Drag and Drop

Module 3.2

Drag and Drop

- One of the great innovations by Macintosh
- Influenced by the graphical user interface work on Xerox PARC's Star Information System
- HalfBrain, launched a web-based presentation application, BrainMatter, which was written in DHTML and used drag and drop as an integral part of its interface

- Drag and drop showed up again with, “Oddpost”, a web-based mail application that allowed users to drag and drop messages between folders.

The Oddpost web mail client performed like a desktop mail application and included drag and drop as a key feature

- Biggest Hindrance: difficulty in saving the user's state after a drag was completed without refreshing the page
- Ajax have become widely known and a full complement of browsers support these techniques
- Drag and Drop has become a more familiar idiom on the Web.

Interesting Moments

- There are a number of individual states at which interaction is possible
- These microstates are called as *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?

- During drag, how will 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?

- There are a lot of events during drag and drop that can be used as opportunities for feedback to the user.
- There are a number of elements on the page that can participate as actors in this feedback loop

The Events (15 events available for cueing the user during a drag and drop interaction)

- Page Load → Before any interaction occurs, pre-signify the availability of drag and drop.
- 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.
- 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.

- Drag Hovers Over Invalid Target →User pauses over an invalid target without dropping the object.
- 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.
- 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

Actors

- During each event, it is possible to 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

- 15 events times 6 actors. That means there are 90 possible interesting moments

A simplified interesting moments grid for the original M y Yahoo! drag and drop design; it provided a way to capture the complexities of drag and drop into a single page

	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

Purpose of Drag and Drop

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

- To allow the user to directly place objects where he/she wants them on the page

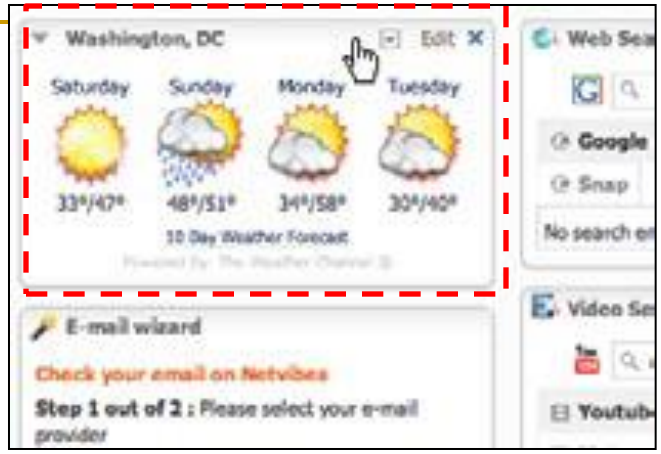
1. **Normal display style**

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

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

2. Invitation to drag

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



3. Dragging

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

4. 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

There are two common approaches to targeting a drop of dragged object

- **Placeholder target**

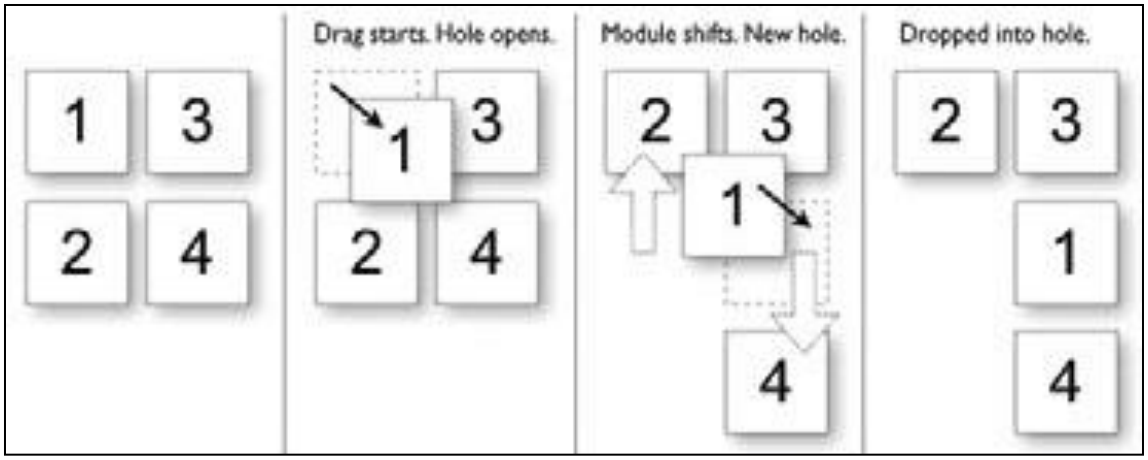
  Hole with dashed outline

- **Insertion target**

  Keep the page as stable as possible and only move around an insertion target (usually an insertion bar)

