UNIT V

WEB INTERFACE DESIGN

Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow. Case Studies.

User Interface Design Basics

User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture.

Choosing Interface Elements

Users have become familiar with interface elements acting in a certain way, so try to be consistent and predictable in your choices and their layout. Doing so will help with task completion, efficiency, and satisfaction.

Interface elements include but are not limited to:

Input Controls: buttons, text fields, checkboxes, radio buttons, dropdown lists, list boxes, toggles, date field

Navigational Components: breadcrumb, slider, search field, pagination, slider, tags, icons

Informational Components: tooltips, icons, progress bar, notifications, message boxes, modal windows

Containers: accordion

There are times when multiple elements might be appropriate for displaying content. When this happens, it's important to consider the trade-offs. For example, sometimes elements that can help save you space, put more of a burden on the user mentally by forcing them to guess what is within the dropdown or what the element might be.

Best Practices for Designing an Interface

Everything stems from knowing your users, including understanding their goals, skills, preferences, and tendencies. Once you know about your user, make sure to consider the following when designing your interface:

Keep the interface simple. The best interfaces are almost invisible to the user. They avoid unnecessary elements and are clear in the language they use on labels and in messaging.

Create consistency and use common UI elements. By using common elements in your UI, users feel more comfortable and are able to get things done more quickly. It is also important

to create patterns in language, layout and design throughout the site to help facilitate efficiency. Once a user learns how to do something, they should be able to transfer that skill to other parts of the site.

Be purposeful in page layout. Consider the spatial relationships between items on the page and structure the page based on importance. Careful placement of items can help draw attention to the most important pieces of information and can aid scanning and readability.

Strategically use color and texture. You can direct attention toward or redirect attention away from items using color, light, contrast, and texture to your advantage.

Use typography to create hierarchy and clarity. Carefully consider how you use typeface. Different sizes, fonts, and arrangement of the text to help increase scanability, legibility and readability.

Make sure that the system communicates what's happening. Always inform your users of location, actions, changes in state, or errors. The use of various UI elements to communicate status and, if necessary, next steps can reduce frustration for your user.

Think about the defaults. By carefully thinking about and anticipating the goals people bring to your site, you can create defaults that reduce the burden on the user. This becomes particularly important when it comes to form design where you might have an opportunity to have some fields pre-chosen or filled out.

DRAG AND DROP

Interesting Moments

At first blush, drag and drop seems simple. Just grab an object and drop it somewhere. But, as always, the devil is in the details. There are a number of individual states at which interaction is possible. We call these microstates interesting moments:

- How will users know what is draggable?
- What does it mean to drag and drop an object?
- Where can you drop an object, and where is it not valid to drop an object?
- What visual affordance will be used to indicate draggability?
- ill valid and invalid drop targets be signified?
- Do you drag the actual object?
- Or do you drag just a ghost of the object?
- Or is it a thumbnail representation that gets dragged?
- What visual feedback should be used during the drag and drop interaction?

What makes it challenging is that there are a lot of events during drag and drop that can be used as opportunities for feedback to the user. Additionally, there are a number of elements

on the page that can participate as actors in this feedback loop.

The Events: There are at least 15 events available for cueing the user during a drag and drop interaction:

Page Load: Before any interaction occurs, you can pre-signify the availability of drag and drop. For example, you could display a tip on the page to indicate draggability.

Mouse Hover: The mouse pointer hovers over an object that is draggable.

Mouse Down: The user holds down the mouse button on the draggable object.

Drag Initiated: After the mouse drag starts (usually some threshold—3 pixels).

Drag Leaves Original Location: After the drag object is pulled from its location or object that contains it.

Drag Re-Enters Original Location: When the object re-enters the original location.

Drag Enters Valid Target: Dragging over a valid drop target.

Drag Exits Valid Target: Dragging back out of a valid drop target.

Drag Enters Specific Invalid Target: Dragging over an invalid drop target.

Drag Is Over No Specific Target: Dragging over neither a valid or invalid target. Do you treat all areas outside of valid targets as invalid?

Drag Hovers Over Valid Target : User pauses over the valid target without dropping the object. This is usually when a spring loaded drop target can open up. For example, drag over a folder and pause, the folder opens revealing a new area to drag into.

Drag Hovers Over Invalid Target: User pauses over an invalid target without dropping the object. Do you care? Will you want additional feedback as to why it is not a valid target?

Drop Accepted : Drop occurs over a valid target and drop has been accepted.

Drop Rejected: Drop occurs over an invalid target and drop has been rejected. Do you zoom back the dropped object?

Drop on Parent Container: Is the place where the object was dragged from special? Usually this is not the case, but it may carry special meaning in some contexts.

The Actors : During each event you can visually manipulate a number of actors. The page elements available include:

- Page (e.g., static messaging on the page)
- Cursor
- Tool Tip
- Drag Object (or some portion of the drag object, e.g., title area of a module)
- Drag Object's Parent Container
- Drop Target

Interesting Moments Grid

That's 15 events times 6 actors. That means there are 90 possible interesting moments—each requiring a decision involving an almost unlimited number of style and timing choices. You can pull all this together into a simple interesting moments grid for Drag and Drop.

	Page Generation	Mouse Hover	Drag Initiated	Drag over Valid	Drag over Invalid	Drag over Original	Drop Accepted	Drop Rejected	Drop on Original
Page Content	Hint	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Cursor	Normal	Move Cursor	Move Cursor	Move Cursor	Move Cursor	Move Cursor	Normal	Normal	Normal
Drag Object	Normal	Normal	Reduced Opacity & Tracking	Reduced Opacity & Tracking	Reduced Opacity & Tracking + Invalid Badge	Reduced Opacity & Tracking	2. Modules animates into the area just below insertion bar 3. Module comes to rest in new area 4. Modules slide up in a self-healing transition to close hole	Normal Opacity + Zoom Back to Original	Normal Opacity + Zoom Back to Original
Orig Location	Normal	Normal	Hole Opens	Hole Remains	Hole Remains	Hole Remains	Hole Remains	Hole refilled with drag object	Hole refilled with drag object
Drop Target	Normal	Normal	Normal	Insertion Bar	N/A	N/A	1. Insertion Bar Removed	N/A	N/A

Figure: A simplified interesting moment's grid for the original My Yahoo! drag and drop design;* it provided a way to capture the complexities of drag and drop into a single page

Purpose of Drag and Drop

Drag and drop can be a powerful idiom if used correctly. Specifically it is useful for:

- **Drag and Drop Module:** Rearranging modules on a page.
- **Drag and Drop List :** Rearranging lists.
- **Drag and Drop Object :** Changing relationships between objects.
- **Drag and Drop Action:** Invoking actions on a dropped object.
- **Drag and Drop Collection :** Maintaining collections through drag and drop.

Drag and Drop Module

One of the most useful purposes of drag and drop is to allow the user to directly place objects where she wants them on the page. A typical pattern is Drag and Drop Modules on a page. Netvibes provides a good example of this interaction pattern
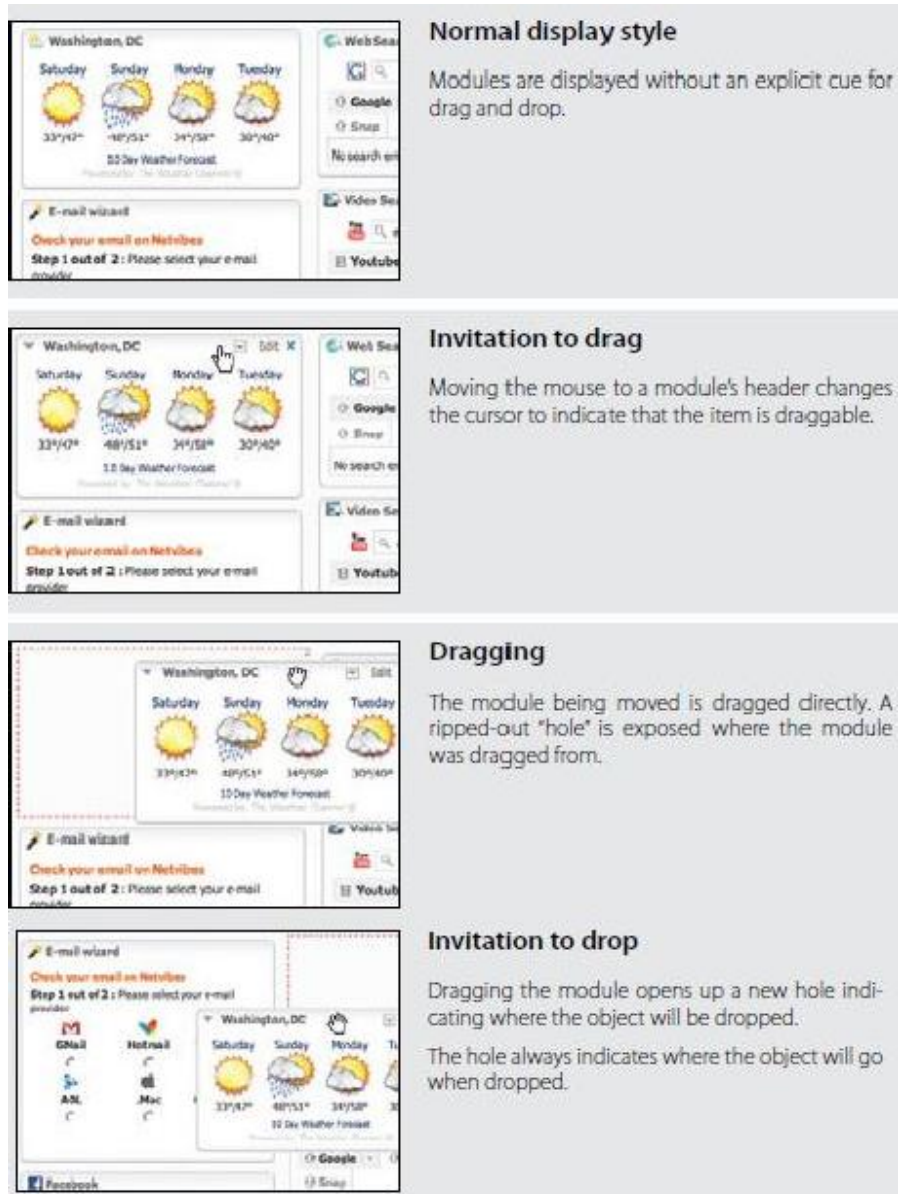
Normal display style

Modules are displayed without an explicit cue for drag and drop.

