## UNIT 5 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 scalability, 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 somefields pre-chosen or filled out.

## 5.1 DRAG AND DROP

### 5.1.1 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 beused 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 dropinteraction:

**Page Load:** Before any interaction occurs, you can pre-signify the availability of drag anddrop. 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 orobject 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 droppingthe object. Do you care? Will you want additional feedback as to why it is not a validtarget?

**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 youzoom 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 MomentsGrid

That_s 15 events times 6 actors. That means there are 90 possible interesting moment search requiring a decision involving an almost unlimited number of style and timing choices. You can pull all this together into a simple interesting moment‘s 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 |

### 5.1.3 Drag and Drop Module

One of the most useful purposes of drag and drop is to allow the user to directlyplace objects where she wants them on the page. A typical pattern is Drag and DropModules on a page. Netvibes provides a good example of this interaction pattern **Considerations**

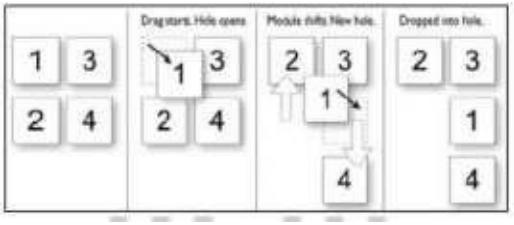
Netvibes allows its modules to be rearranged with drag and drop. A number of interesting

moments decide the specific interaction style for this site. Figure shows the interesting moment‘s grid for Netvibes. 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 theplaceholder 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 modulewill 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 arounda 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 looksimilar. 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 themodule 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 iGoogletake 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.

**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?

### 5.1.4 Drag and Drop List

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

**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 immediatelysee 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. Youcan also use an insertion bar within a list to indicate where a dropped item will land. Netflixuses an insertion target when movies are dragged to a new location in a user_s movie queueThe 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. Clickthe —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 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 dragand drop. After the move to top completes, a simple tip could appear to invite users to dragand 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 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 fororganizational charts. Drag and Drop Object is used to rearrange.

**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.

### 5.1.6 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

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.

**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 insteadof 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?

### 5.1.7 Drag and Drop Collection

A 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. Drag and drop is a natural way collect items for purchase. It mimics the shopping experience in the real world. Grab an item. Drop it in your basket. This is fast and convenient once you know about the feature. However, as a general rule, you should never rely solely.

**The Challenges of Drag and Drop**

As you can see from the discussion in this chapter, Drag and Drop is complex. There are four broad areas where Drag and Drop may be employed: Drag and Drop Module, Drag andDrop List, Drag and Drop Object, and Drag and Drop Action. And in each area, there are a large number of interesting moments that may be handled in numerous ways. Being consistent in visual and interaction styles across all of these moments for all of these types of interactions is a challenge in itself. And keeping the user informed throughout the process with just the right amount of hints requires design finesse.

## 5.2 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.

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.

**5.2.2 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


**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).

**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. Gesture-based interfaces seemed even further out. Yet these became reality with the Nintendo Wii.

### 5.3.2 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.

### 5.3.3 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).

### 5.3.4 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


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 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.

### 5.3.5 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.

### 5.3.6 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


**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 Base camp, 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. This would allow the user to rename and unsubscribe without leaving the context of the reading pane.

### 5.3.7 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

**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 (recallFitts_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.

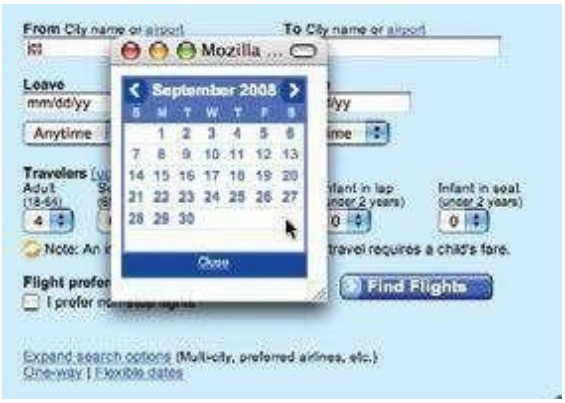
**Default action**

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 application: allowing a double-click to activate the first item (default action) in a right-click menu.

## 5.4 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 Lightweight overlays are shown within the browser page as an overlay. 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.




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


We will look at three specific types of overlays: Dialog Overlays, Detail Overlays, and Input Overlays.

### 5.4.1 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

### 5.4.2 Detail Overlay

The second type of overlay is somewhat new to web applications. 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. Taking another example from Netflix, information about a specific movie is displayed as the user hovers over the movie‘s box shot.

### 5.4.3 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 card such as its gold card.

## 5.5 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.

### 5.5.1 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.

### 5.5.2 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

### 5.5.3 Detail Inlay

A common idiom is to provide additional detail about items shown on a page. We saw this with the example of the Netflix movie detail pop up . 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

**Inlay Versus Overlay?**

Use an overlay when there may be more than one place a • dialog can be activated from (the exception may be showing details for items in a list).