Approach 1: Placeholder target

- The idea 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



A placeholder target always shows where the dragged module will end after the drop;
Module 1 is being dragged from the upper right to the position between modules 3 and 4

Boundary-based placement

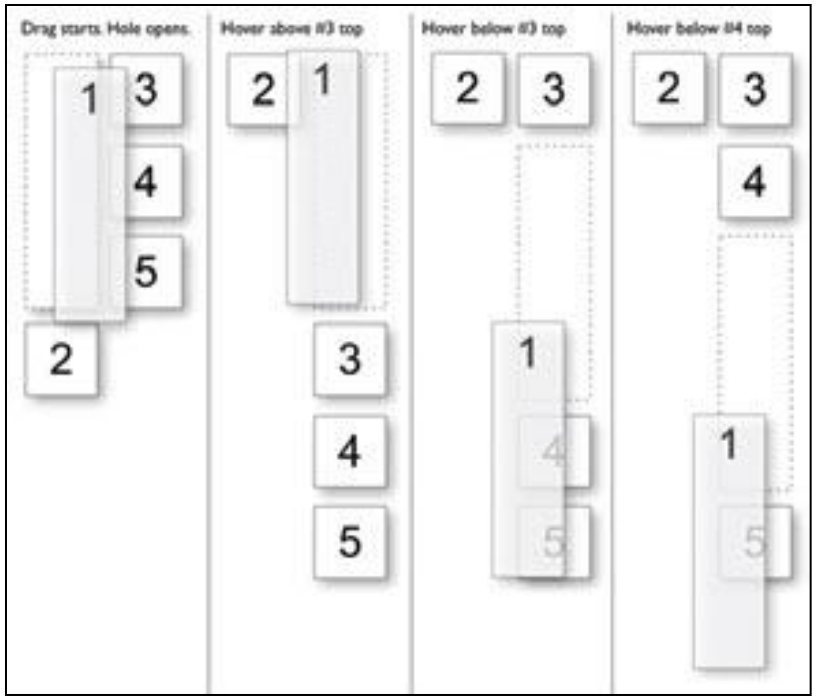
- targeting is determined by the boundaries of the dragged object and the boundaries of the dragged-over object.
- The mouse position is usually ignored because modules are only draggable in the title
- In Netvibes, the placeholder changes position only after the dragged module's title bar has moved beyond the dragged-over module's title bar.

In Netvibes, dragging a small module to be placed above a large module requires dragging a large distance; the “To Do List” has to be dragged to the top of the “Blog Directory” module

In contrast, moving the small module below the large module actually requires less drag distance since you only have to get the title bar of the small module below the title bar of the large module

Dragging a small module below a large module requires a smaller drag distance;
since the targeting is based on the header of the dragged-over module, the drag distance in this scenario is less than in the previous figure

Boundary-based drop targeting is non-symmetrical in the drag distance when dragging modules up versus dragging modules down



The Netvibes approach requires the dragged object's title to be placed above or below a module before the placement position changes; this results in inconsistent drag distances

Approach taken by iGoogle

- Instead of basing the drag on the *title bar, iGoogle calculates the placeholder targeting on the dragged-over object's midpoint.*