Invitation to drag

Moving the mouse to a module's header changes the cursor to indicate that the item is draggable.

Dragging

The module being moved is dragged directly. A ripped-out "hole" is exposed where the module was dragged from.

Invitation to drop

Dragging the module opens up a new hole indicating where the object will be dropped.

The hole always indicates where the object will go when dropped.

Figure: Netvibes allows modules to be arranged directly via drag and drop; the hole cues what will happen when a module is dropped

Considerations

Netvibes allows its modules to be rearranged with drag and drop. A number of interesting moments decide the specific interaction style for this site. Figure shows the interesting moments grid for Netvibes.

	Mouse Hover	Mouse Down	Drag Initiated	Drag Hovers over Valid Target*	Drop Accepted
Cursor	Change to a hand with finger pointing.*	Change to a hand/move cursor.	No change.*		Cursor returns to normal style.
Dragged Module			Module is dragged with full opacity.		Dragged version is removed.
Dragged Modules Original Location			Hole is shown as a red dashed outline.		Hole is removed.
Drop Target				Hole (red dashed outline) is moved to the new drop spot. Other modules shift to close prior hole.	Module is placed in the new location.
Notes	* A better approach might be to signal draggability with the hand/move cursor.		* On drag initiated, it would be better to switch to a hand that looks like a grab.	* Triggers when the dragged module's title bar has moved past the midpoint of the module's header.	

While dragging, it is important to make it clear what will happen when the user drops the dragged object. There are two common approaches to targeting a drop:

• Placeholder target

• Insertion target

Placeholder target

Netvibes uses a placeholder (hole with dashed outline) as the drop target. The idea (illustrated in Figure) is to always position a hole in the spot where the drop would occur. When module 1 starts dragging, it gets ‒ripped‖ out of the spot. In its place is the placeholder target (dashed outline). As 1 gets dragged to the spot between 3 and 4, the placeholder target jumps to fill in this spot as 4 moves out of the way.
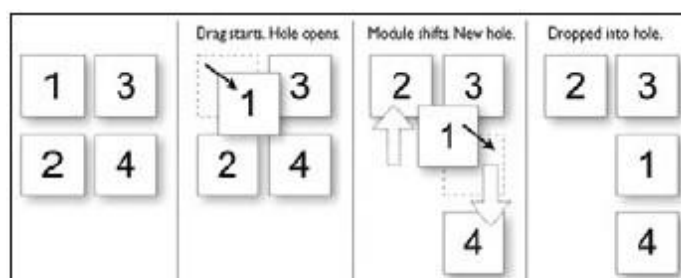
The hole serves as a placeholder and always marks the spot that the dragged module will land when dropped. It also previews what the page will look like (in relation to the other modules) if the drop occurs there. For module drag and drop, the other modules only slide up or down within a vertical column to make room for the dragged module.

One complaint with using placeholder targets is that the page content jumps around a lot during the drag. This makes the interaction noisier and can make it harder to understand what is actually happening. This issue is compounded when modules look similar. The user starts dragging the modules around and quickly gets confused about what just got moved. One way to resolve this is to provide a quick animated transition as the modules move. It is important, however, that any animated transitions not get in the way of the normal interaction.

Boundary-based placement. Since most sites that use placeholder targeting drag the module in its original size, targeting is determined by the boundaries of the dragged object and the boundaries of the dragged-over object. The mouse position is usually ignored because modules are only draggable in the title (a small region). Both Netvibes and iGoogle take the boundary-based approach. But, interestingly, they calculate the position of their placeholders differently.

In Netvibes, the placeholder changes position only after the dragged module's title bar has moved beyond the dragged-over module's title bar. In practice, this means if you are moving a small module to be positioned above a large module, you have to move it to the very top of

the large module. In Figure you have to drag the small ─To Do List‖ module all the way to the top of the ─Blog Directory‖ module before the placeholder changes position.

Insertion target

Placeholder positioning is a common approach, but it is not the only way to indicate drop targeting. An alternate approach is to keep the page as stable as possible and only move around an insertion target (usually an insertion bar). A previous version of My Yahoo! Used the insertion bar approach as the dragged module was moved around.
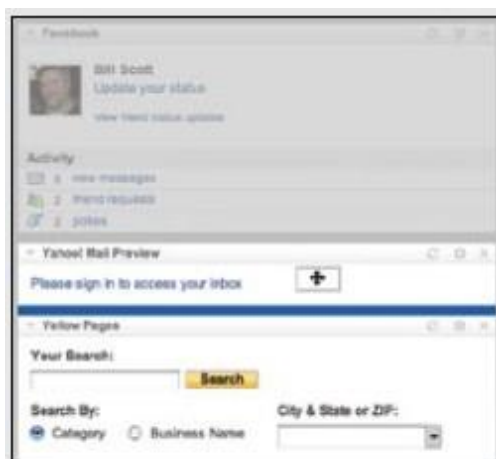
Insertion bar appears

The dragged module representation is now be low the Yahoo! Mail Preview module's midpoin

The insertion bar is rendered to show that th Facebook module will be placed just below th Mail Preview module if dropped.

Drag distance

Dragging the thumbnail around does have other issues. Since the object being dragged is small, it does not intersect a large area. It requires moving the small thumbnail directly to the place it will be dropped. With iGoogle, the complete module is dragged. Since the module will always be larger than the thumbnail, it intersects a drop target with much less movement. The result is a shorter drag distance to accomplish a move

Drag rendering

How should the dragged object be represented? Should it be rendered with a slight transparency (ghost)? Or should it be shown fully opaque? Should a thumbnail representation be used instead?

Drag and Drop List

The Drag and Drop List pattern defines interactions for rearranging items in a list. 37 Signal's Backpackit allows to-do items to be rearranged with Drag and Drop List

Normal display state

List items are displayed without any indication that the items can be rearranged.

Invitation to drag

One of the in-context tools revealed during mouse hover shows a four-way arrow indicating that the object can be moved.

Dragging this object allows the list item to be moved.

Dragging

Rearranging occurs in real time. An empty slot is exposed where the dragged item will fit.

Dropped

The item snaps into the new location (where the hole was opened up).

Considerations

Backpackit takes a real-time approach to dragging items. Since the list is constrained, this is a natural approach to moving objects around in a list. You immediately see the result of the drag.

Placeholder target

This is essentially the same placeholder target approach we discussed earlier for dragging and dropping modules. The difference is that when moving an item in a list, we are constrained to a single dimension. Less feedback is needed. Instead of a ‒ripped-out‖ area (represented earlier with a dotted rectangle), a simple hole can be exposed where the object will be placed when dropped.

Insertion target

Drag and Drop Modules, placeholder targeting is not the only game in town. You can also use an insertion bar within a list to indicate where a dropped item will land. Netflix uses an insertion target when movies are dragged to a new location in a user's movie queue

The upside to this approach is that the list doesn't have to shuffle around during drag. The resulting experience is smoother than the Backpack it approach. The downside is that it is not as obvious where the movie is being positioned. The insertion bar appears under the ghosted item. The addition of the brackets on the left and right of the insertion bar is an attempt to make the targeting clearer.

Non‒drag and drop alternative

Besides drag and drop, the Netflix queue actually supports two other ways to move objects around:

Edit the row number and then press the ‒Update DVD Queue‖ button. Click

the ‒Move to Top‖ icon to pop a movie to the top.

Modifying the row number is straightforward. It's a way to rearrange items without drag and drop. The ‒Move to Top‖ button is a little more direct and fairly straightforward (if the user really understands that this icon means ‒move to top‖). Drag and drop is the least discoverable of the three, but it is the most direct, visual way to rearrange the list. Since rearranging the queue is central to the Netflix customer's satisfaction, it is appropriate to allow multiple ways to do so.

Hinting at drag and drop

When the user clicks the ‒Move to Top‖ button, Netflix animates the movie as it moves up. But first, the movie is jerked downward slightly and then spring-loaded to the top

Click "Move to Top"

Clicking the "Move to Top" button starts the movie moving to the top.

