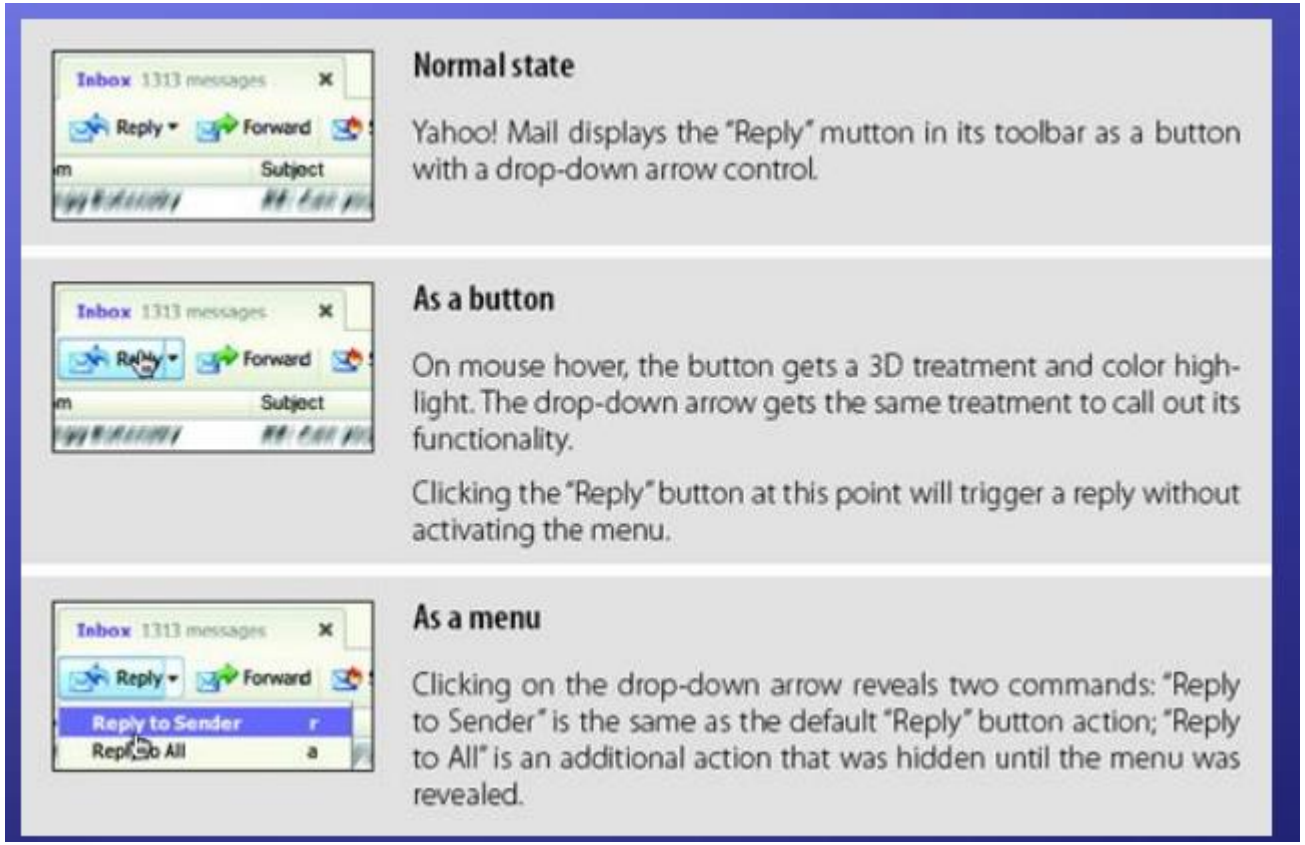
Best Practices – Toggle Reveal Tool

- Toggle a tool mode for an area or page when the actions are not the main flow, but you want to provide the most direct way to act on these objects when the need arises.
- Make the activation and deactivation of the edit mode as symmetrical as possible.
- Keep the transition between display and edit as seamless as possible to provide a "soft mode" for editing

Multi Level Tool



Muttons – Menu + Buttons



Muttons

Muttons are used to:

- Provide a default button action ("Reply to Sender")
- Provide a clue that there are additional actions
- Provide additional actions in the drop-down

Best Practices for Multi Level Tools

- Use Multi-Level Tools when you want to avoid revealing Contextual Tools on a mouse hover.
- Use Multi-Level Tools to make activation explicit.
- Use muttons when you have a default action that the user normally takes but alternate actions are still fairly frequent.
- Avoid cascades where possible. Users have a hard time maneuvering the various mouse turns that are required to get to these secondary menus.
- Keep actions as close to the activation point as possible.

Secondary Menu

Normal view of route

Routes give no indication of additional functionality when not hovered over.

Secondary Menu

Invitation

When the mouse is over the route, potential stops are marked with a white circle.

Menu

Right-clicking on the item exposes four commands that act on the point selected: "Add a destination", "Zoom in", "Zoom out", and "Center map here".

Secondary Menu

- Secondary Menus have not been common in web applications.
- Conflict with browser menu
  - The browser inserts its own right-click menu.
  - Standard browser menu
  - application-specific menu