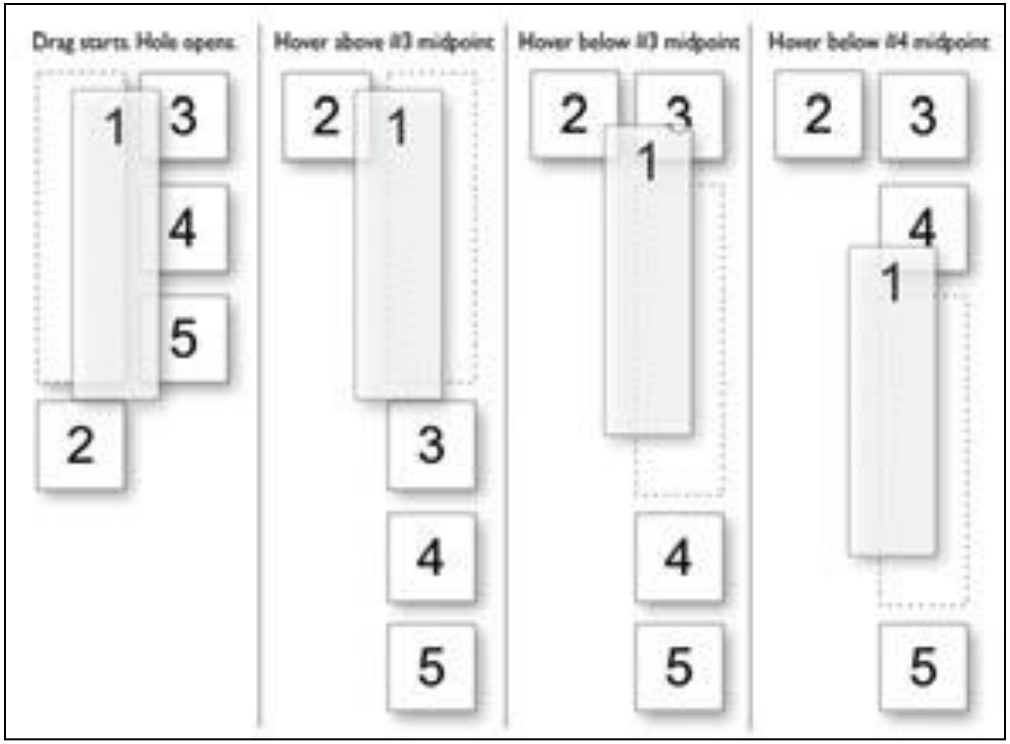
When dragging a module downward, iGoogle moves the placeholder when the bottom of the dragged module crosses the midpoint of the object being dragged over; the distance to accomplish a move is less than in the Netvibes approach

- When we drag the stock module up to place it above the moon phase module, iGoogle moves the placeholder when the top of the stock module crosses the midpoint of the moon phase module.

When dragging a module upward, iGoogle moves the placeholder when the top of the dragged module crosses the midpoint of the object being dragged over; dragging modules up or down requires the same effort, unlike in the Netvibes example

- Module 1 is dragged from the first column to the second column, the placeholder moves above module 3. As module 1 is dragged downward, the placeholder moves below 3 and 4 as the bottom of module 1 crosses their midpoints.

To create the best drag experience, use the original midpoint location of the module being dragged over to determine where to drop the dragged module: module 1 is being dragged into the position just below module 4



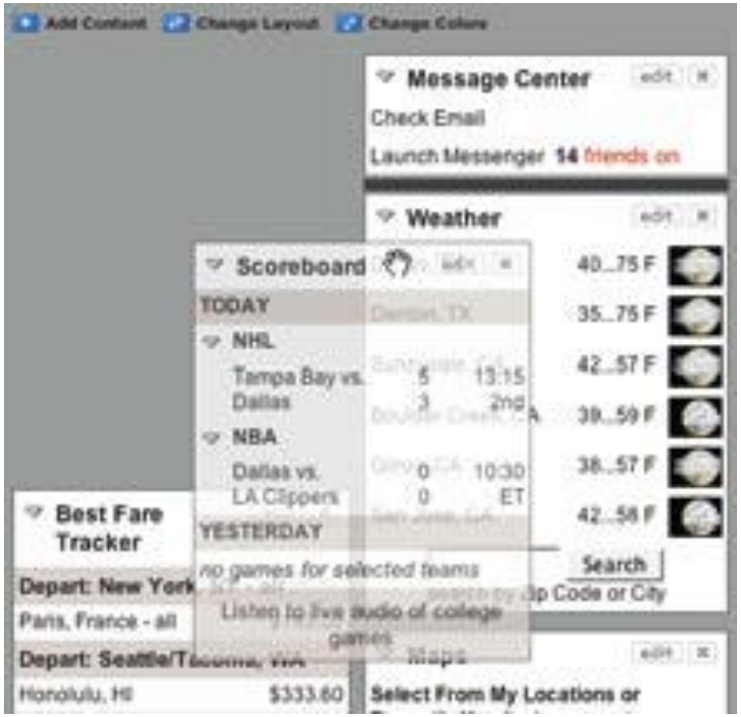
Interesting moments grid for the iGoogle drag and drop interaction.