Spring loaded

The movie does not immediately start moving up. Instead, it drops down and to the right slightly. This gives the feeling that the movie is being launched to the top.

Animated move to top

The movie then animates very quickly to show it is moving to the top.

The combination of the downward jerk and then the quick animation to the top gives a subtle clue that the object is draggable. This is also an interesting moment to advertise drag and drop. After the move to top completes, a simple tip could appear to invite users to drag and drop. The tip should probably be shown only once, or there should be a way to turn it off. Providing an invitation within a familiar idiom is a good way to lead users to the new idiom.

Drag lens

Drag and drop works well when a list is short or the items are all visible on the page. But when the list is long, drag and drop becomes painful. Providing alternative ways to rearrange is one way to get around this issue. Another is to provide a drag lens while dragging. A drag lens provides a view into a different part of the list that can serve as a shortcut target. It could be a fixed area that is always visible, or it could be a miniature view of the list that provides more rows for targeting. The lens will be made visible only during dragging.

Drag and Drop Object

Another common use for drag and drop is to change relationships between objects. This is appropriate when the relationships can be represented visually. Drag and drop as a means of visually manipulating relationships is a powerful tool.

Cogmap is a wiki for organizational charts. Drag and Drop Object is used to rearrange
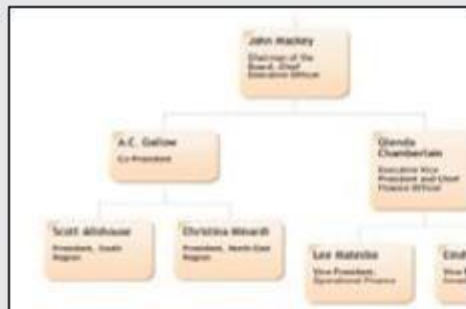
Normal display state

An organizational chart visually represents relationships.

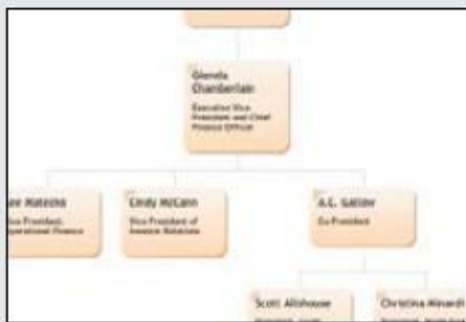**Invitation to drag**

When the mouse hovers over a member of the organization, the cursor changes to show draggability. In addition, the texture in the top-left corner changes to represent a dimpled surface. This hints at draggability.

Dragging

An insertion bar is used to indicate where the member will be inserted when dropped.

Dropped

When the dragged member is dropped, the chart is rearranged to accommodate the new location.

Fig: Cogmap allows organizational charts to be rearranged on the fly with drag and drop

Considerations

When object relationships can be clearly represented visually, drag and drop is a natural choice to make these type of changes. Cogmap uses the target insertion approach. This allows the dragging to be nondistracting, since the chart does not have to be disturbed during targeting.

Drag feedback: Highlighting

Bubbl.us, an online mind-mapping tool, simply highlights the node that will be the new parent
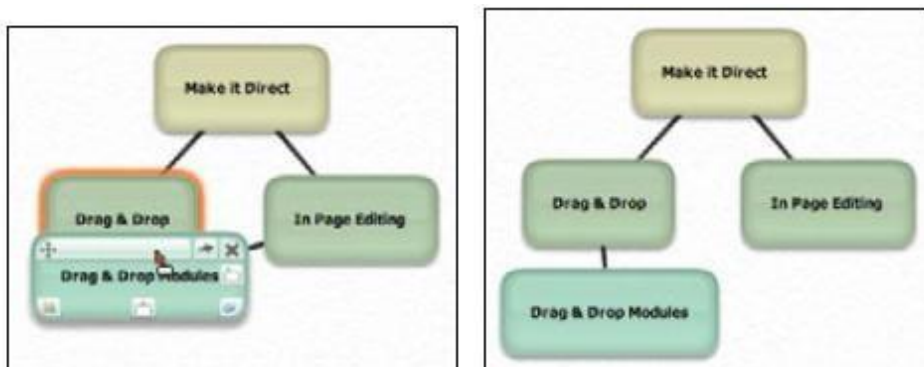
Drag and Drop Action

Drag and drop is also useful for invoking an action or actions on a dropped object. The Drag and Drop Action is a common pattern. Its most familiar example is dropping an item in the trash to perform the delete action.

Normally uploading files to a web application includes pressing the upload button and browsing for a photo. This process is repeated for each photo.

Normal display state

"Add Photos" allows either browsing for photos or simply dragging and dropping them into the target zone below.

Invitation to drag

The invitation is clear. By using the drop target area as an advertisement for the drag feature, the process is discoverable (as well as natural).

Dropped

Photos dropped are collected into an upload area. Pressing "Start Upload" starts the uploading process.

Completed

All items are marked Complete when finished.

Considerations

This is not a trivial implementation. But it does clearly illustrate the benefit of drag and drop for operating on a set of files. The traditional model requires each photo to be selected individually for upload. Drag and drop frees you to use whatever browsing method is available on your system and then drop those photos for upload.

Anti-pattern: Artificial Visual Construct

Unfortunately, drag and drop can sometimes drive the design of an interface instead of being an extension of a natural interface. These interactions are almost always doomed, as they are the tail wagging the proverbial dog. Rating movies, books, and music is a common feature found on many sites. But what happens if you try to use drag and drop to rate movies?

Drag and Drop Collection

 variation on dragging objects is collecting objects for purchase, bookmarking, or saving into a temporary area. This type of interaction is called Drag and Drop Collection. Drag and drop is a nice way to grab items of interest and save them to a list. The Laszlo shopping cart example illustrates this nicely.

Normal display state

The shopping cart is docked on the right part of the screen.

Invitation to drag

You can add to the cart with the "+ cart" button or you can drag the item to the shopping cart. If you use the button, the item flies to the cart; the cart bumps open and closed briefly to indicate that the item has been entered.

Dragging

The item gets a dragging treatment.

Dropped

The cart is populated with the new item.

DIRECT SELECTION

- **Toggle Selection:** Checkbox or control-based selection.
- **Collected Selection:** Selection that spans multiple pages.
- **Object Selection:** Direct object selection.
- **Hybrid Selection:** Combination of Toggle Selection and Object Selection

Toggle Selection

The most common form of selection on the Web is Toggle Selection. Checkboxes and toggle buttons are the familiar interface for selecting elements on most web pages.
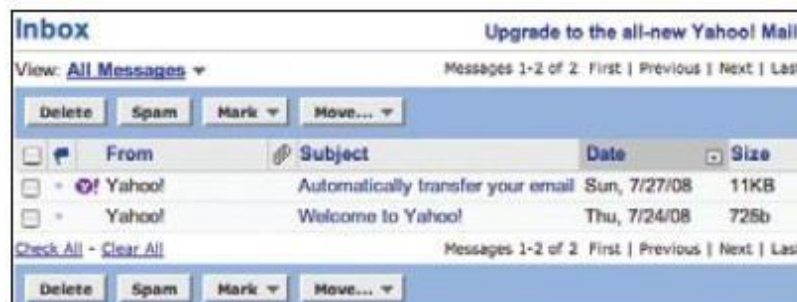
The way to select an individual mail message is through the row's checkbox. Clicking on the row itself does not select the message. We call this pattern of selection Toggle Selection since toggle-style controls are typically used for selecting items.

Once items have been check-selected, actions can be performed on them. Usually these actions are performed on the selection by clicking on a separate button (e.g., the Delete button). Gmail is a good example of actions in concert with Toggle Selection

Unselected state

Each mail message has a checkbox that controls whether it is selected or not.

Selected items

Two mail messages have been selected. In addition to the checkbox selection, the selected items are highlighted in light yellow.

Action triggered

Delete will operate on the selected items.

Action completed

The two selected email messages have been deleted.

Considerations

Toggle Selection with checkboxes has some nice attributes:

- Clear targeting, with no ambiguity about how to select the item or deselect it.
- Straightforward discontinuous selection, and no need to know about Shift or Control key ways to extend a selection. Just click the checkboxes in any order, either in a continuous or discontinuous manner.
- Clear indication of what has been selected.

Scrolling versus paging

The previous examples were with paged lists. But what about a scrolled list? Yahoo! Mail uses a scrolled list to show all of its mail messages (Figure). While not all messages are visible at a time, the user knows that scrolling through the list retains the currently selected items. Since the user understands that all the messages not visible are still on the same continuous pane, there is no confusion about what an action will operate on—it will affect all selected items in the list. Sometimes the need for clarity of selection will drive the choice between scrolling and paging.
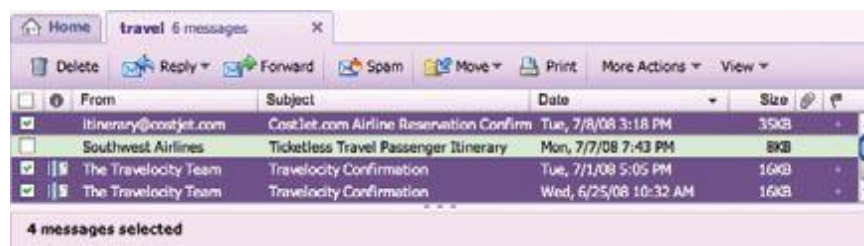
Making selection explicit

With Yahoo! Bookmarks you can manage your bookmarks by selecting bookmarked pages and then acting on them. The selection model is visually explicit

Collected Selection

Toggle Selection is great for showing a list of items on a single page. But what happens if you want to collect selected items across multiple pages? Collected Selection is a pattern for keeping track of selection as it spans multiple pages.