- Use an overlay to interrupt the process.
- Use an overlay if there is a multi-step process.
- Use an inlay when you are trying to avoid covering information on the page needed in the dialog.

  Use an inlay for contextual information or details about one of many items (as in a list): a typical example is expanding list items to show detail.

## 5.6 Virtual pages

Overlays allow you to bring additional interactions or content in a layer above the current page. Inlays allow you to do this within the page itself. However, another powerful approach to keeping users engaged on the current page is to create a virtual page. That is to say, we create the illusion of a larger virtual page.

Patterns that support virtual pages include:

- Virtual Scrolling
- Inline Paging
- Scrolled Paging

- Panning
- Zoomable User Interface

### 5.6.1 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

Virtual Scrolling demonstrate three different ways to manage the virtual space:

- Yahoo! Mail creates the illusion that all data has been loaded up-front by having the scrollbar reflect the total virtual space.
- Microsoft Live Search creates the virtual illusion as the user moves down throughthe search results.

  And PicLens does the same with the caveat that it • shows a virtual window in the larger virtual space (by only providing a scroller control for where the user is and some before and after context).

### 5.6.2 Inline Paging

What if instead of scrolling through content we just wanted to make pagination feel less like a page switch? By only switching the content in and leaving the rest of the page stable, we can create an Inline Paging experience. This is what Amazon‘s Endless.com site does with its search results.

### 5.6.3 Scrolled Paging

Besides Virtual Scrolling and Virtual Paging, there is another option. You can combine both scrolling and paging into Scrolled Paging. Paging is performed as normal. But instead the content is –scrolled|| into view.

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. Yahoo! Underground uses a Carousel to provide a way to page/scroll through articles.

### 5.6.4 Virtual Panning

One way to create a virtual canvas is to allow users the freedom to roam in two-dimensional space. 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

### 5.6.5 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.

### 5.6.6 Paging Vs 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.

## 5.7 Process flow

It has long been common practice on the Web to turn each step into a separate page. While this may be the simplest way break down the problem, it may not lead to the best solution. For some Process Flows it makes sense to keep the user on the same page throughout the process.

### 5.7.1 Google 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. Blogger forces you to delete these comments through a three-step process. Each step is an individual page, all punctuated with a page refresh.

My (Bill's) blog site was recently spammed. It turns out that my 100 or so articles all had 4 or more spam comments. That means that I had to delete more than 400 spam comments.

Given the way Google Blogger implemented comment deleting, I had to follow these steps for each comment on each blog article:

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.
6. Repeat steps 1–5 for each article with spam comments.

It took 1,600 clicks, 1,200 page refreshes, 400 scroll operations, and several hours to finally rid myself of all of the spam comments. If the delete action could have been completed in the same page as the comments, that would have eliminated hundreds of clicks and well over a thousand page refreshes, and scrolling would have been all but eliminated. I would not have wasted all the mental energy to reorient myself after each page transition. And I would have been a much happier man. This is a common interaction flow on the Web. It turns out to be simpler to design and implement a process as a series of pages rather than a single interactive space.

### 5.7.2 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.

### 5.7.3 Interactive single page processor

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.

### 5.7.4 Inline assistant process

Another common place where multiple pages are used to complete a process is when adding items to a shopping cart. As mentioned earlier, Amazon provides the typical

experience So what magic can we apply to move this from a multi-page experience to a single-page experience? Instead of thinking about the cart as a process, we can think aboutit as a real-world object. Given this mindset, the cart can be realized in the interface as an object and be made available on the page. The Gap employed an Inline Assistant Process pattern for its shopping cart when it re-launched its site a few years back.

### 5.7.5 Dialog overlay process

As mentioned before, any page switch is an interruption to the user‘s mental flow. In addition, any context switch is a chance for a user to leave the site. We seek an experience that has as little mental friction as possible. But sometimes the step-by-step flow is necessary. The Netflix approach just described uses a Dialog Overlay Process to encapsulate a multi- step flow inside a Dialog Overlay.. Overlays allow us to keep the context of the page yet present a virtual space to conduct a conversation with the user. Discover.com recently expanded its account section with a more detailed profile. The profile captures things like your payment date, mobile fraud alerts, paperless statements, and general contact information .The overlay pops up when you first enter your account.

### 5.7.6 Configuration process

Sometimes a 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 Being able to tweak the colors, wheels, interior, and options for a car and see the results in real time is an engaging experience. Live Previews allow the user to see the effect of his changes on a simulated version of the real thing.

### 5.7.7 Static single page process

The Apple example illustrates another way to get rid of multiple pages in a Process Flow. Just 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 justone step to complete the process can encourage users to finish the task. But if the single stepseems 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 asingle click to include or exclude an item from the purchase. EBay provides two ways tosell an item. An introductory panel gathers the description of the item for sale. The Customize your listings…‖ option takes the user through a traditional multi-page process.