	Mouse Hover	Mouse Down	Drag Initiated	Drag Hovers over Valid Target*	Drop Accepted
Cursor	Change to a hand pointer.	Change to normal style.*			
Dragged Module			Slightly transparent.		Dragged module removed.
Dragged Modules Original Location			Hole is shown as a gray, thick, dashed outline.		Hole is removed.
Drop Target				Hole (gray, thick, 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 is to switch to a hand that looks like it grabbed the module.*	** Drag initiaties instantly on mouse down.*	** Triggers when the mid-point of the dragged object enters a valid drop target.*	

Approach 2: Insertion target

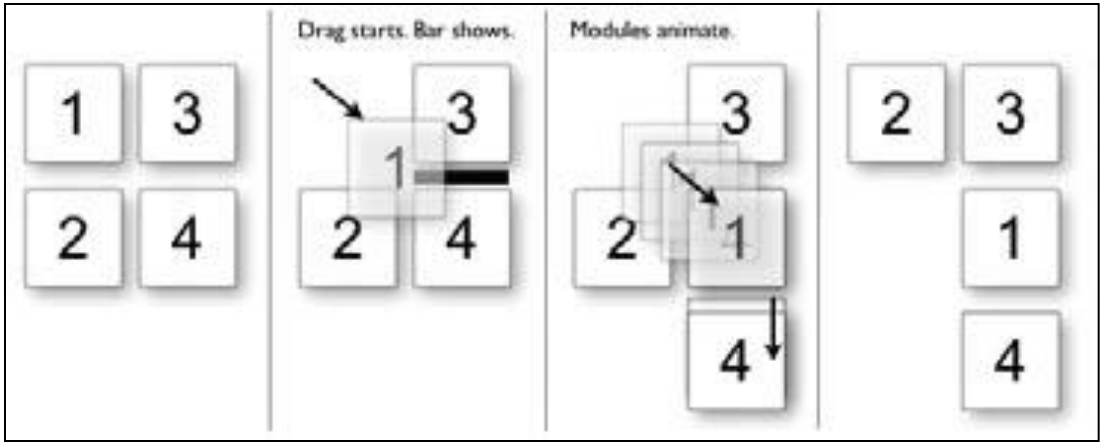
- Keep the page as stable as possible and only move around an insertion target (usually an insertion bar)

My Yahoo! uses the insertion bar approach



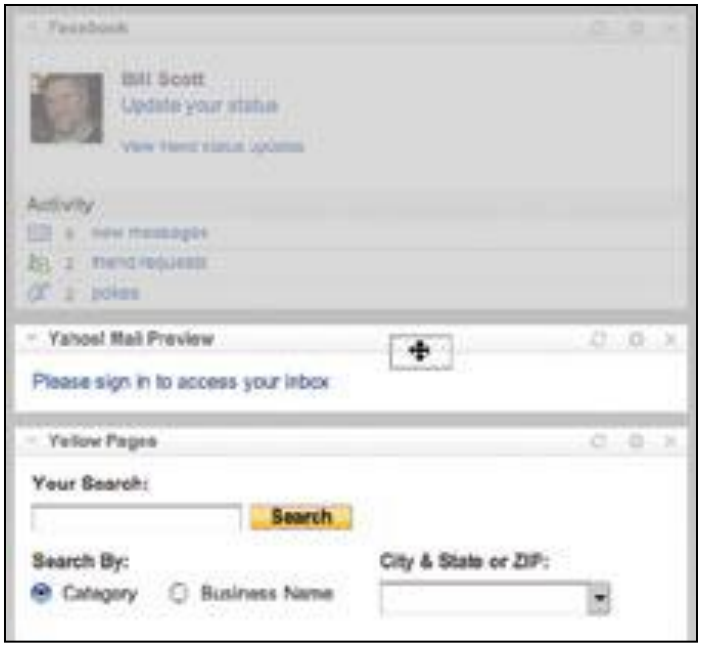
- While the module is dragged, the page remains stable. No modules move around.
- Instead an insertion bar marks where the module will be placed when dropped

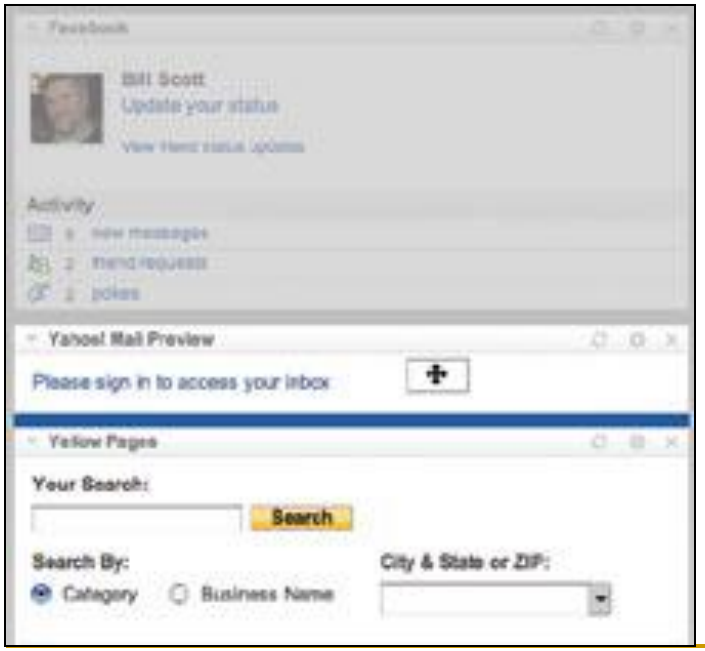
When module 1 is dragged to the position between 3 and 4, an insertion bar is placed there. This indicates that if 1 is dropped,
then 4 will slide down to open up the drop spot.