In Gmail, you can select items as you move from page to page. The selections are remembered for each page. If you select two items on page one, then move to page two and select three items, there are only three items selected. This is because actions only operate on a single page. This makes sense, as users do not normally expect selected items to be remembered across different pages.

Considerations

Gmail does provide a way to select all items across different pages. When selecting all items on a individual page (with the ―All‖ link), a prompt appears inviting the user to ―Select all 2785 conversations in Spam‖. Clicking that will select all items across all pages (Figure). The ―Delete Forever‖ action will operate on all 2785 conversations, not just the 25 selected on the page.

Keeping the selection visible

The real challenge for multi-page selection is finding a way to show selections gathered across multiple pages. You need a way to collect and show the selection as it is being created. Here is one way that Collected Selection comes into play. LinkedIn uses Collected Selection to add potential contacts to an invite list.

The list of potential invitees is shown in a paginated list on the lefthand side. Clicking the checkbox adds them to the invite list. The invite list becomes the place where selected contacts across multiple pages are remembered.

Collected Selection and actions

In the menu system it was hard to discern whether the user meant to operate on the selection (photos on the page could be selected through an Object Selection model) or on the collected items in the tray. To resolve this ambiguity, the drop-down menus contained two identical sets of commands. The first group of commands in the menu operated on the collected items in the tray. The second set of commands operated on the selected objects. Needless to say, this was confusing since it required the user to be fully aware of these two selection models when initiating a command.

One way to remove this ambiguity would have been to have a single set of commands that operated on either the tray or the photos—depending on which had the focus. This would require a way to select the tray and a way to deselect it (by clicking outside the tray). A possible approach would be to slightly dim the photo gallery when the tray is selected (causing it to clearly have the focus), and do the opposite when the tray is not the focus.

Object Selection

Object Selection, is when selection is made directly on objects within the interface. Sometimes using a checkbox does not fit in with the style of interaction desired. Laszlo's WebTop mail allows the user to select messages by clicking anywhere in the row

Nothing selected

Normal view when nothing is selected and the mouse is not over a message.

Hovered state

When the mouse hovers over a row, the row is subtly highlighted to indicate focus and what will be selected if the user clicks.
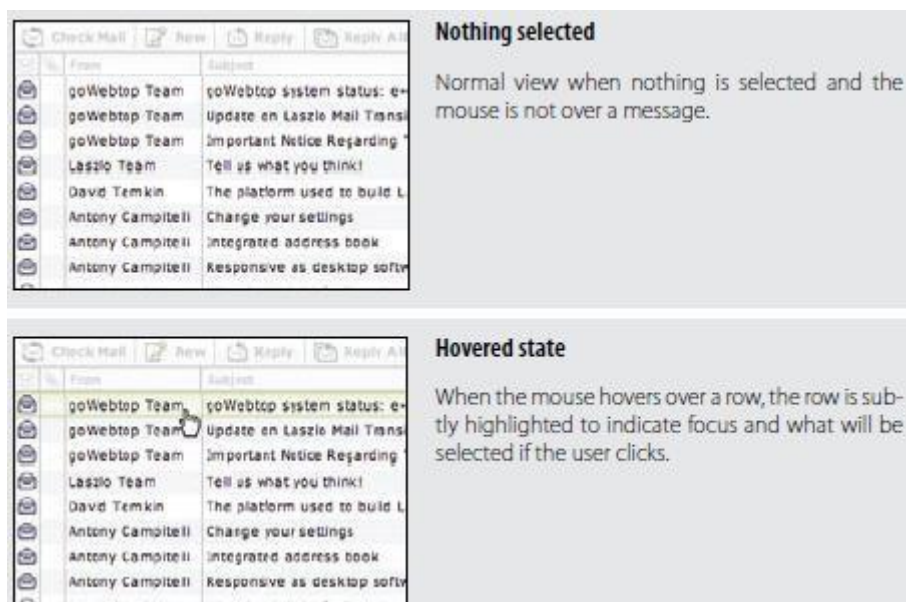
Considerations

Desktop applications tend to use Object Selection. It is also natural that web-based mail applications that mimic desktop interactions employ this same style of selection. Instead of showing a control (like a checkbox), the object itself can be selected and acted on directly. Object Selection can be extended by holding down the Shift key while clicking on a different item. The Command key (Macintosh) or Control key (Windows) can be used to individually add items in a discontinuous manner. The downside to this approach is that it is not obvious to use the modifier keys for extending the selection. Toggle Selection's use of toggle buttons makes the selection extension model completely obvious.

Desktop-style selection

For now Object Selection is not as common on the Web. Given that most sites have been content-oriented, there have been few objects to select. Also, with the Web's simple event model, Object Selection was not easy to implement. In typical web pages, keyboard events have rarely made sense since they are also shared with the browser.

Object Selection interactions include ways to use the mouse to drag-select objects. Yahoo! Photos introduced this same type of object selection to its photo gallery (Figure below). Individually clicking on a photo selects it. Using the Shift key and clicking also extends the selection. In addition, using the Control key and clicking discontinuously selects photos. And like most desktop applications, you can drag a selection box around a group of items to add them to the selected set (in this case, photos).

Figure: Yahoo! Photos 3.0 created a rich drag selection mechanism for selecting photos

Hybrid Selection

Mixing Toggle Selection and Object Selection in the same interface can lead to a confusing interface. Referring back to Yahoo! Bookmarks, you'll see an odd situation arise during drag and drop

Figure: In Yahoo! Bookmarks, one item is selected, but two items can be dragged by dragging on the unselected item.

Considerations

There are a few important issues to consider when using Hybrid Selection.

Confusing two models

One bookmark element is selected (notice the checkbox Toggle Selection). The second bookmark element (—Dr. Dobb's‖) is unselected (the checkbox is clear). In the right panel of clicking and dragging on the unselected bookmark element initiates a drag. The drag includes both the selected element and the unselected element. Since only one is shown as selected, this creates a confusing situation.

This occurs because three things are happening in the same space:

- Toggle Selection is used for selecting bookmarks for editing, deleting, etc.
- Object Selection is used for initiating a drag drop.
- Mouse click is used to open the bookmark on a separate page.

The problem is that more than one interaction idiom is applied to the same place on the same page. In this case, if you happen to try to drag, but instead click, you will be taken to a new page. And if you drag an unselected item, you now have two items selected for drag but only one shown as selected for other operations. This is definitely confusing. Simply selecting the item (automatically checking the box) when the drag starts would keep the selection model consistent in the interface. However, it might lead the user to expect a single click to also do the same (which it cannot since it opens the bookmark). So, mixing the two selection models together can be problematic. However, there is a way to integrate the Toggle Selection and Object Selection and have them coexist peacefully as well as create an improved user experience.

CONTEXTUAL TOOLS

Interaction in Context

Desktop applications separate functionality from data. Menu bars, toolbars, and palettes form islands of application functionality. Either the user chooses a tool to use on the data or makes a selection and then applies the tool. They were completely content-oriented. Rich tool sets were not needed for simply viewing and linking to content pages. Even in e-commerce sites like Amazon or eBay, the most functionality needed was the hyperlink and —Submit‖ button. Touch-based interfaces were the stuff of research labs and, more recently, interesting You- Tube videos. But now they're as close as our phones. Most notably, the Apple iPhone brought touch to the masses (Figure below). Gesture-based interfaces seemed even further out. Yet these became reality with the Nintendo Wii.

Figure: The Apple iPhone introduced touch-based interfaces to the consumer market.

Fitts's Law

Fitts's Law is an ergonomic principle that ties the size of a target and its contextual proximity to ease of use. Bruce Tognazzini restates it simply as:

The time to acquire a target is a function of the distance to and size of the target.

In other words, if a tool is close at hand and large enough to target, then we can improve the user's interaction. Putting tools in context makes for lightweight interaction.