1. Dragging a module down

The gray outline being dragged. The small gray rectangle represents the dragged module. Since the drag representation is above the midpoint of the Yahoo! Mail Preview module, no change is indicated for a drop





2. Insertion bar appears

The dragged module representation is now below the Yahoo! Mail Preview module's midpoint. The insertion bar is rendered to show that the Facebook module will be placed just below the 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
- Drag rendering
  - My Yahoo!→small gray rectangle; Netvibes → full size as opaque; iGoogle → partial transparency

Best Practices

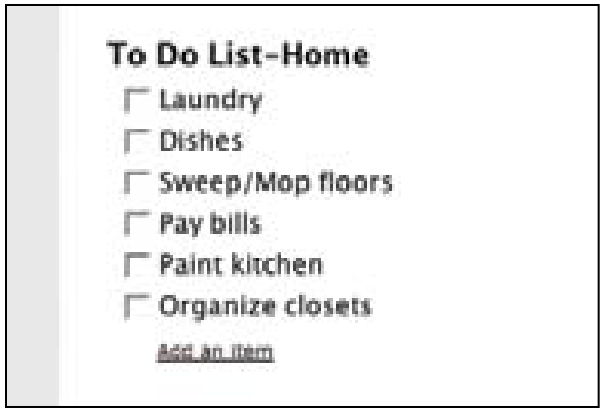
Best Practices for Drag and Drop Module

Here are some best practices to keep in mind:

- Use the placeholder approach when showing a clear preview during drag is important.
- Use the insertion bar approach when you want to avoid page jitter.
- Use the midpoint of the dragged object to determine drag position.
- Use a slightly transparent version of the object being dragged (ghost) instead of an opaque version.
- If you drag thumbnail representations, use the insertion bar targeting approach.

Drag and Drop List

- The Drag and Drop List pattern defines interactions for rearranging items in a list.

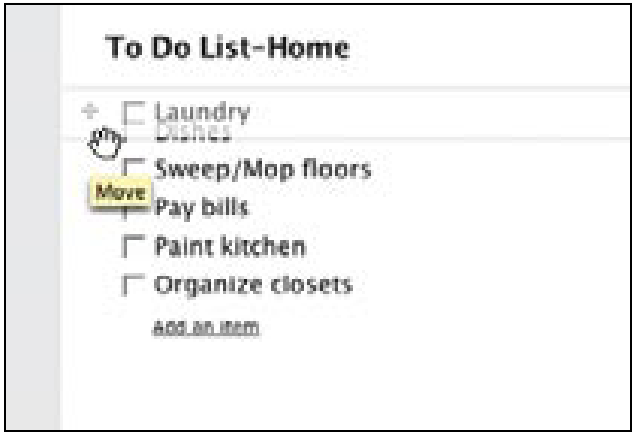


1. **Normal display state**

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

2. 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.



3. Dragging

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

4. Dropped

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

Considerations

- Approach 1: Placeholder target
  - When moving an item in a list, it is 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.

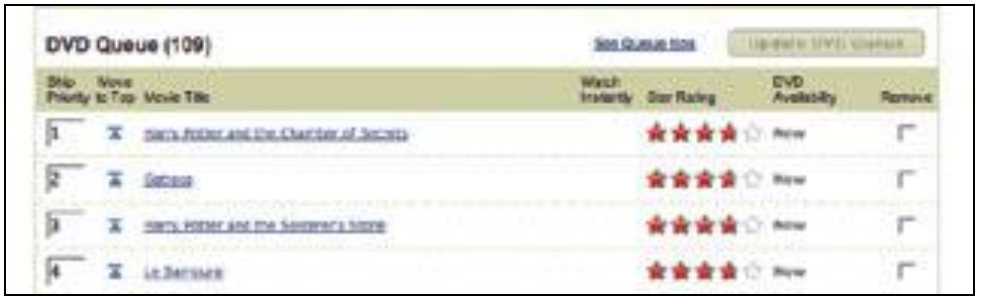
Example : Apple's iPhoto

iPhoto uses cursor position: when the cursor crosses a threshold (the edge of the next photo), a new position is opened up

Approach 2: Insertion target

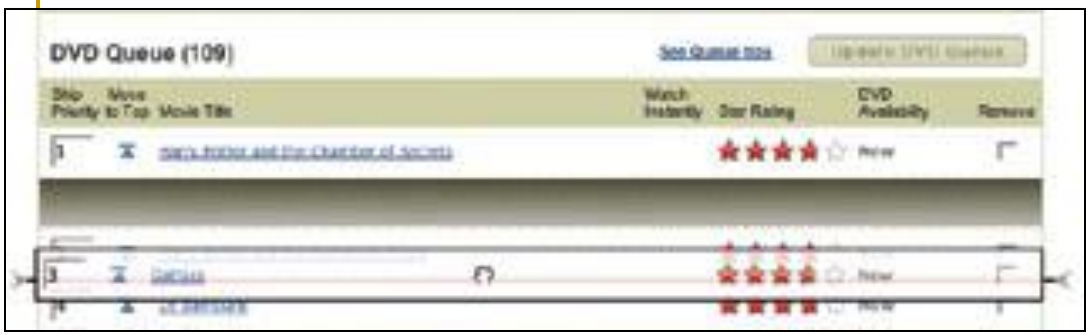
1. Normal display state

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



2. Invitation to drag

The cursor changes to indicate draggability.

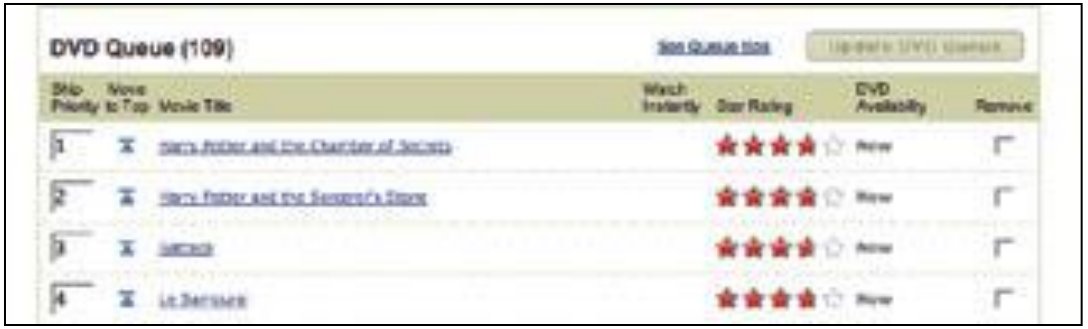


3. Dragging

A hole is marked where the item is pulled from. The dragged item's index number changes and an insertion bar indicates where it will be moved to.

4. Dropped

The item is moved immediately into the spot marked by the insertion bar.

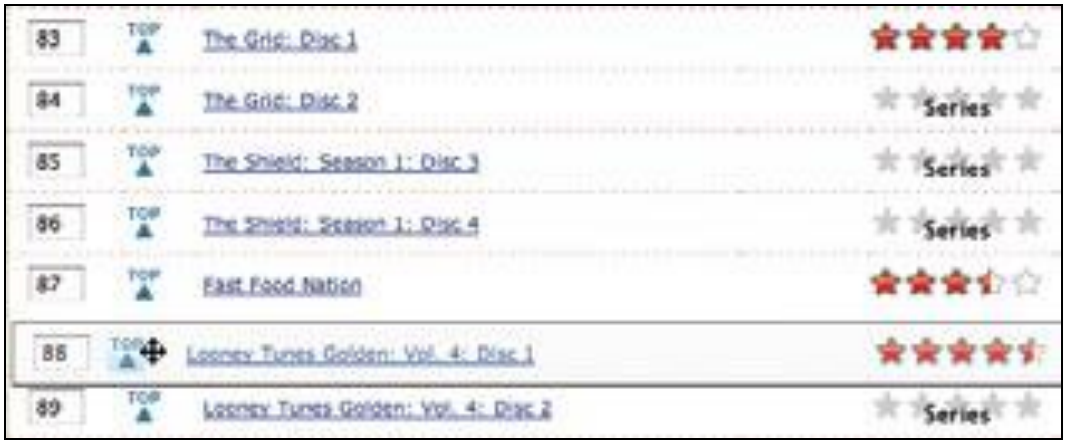


- Non–drag and drop alternative
  - Two other ways to move objects around
    - Edit the row number and then p • ress the “Update DVD Queue” button.
    - Click the “Move to Top” icon to pop a movie to the top
  - Modifying the row number is straightforward. It’s a way to rearrange items without drag and drop.
  - The “Move to Top” button is a little more direct and fairly straightforward

Hinting at drag and drop

1. Click “Move to Top”

Clicking the “Move to Top” button starts the movie moving to the top.