Contextual Tools

Contextual Tools are the Web's version of the desktop's right-click menus. Instead of having to right-click to reveal a menu, we can reveal tools in context with the content. We can do this in a number of ways:

- **Always-Visible Tools:** Place Contextual Tools directly in the content.
- **Hover-Reveal Tools:** Show Contextual Tools on mouse hover.
- **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 Tools

The simplest version of Contextual Tools is to use Always-Visible Tools. Digg is an example of making Contextual Tools always visible
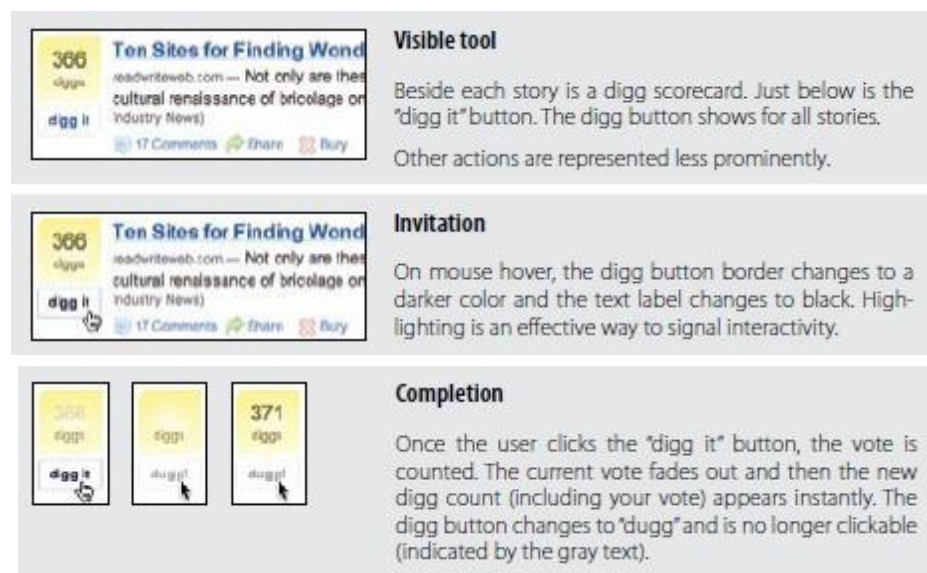

Figure: digg's —digg it‖ button is a simple Contextual Tool that is always visible

Considerations

The ‒digg it‖ button and Digg scorecard provide Always-Visible Tools next to each story. Clear call to action Why not hide the tools and only reveal them when the mouse is over the story? Since digging stories is central to the business of Digg, always showing the tool provides a clear call to action. There are other actions associated with news stories (comments, share, bury, etc.) but they are represented less prominently. In the case of Digg, the designers chose to show these at all times.

Relative importance

The ‒digg it‖ action is represented as a button and placed prominently in the context of the story. The ‒bury it‖ action is represented as a hyperlink along with other ‒minor‖ actions just below the story. The contrast of a button and a hyperlink as well as its placement gives a strong indication as to the relative importance of each action.

Discoverability

Discoverability is a primary reason to choose Always-Visible Tools. On the flip side, it can lead to more visual clutter. In the case of Digg and Netflix, there is a good deal of visual space given to each item (story, movie). But what happens when the items you want to act on are in a list or table?

Generally Contextual Tools in a list work well when the number of actions is kept to a minimum. Gmail provides a single Always-Visible Tool in its list of messages—the star rating—for flagging emails
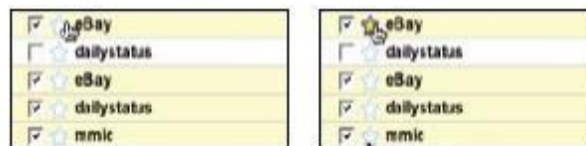
Fig: Google Mail uses Contextual tools to flag favourites.

Simply clicking the star flags the message as important. The unstarred state is rendered in a visually light manner, which minimizes the visual noise in the list.

Hover-Reveal Tools

One way to do this is to reveal the tools when the user pauses the mouse over an object. The Hover-Reveal Tools pattern is most clearly illustrated by 37 Signal‘s Backpackit (Figure below). To-do items may be deleted or edited directly in the interface. The tools to accomplish this are revealed on mouse hover.
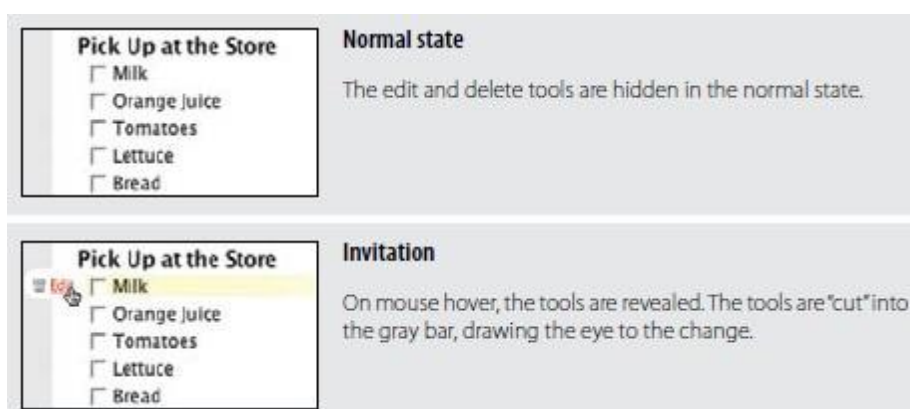
Discoverability

A serious design consideration for Hover-Reveal Tools is just how discoverable the additional functionality will be. while the Contextual Tools are revealed on hover, the checkbox is always visible for each to-do item. To check off an item, users have to move the mouse over it. When they do, they will discover the additional functionality.

Contextual Tools in an overlay

There are several actions available for a focused object. Instead of placing tools beside the object being acted on, the revealed tools can be placed in an overlay. However, there can be issues with showing contextual tools in an overlay:

1. Providing an overlay feels heavier. An overlay creates a slight contextual switch for the user's attention.

2. The overlay will usually cover other information—information that often provides context for the tools being offered.

3. Most implementations shift the content slightly between the normal view and the overlay view, causing the users to take a moment to adjust to the change.

4. The overlay may get in the way of navigation. Because an overlay hides at least part of the next item, it becomes harder to move the mouse through the content without stepping into a ‒landmine.‖

Toggle-Reveal Tools

A variation on the two previous approaches is to not show any Contextual Tools until a special mode is set on the page. A good example of Toggle-Reveal Tools is in Basecamp's category editing

Not visible

Each category is listed in this section. The "Edit" link at the top is the way to edit the category section.

Visible in edit mode

Each category gets a "Rename" link and where appropriate a trashcan is displayed (for empty categories that may be deleted).

This is a "soft" mode, since the user can ignore the additional tools and choose to do something different on the page.

Considerations

Here are a few considerations to keep in mind when using Toggle-Reveal Tools.

Soft mode

Generally, it is a good thing to avoid specific modes in an interface. However, if a mode is soft it is usually acceptable. By ―soft‖ we mean the user is not trapped in the mode. With Basecamp, the user can choose to ignore the tools turned on. It just adds visual noise and does not restrict the user from doing other actions. This is a nice way to keep the interaction lightweight.

When would you use this technique? When the actions are not the main thing and you want to reduce visual noise. This fits the category example perfectly. Items are renamed or deleted occasionally. It is common, however, to want to click through and see the contents of a category (the category is always hyperlinked). Hence, make it readable and easily navigable in the normal case—but still give the user a way to manage the items in context.

Google Reader could potentially be improved in this manner. In the current interface, clicking ‒Manage Subscriptions‖ takes the user to another page to edit subscriptions. One possible change is the addition of an ‒edit‖ button that toggles in a set of context tools for each subscription (Figure below). This would allow the user to rename and unsubscribe without leaving the context of the reading pane.

Multi-Level Tools

Contextual Tools can be revealed progressively with Multi-Level Tools. Songza* provides a set of tools that get revealed after a user clicks on a song. Additional tools are revealed when hovering over the newly visible tools

Normal state

The tools are not visible normally. Mouse hover just highlights the song—it does not reveal the **Contextual Tools.**

Click activation

On mouse click, a cloverleaf-style menu is shown with the four basic functions: play, rate, add, and share.
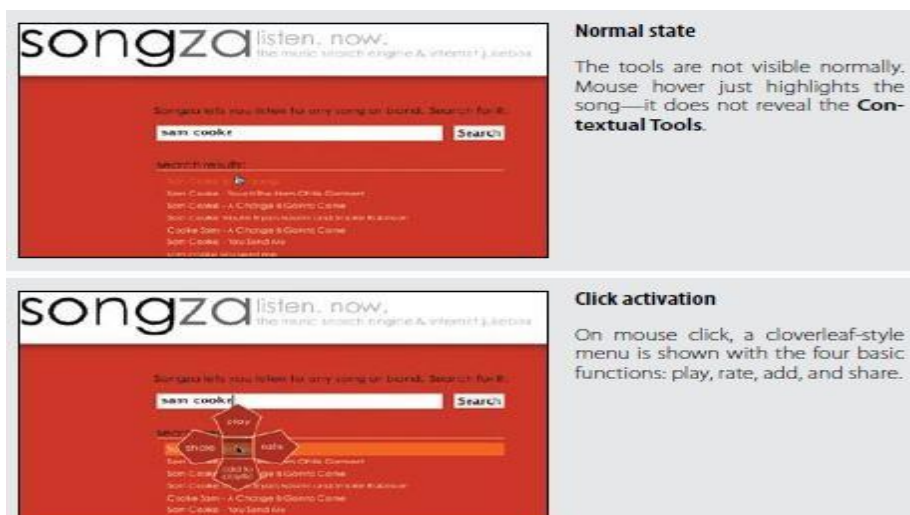
Fig: Songza uses a multi-level contextual tool menu

Radial menus

Radial menus* such as in Songza have been shown to have some advantages over more traditional menus. First, experienced users can rely on muscle memory rather than having to look directly at the menu items. Second, the proximity and targeting size make the menu easy to navigate since the revealed menu items are all equally close at hand (recall Fitts's Law).