2. 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.

3. 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.

- Drag lens
  - 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

A good example of this is dragging the insertion bar while editing text on the iPhone

The iPhone provides a drag magnifier lens that makes It easier to position the cursor

Best Practices

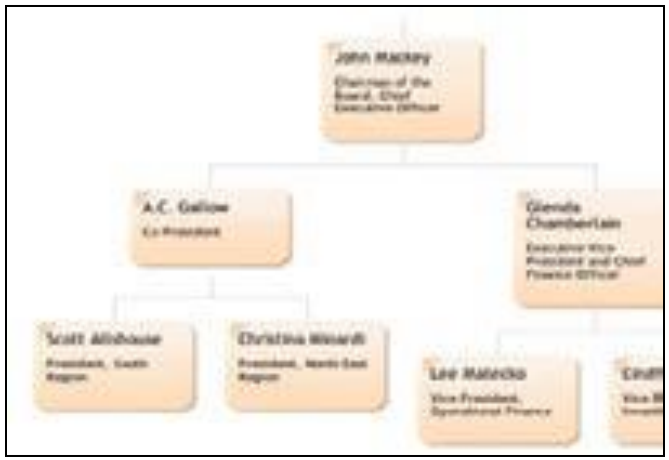
Best Practices for Drag and Drop List

Here are some best practices to keep in mind:

- If possible, drag the items in a list in real time using the placeholder target approach.
- Use the mouse position for drag target positioning.
- If the goal is speed of dragging or if dragged items are large, consider using the insertion target approach, as rendering an insertion bar is inexpensive compared to dynamically rearranging the list.
- Since drag and drop in lists is not easily discoverable, consider providing an alternate way to rearrange the list.
- When the user rearranges the list with an alternate method, use that moment for a one-time advertisement for drag and drop.

Drag and Drop Object

- Another common use for drag and drop is to change relationships between objects, when represented visually
- Drag and Drop Object is a powerful tool for visually manipulating relationships

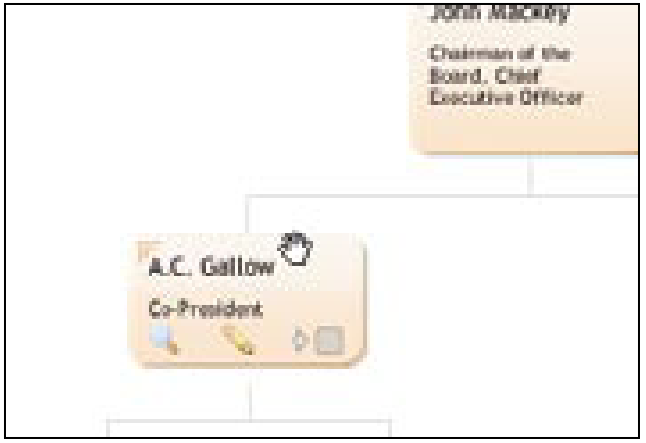


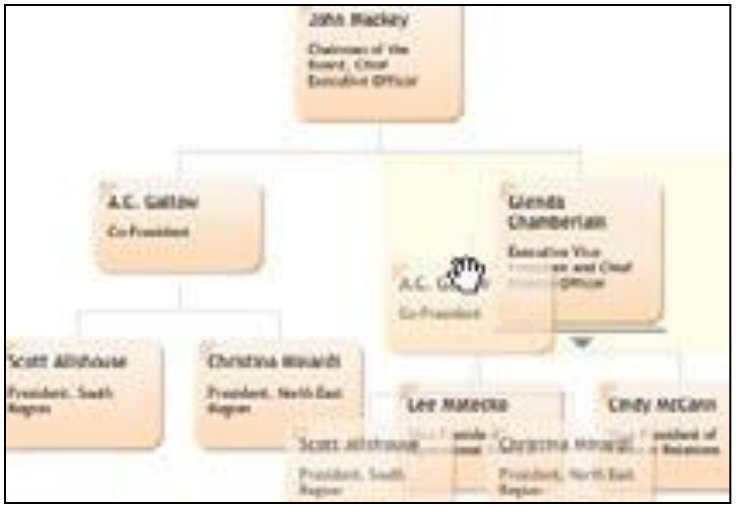
Normal display state

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.



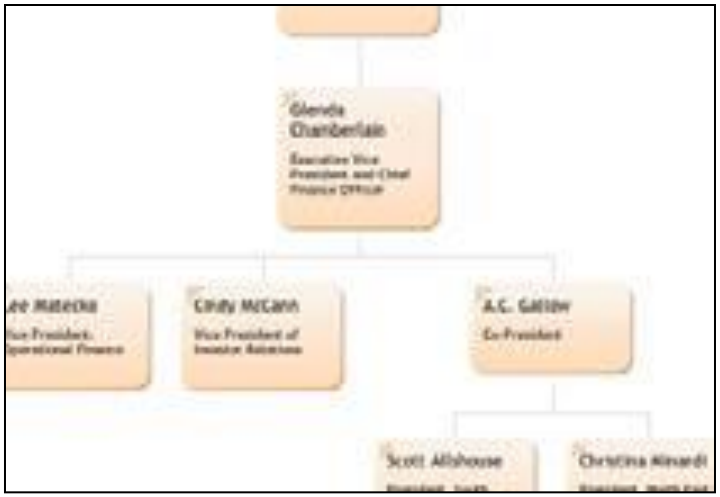


Dragging

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



Drag and Drop Action

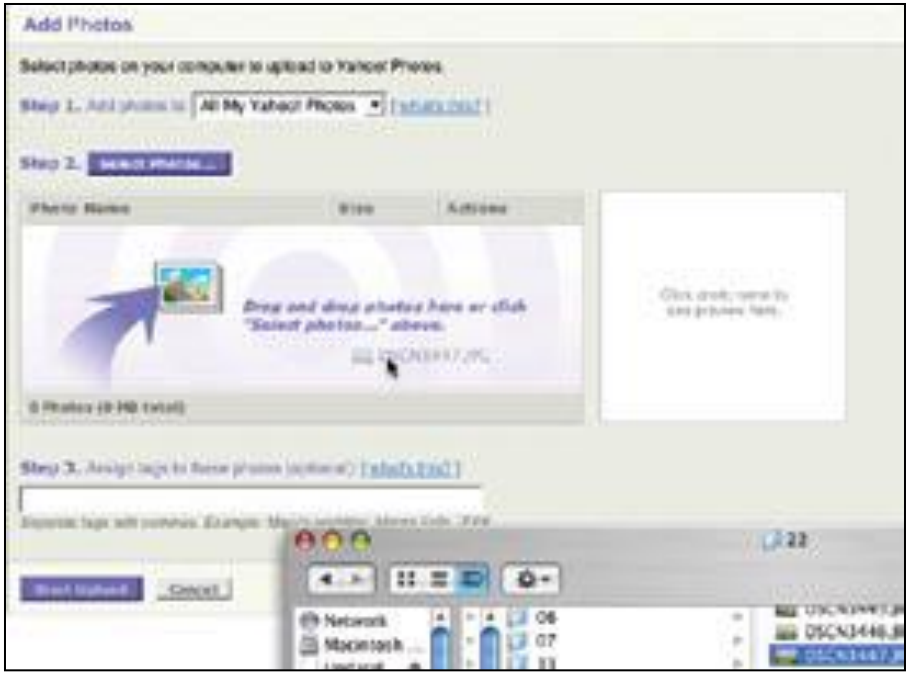
- Drag and drop is also useful for invoking an action or actions on a dropped object
- The Drag and Drop Action is a common pattern, *example: dropping an item in the trash to perform the delete action.*

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 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.

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

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.

Best Practices for Drag and Drop

General Best Practices for Drag and Drop

- Keep page jitter to a minimum while dragging objects.
- Initiate dragging if the user presses the mouse down and moves the mouse three pixels, or if she holds the mouse down for at least half a second.
- Use drag and drop for performing direct actions as an alternate method to more direct mechanisms in the interface.
- Hint at the availability of drag and drop when using alternatives to drag and drop.
- Pay attention to all of the interesting moments during drag and drop. Remember, you must keep the user informed throughout the process.
- Use **Invitations** (discussed more in Chapters 9 and 10) to cue the user that drag and drop is available.