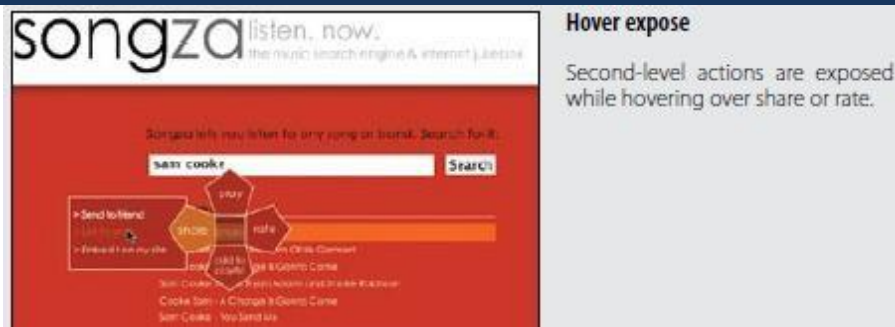The one potential downside to this approach is that rating a song requires several steps: an initial click on the song, moving the mouse over the ‒rate‖ menu item, then clicking either the thumbs up or thumbs down option. If rating songs was an important activity, the extra effort might prevent some users from doing so. An alternate approach would be to replace ‒rate‖ directly with the thumbs up and the thumbs down options.

Activation

Another interesting decision Songza made was to not activate the radial menu on hover. Instead, the user must click on a song to reveal the menu. Activating on click makes the user intent more explicit. Making activation more explicit avoids the issues described earlier in the Hover and Cover anti-pattern. The user has chosen to interact with the song. Conversely, with a mouse hover, it's never quite clear if the user meant to activate the menu or just happened to pause over a song title.

Default action

Playing a song requires moving to the top leaf. One possible solution would be to place the ‒play‖ option in the middle of the menu (at the stem) instead of in one of the leaves. Clicking once would activate the menu. Clicking a second time (without moving the mouse) would start playing the song. This interaction is very similar to one commonly used in desktop applications: allowing a double-click to activate the first item (default action) in a right-click menu.

Contextual toolbar

Picnik is an online photo-editing tool that integrates with services like Flickr. In all, there are six sets of tools, each with a wide range of palette choices. Picnik uses Multiple-Level Tools to expose additional functionality. By wrapping the photo with tools in context and progressively revealing the levels of each tool, Picnik makes editing straightforward

Figure: Picnik wraps layers of Contextual Tools around the image being edited

Muttons

Multi-Level Tools is the ‒mutton‖ (menu + button = mutton). Muttons are useful when there are multiple actions and we want one of the actions to be the default.
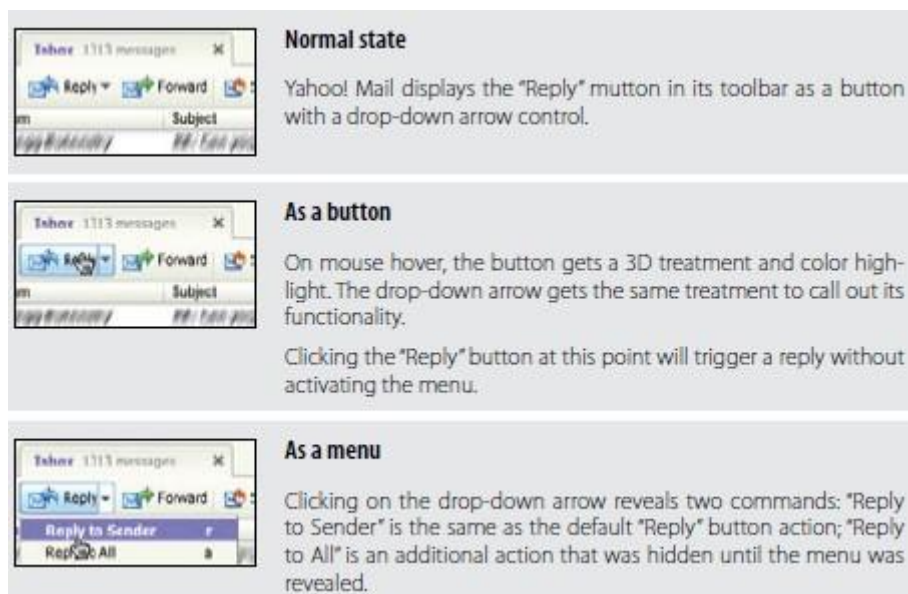
Figure : Yahoo! Mail‗s ‒Reply‖ button looks like a drop-down when hovered over; clicking ‒Reply‖ replies to sender, and clicking the drop-down offers the default action as well as ‒Reply to All‖

Clicking ‒Reply‖ performs the individual reply. To reply to all, the menu has to be activated by clicking on the drop-down arrow to show the menu. 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

If muttons are not implemented correctly, they can be problematic for those using accessibility technologies. Because an earlier version of Yahoo! Mail did not make the mutton keyboard accessible, Yahoo!‗s accessibility guru, Victor Tsaran, was convinced that there was no ‒Reply to All‖ command in the Yahoo! Mail interface.

Secondary Menu

Desktop applications have provided Contextual Tools for a long time in the form of Secondary Menus. These menus have been rare on the Web. Google Maps uses a secondary menu that is activated by a right-click on a route. It shows additional route commands

Normal view of route

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

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".

Considerations

Secondary Menus have not been common in web applications.

Conflict with browser menu

One problem is the browser inserts its own right-click menu. Replacing the menu in normal content areas can confuse users, as they may not know if the standard browser menu or the application-specific menu will be shown. It will depend on whether it is clear that an object exists in the interface (as in the route line above), and if the menu is styled differently enough to disambiguate the menus.

Discoverability

As a general rule, never put anything in the Secondary Menu that can't be accomplished elsewhere. Secondary Menus are generally less discoverable. More advanced items or shortcuts, however, can be placed in the Secondary Menu as an alternate way to accomplish the same task.

Accessibility

Right-click is not the only way to activate a Secondary Menu. You can activate the menu by holding down the mouse for about one second. This provides a more accessible approach to popping up a Secondary Menu. This technique is used in the Macintosh Dock.

Clicking and holding down on an application in the dock will reveal the Secondary Menu without requiring a right-click activation.

Acting on multiple objects

Keep in mind that all of the other Contextual Tools presented in this chapter have a limitation on the number of items they can operate on. Always-Visible Tools, Hover-Reveal Tools, Toggle-Reveal Tools, and Multi-Level Tools all operate on a single item at a time (even Toggle-Reveal Tools just shows a tool per item). Secondary Menus are different.

OVERLAYS

Overlays are really just lightweight pop ups. We use the term lightweight to make a clear distinction between it and the normal idea of a browser pop up. Browser pop ups are created as a new browser window (Figure below). Lightweight overlays are shown within the browser page as an overlay (Figure below). Older style browser pop ups are undesirable because:

Browser pop ups display a new browser window.

As a result these windows often take time and a sizeable chunk of system resources to create.

Browser pop ups often display browser interface controls (e.g., a URL bar). Due to security concerns, in Internet Explorer 7 the URL bar is a permanent fixture on any browser pop-up window.

Figure : If Orbitz used a browser pop-up window for its calendar chooser (it does not), this is how it might look.

By using either Flash or Ajax-style techniques (Dynamic HTML), a web application can present a pop up in a lightweight overlay within the page itself. This has distinct advantages:

- Lightweight overlays are just a lightweight in-page object. They are inexpensive to create and fast to display.
- The interface for lightweight overlays is controlled by the • web application and not the browser.
- There is complete control over the visual style for the overlay. This allows the overlay to be more visually integrated into the application's interface

Figure : Orbitz uses a lightweight DHTML overlay for its calendar chooser; since it does not require the overhead of a separate browser window, it can pop up quickly and is better integrated into the page visually

Dialog Overlay

Dialog Overlays replace the old style browser pop ups. Netflix provides a clear example of a very simple Dialog Overlay. In the ―previously viewed movies for sale‖ section, a user can click on a ―Buy‖ button to purchase a DVD. Since the customer purchasing the DVD is a member of Netflix, all the pertinent shipping and purchasing information is already on record. The complete checkout experience can be provided in a single overlay.

Considerations

Because the overlay is a lightweight pop up, the confirmation can be displayed more rapidly and the application has complete control over its look and placement.

Lightbox Effect

One technique employed here is the use of a Lightbox Effect. In photography a lightbox provides a backlit area to view slides. On the Web, this technique has come to mean bringing something into view by making it brighter than the background. In practice, this is done by dimming down the background.

The Lightbox Effect is useful when the Dialog Overlay contains important information that the user should not ignore. Both the Netflix Purchase dialog and the Flickr Rotate dialog are good candidates for the Lightbox Effect. If the overlay contains optional information, then the Lightbox Effect is overkill and should not be used.

Modality

Overlays can be modal* or non-modal. A modal overlay requires the user to interact with it before she can return to the application

The Lightbox Effect emphasizes that we are in a separate mode. As a consequence, it is not needed for most non-modal overlays. As an example, refer back to the Orbitz calendar pop up. Since the overlay is really more like an in-page widget, it would not be appropriate to make the chooser feel heavier by using a Lightbox Effect.

Staying in the flow

Overlays are a good way to avoid sending a user to a new page. This allows the user to stay within the context of the original page. However, since overlays are quick to display and inexpensive to produce, sometimes they can be tempting to use too freely, and in the process, may actually break the user's flow.

Detail Overlay

The Detail Overlay allows an overlay to present additional information when the user clicks or hovers over a link or section of content. Toolkits now make it easier to create overlays across different browsers and to request additional information from the server without refreshing the page.

Box shots

In the more recent versions of the Netflix site, large box shots are employed without synopsis text. Box shots convey a lot of information.

ri.M Assistant Professor Department of computer science and

Detail overlay activation

However, often more information is needed to decide whether a movie should be played or added to a movie queue.

By providing a synopsis along with personalized recommendation information, the user can quickly make a determination.

The movie detail information is displayed after a slight delay.

Detail overlay deactivation

Moving the mouse outside the box shot immediately removes the movie detail information.

Activation

The overlay is displayed when the mouse hovers over a box shot. There is about a halfsecond delay after the user pauses over a movie. The delay on activation prevents users from accidentally activating the overlay as they move the cursor around the screen. Once the user moves the cursor outside the box shot, the overlay is removed immediately. Removing it quickly gives the user a fast way to dismiss it without having to look for a ―Close‖ box.

Anti-pattern: Mouse Traps

It is important to avoid activating the Detail Overlay too easily. We have seen usability studies that removed the delay in activation, and users reported that the interface was ―too noisy‖ and ―felt like a trap‖. We label this anti-pattern the Mouse Trap.

The reasoning for this is not clear, but Amazon uses the Mouse Trap anti-pattern in one of its ―associate widgets‖. Original Motion Picture Soundtrack‖ activates an overlay providing information on the soundtrack and a purchase option

Input Overlay

Input Overlay is a lightweight overlay that brings additional input information for each field tabbed into. American Express uses this technique in its registration for premium cards such as its gold card
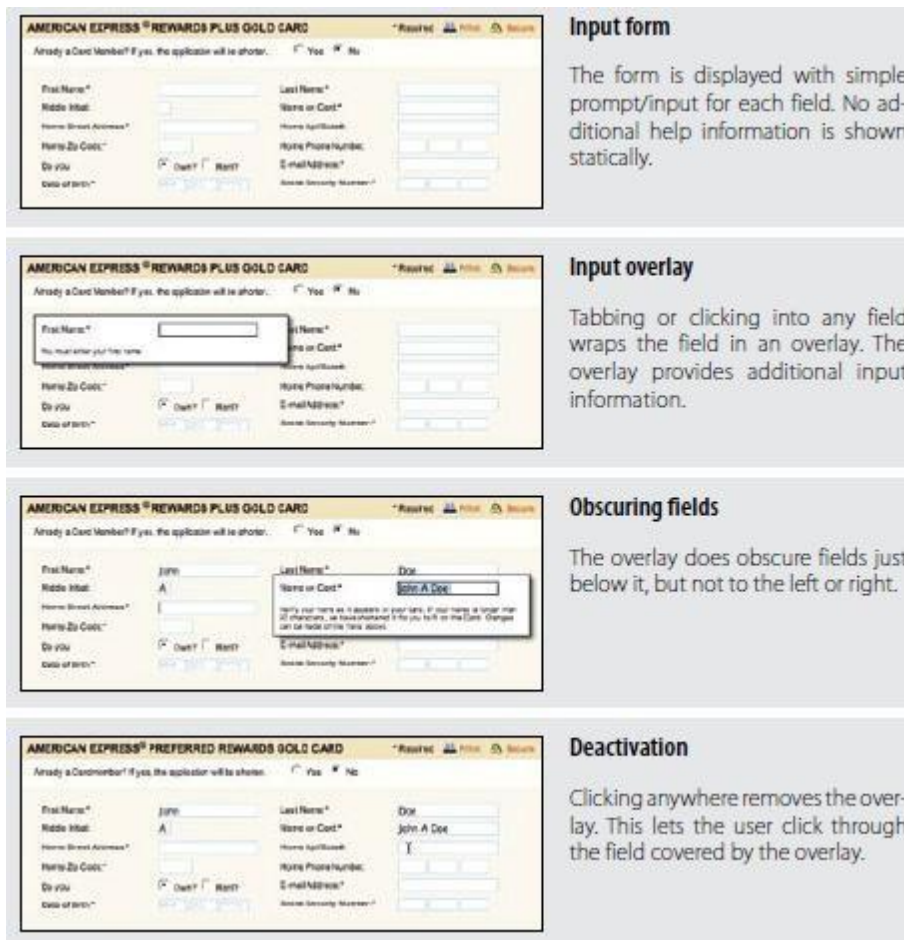
Figure 5-15. American Express provides Input Overlays to guide the user through the signup process

Considerations

There are a few things to keep in mind when using Input Overlays.

Clear focus

The overlay contains additional input help information. This allows the normal display of the form to be displayed in a visually simple manner (just prompts and inputs). The overlay creates focus on the given input field. Instead of seeing an ocean of inputs, the user is focused on just entering one field.

Display versus editing

when the Input Overlay is shown, the prompt is displayed in exactly the same manner as when the overlay doesn't show. This is critical, as it makes the overlay feel even more lightweight. If the overlay prompt were bold, for example, the change would be slightly distracting and take the focus away from input. The only difference between the non-overlay field and the overlay version is a slightly thicker input field border. This draws the eye to the task at hand—input.

- Field traversal
- Tab navigation
- One-click deactivation

INLAYS

Information, or dialog with the user needs to be an overlay. Another approach is to inlay the information directly within the page itself. To distinguish from the pop-up overlay, we call these in-page panels Inlays.

Dialog Inlay

A simple technique is to expand a part of the page, revealing a dialog area within the page. The BBC recently began experimenting with using a Dialog Inlay as a way to reveal customization controls for its home page

Activation

The "Customize homepage" button activates the customization inlay.

Customization inlay (slide)

The inlay slides into view.

Customization inlay

The additional customization controls for the BBC home page are shown directly in context with the rest of the page.

Azh gnini Assistant Professor Department of computer science and

Considerations

Of course an overlay could have been used instead. However, the problem with overlays is that no matter where they get placed, they will end up hiding information. Inlays get around this problem by inserting themselves directly into the context of the page.

In context

This Dialog Inlay is similar to a drawer opening with a tray of tools. Instead of being taken to a separate page to customize the home page appearance, the user can make changes and view the effects directly. The advantage is the ability to tweak the page while viewing the actual page.

List Inlay

Lists are a great place to use Inlays. Instead of requiring the user to navigate to a new page for an item's detail or popping up the information in an Overlay, the information can be shown with a List Inlay in context. The List Inlay works as an effective way to hide detail until needed—while at the same time preserving space on the page for high-level overview information.

Google Reader provides an expanded view and a list view for unread blog articles. In the list view, an individual article can be expanded in place as a List Inlay

Google Reader list view

In list view, articles are shown as a list of blog article titles.

Inlay list

Clicking on a single article expands it in place, in context with the rest of the list.
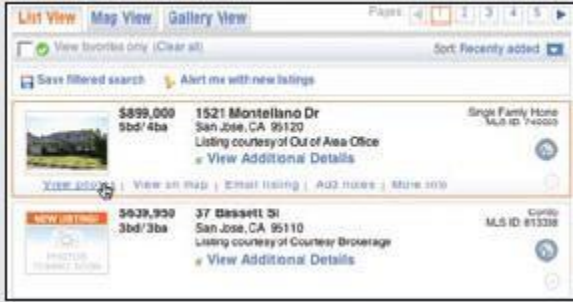
Detail Inlay

A common idiom is to provide additional detail about items shown on a page. We saw his with the example of the Netflix movie detail pop up in . Hovering over a movie revealed a Detail Overlay calling out the back-of-the-box information. Details can be shown inline as well. Roost allows house photos to be viewed in-context for a real estate listing with a Detail Inlay
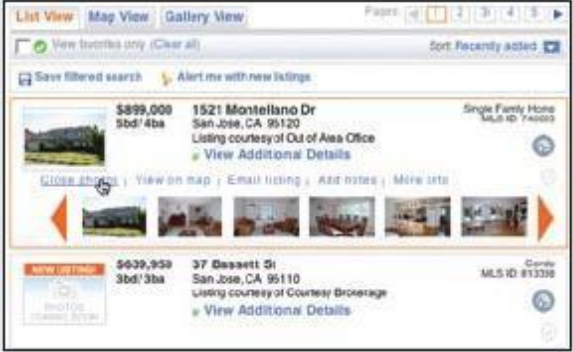

In-context tools

Hovering over a real estate listing brings in a set of in-context tools, including the "View photos" tool.

House photos inlay

Clicking on the "View photos" link expands the real estate item to include a carousel of house photos.

Detail overlay

The **Detail Inlay** contains thumbnails of house photos. Clicking on an individual thumbnail pops up a **Detail Overlay** with a larger photo of the house.

Combining inlays and overlays

Roost's solution was to combine several patterns. First, it uses the Hover Reveal, a Contextual Tools pattern, to reveal a set of tools when the user hovers over a listing. Second, it uses the Detail Inlay pattern to show a carousel of photos when the user clicks on the ‒View photos‖ link. And finally, it uses a Detail Overlay to blow up a thumbnail when clicked on.

VIRTUAL PAGES

Patterns that support virtual pages include:

- Virtual Scrolling
- Inline Paging
- Scrolled Paging
- Panning
- Zoomable User Interface

Virtual Scrolling

The traditional Web is defined by the ‒page.‖ In practically every implementation of websites (for about the first 10 years of the Web's existence) pagination was the key way to get to additional content. Of course, websites could preload data and allow the user to scroll through it. However, this process led to long delays in loading the page. So most sites kept it simple: go fetch 10 items and display them as a page and let the user request the next page of content. Each fetch resulted in a page refresh.

The classic example of this is Google Search. Each page shows 10 results. Moving through the content uses the now-famous Google pagination control Another approach is to remove the artificial page boundaries created by paginating the data with Virtual Scrolling.

Loading status

There are a few downsides to the Yahoo! Mail version of Virtual Scrolling. First, if the loading is slow, it spoils the illusion that the data is continuous. Second, since the scrollbar does not give any indication of where users are located in the data, they have to guess how far down to scroll. A remedy would be to apply a constantly updating status while the user is scrolling.

Progressive loading

Microsoft has applied Virtual Scrolling to its image search. However, it implements it in a different manner than Yahoo! Mail. Instead of all content being virtually loaded (and the scrollbar reflecting this), the scrollbar reflects what has been loaded.

Inline Paging

Switching the content in and leaving the rest of the page stable, we can create an Inline Paging experience.

In-page update

Keeping the context stable creates a better flow experience. With Inline Paging it feels like the user never leaves the page even though new virtual pages of results are being brought into view.

Natural chunking

Inline Paging can also be useful when reading news content online. The International Herald Tribune applied this as a way to page through an article while keeping the surrounding context visible at all times

Back button

The biggest issue with Inline Paging is whether the back button works correctly. One criticism of Endless.com is that if the user pages through search results and then hits the back button, it jumps to the page just before the search. This unexpected result could be fixed by making the back button respect the virtual pages as well. This is the way Gmail handles the back button.* Clicking back moves you through the virtual pages.

Interactive content loading

The iPhone employs inline paging when displaying search results in the iTunes store

Scrolled Paging: Carousel

The Carousel pattern takes this approach. A Carousel provides a way to page-in more data by scrolling it into view. On one hand it is a variation on the Virtual Scrolling pattern. In other ways it is like Virtual Paging since most carousels have paging controls. The additional effect is to animate the scrolled content into view.

Time-based

Carousels work well for time-based content. Flickr employs a Carousel to let users navigate back and forth through their photo collection

Animation direction

Inexplicably, AMCtheatres.com animates its Carousel the opposite way. This leads to a confusing experience, and it's harder to know which control to click

Virtual Panning

A great place for Virtual Panning is on a map. Google Maps allows you to pan in any direction by clicking the mouse down and dragging the map around

Start drag

Clicking and holding down changes the cursor into a hand (signifying panning).

Dragging

Dragging the canvas moves the map in real time.

Zoomable User Interface

A Zoomable User Interface (ZUI) is another way to create a virtual canvas. Unlike panning or flicking through a flat, two-dimensional space, a ZUI allows the user to also zoom in to elements on the page. This freedom of motion in both 2D and 3D supports the concept of an infinite interface.

Practically speaking, ZUIs have rarely been available in everyday software applications, much less on the Web. But with more advanced features added to Flash and the advent of Silverlight, this type of interface is starting to emerge and may be commonplace in the not-too-distant future.

Paging Versus Scrolling

Leading web designers and companies have taken different approaches to solving the same problems. Yahoo! Mail chose Virtual Scrolling. Gmail chose Inline Paging. How do you choose between paging and scrolling? While there are no hard and fast rules, here are some things to consider when making the decision:

• When the data feels ‒more owned‖ by the user—in other words, the data is not transient but something users want to interact with in various ways. If they want to sort it, filter it, and so on, consider Virtual Scrolling (as in Yahoo! Mail).

• When the data is more transient (as in search results) and will get less and less relevant the further users go in the data, Inline Paging works well (as with the iPhone).

• For transient data, if you don't care about jumping around in the data to specific sections, consider using Virtual Scrolling (as in Live Image Search).

• If you are concerned about scalability and performance, paging is usually the best choice. Originally Microsoft's Live Web Search also provided a scrollbar. However, the scrollbar increased server-load considerably since users are more likely to scroll than page.

• If the content is really continuous, scrolling is more natural than paging.

• If you get your revenue by page impressions, scrolling may not be an option for your business model.

• If paging causes actions for the content to become cumbersome, move to a scrolling model. This is an issue in Gmail. The user can only operate on the current page.

Changing items across page boundaries is unexpected. Changing items in a continuous scrolled list is intuitive.

PROCESS FLOW

Process FlowGoogle Blogger

The popular site Google Blogger generally makes it easy to create and publish blogs. One thing it does not make easy, though, is deleting comments that others may leave on your blog. This is especially difficult when you are the victim of hundreds of spam comments left by nefarious companies hoping to increase their search ranking.

1. Scroll to find the offending comment.
2. Click the trash icon to delete the comment.
3. After page refreshes, click the ―Remove Forever‖ checkbox.
4. Click the ―Delete Comment‖ button.
5. After the page refreshes, click the link to return to my blog article.
   Repeat steps 1–5 for each article with spam comments.

The Magic Principle

Alan Cooper discusses a wonderful technique for getting away from a technology-driven approach and discovering the underlying mental model of the user. He calls it the ―magic principle.‖* Ask the question, ―What if when trying to complete a task the user could invoke some magic?‖ For example, let's look at the problem of taking and sharing photos.

The process for this task breaks down like this:

- Take pictures with a digital camera.
- Sometime later, upload the photos to a photo site like Flickr. This involves:
- ➢ Finding the cable.
- ➢ Starting iTunes.
- ➢ Importing all photos.
- ➢ Using a second program, such as Flickr Uploadr, to upload the photos to Flickr.
- ➢ Copying the link for a Flickr set (which involves first locating the page for the uploaded set).
- Send the link in email to appropriate friends.

If some magic were invoked, here is how it might happen:

- The camera would be event-aware. It would know that is your daughter's eighth birthday.
- When finished taking pictures of the event, the camera would upload the pictures to Flickr.
- Flickr would notify family and friends that the pictures of the birthday party are available.

Thinking along these lines gets some of the artifacts out of the way. Of course the magic could be taken to the extreme: just eliminate the camera altogether! But by leaving some

elements in the equation, the potentially unnecessary technology pieces can be exposed. How about the cable? What if the camera could talk magically to the computer?

Process Flow patterns:

- Interactive Single-Page Process
- Inline Assistant Process
- Configurator Process
- Overlay Process
- Static Single-Page Process

Interactive Single-Page Process

Consumer products come in a variety of shapes, sizes, textures, colors, etc. Online shoppers will not only have to decide that they want shoes, but do they want blue suede shoes? And what size and width do they want them in? In the end the selection is constrained by the available inventory. As the user makes decisions, the set of choices gets more and more limited.

This type of product selection is typically handled with a multi-page workflow. On one page, the user selects a shirt and its color and size. After submitting the choice, a new page is displayed. Only when the user arrives at this second page does he find out that the ‒true navy‖ shirt is not available in the medium size. The Gap accomplishes this kind of product selection in a single page using Interactive Single-Page Process.

Benefits

Adobe calls out the Broadmoor one-page reservation interface in its Adobe Showcase.

* It states the benefits of this method:

- Reduces entire reservation process to a single screen.
- Reduces the number of screens in the online reservation process from five to one.

Other online reservation applications average 5 to 10 screens.

- Seventy-five percent of users choose OneScreen in favor of the HTML version.
- Allows users to vary purchase parameters at will and immediately view results.
- Reduces the time it takes to make a reservation from at least three minutes to less than one.

Additionally, Adobe notes that conversion rates (users who make it through the reservation process) are much higher with the Interactive Single-Page Process.

Inline Assistant Process

The Gap employed an Inline Assistant Process pattern for its shopping cart when it re-launched its site a few years back

Dialog Overlay Process

Dialog Overlay Process to encapsulate a multi-step flow inside a Dialog Overlay.

Configurator Process

Process Flow is meant to invoke delight. In these cases, it is the engagement factor that becomes most important. This is true with various Configurator Process interfaces on the Web. We can see this especially at play with car configurators. Porsche provides a configurator that allows users to build their own Porsche

Static Single-Page Process

Put the complete flow on one page in a Static Single-Page Process. The user sees all the tasks needed to complete the full process. This can be both good and bad. Seeing just one step to complete the process can encourage users to finish the task. But if the single step seems too long or too confusing, the user will most likely bail out of the process early. In other words, if placing all the tasks on a single page is enough to cause the user to bail out, it is not a good idea. In the case of the Apple store, each item is optionally set, and it's just a single click to include or exclude an item from the purchase.