

Drag and Drop Design Pattern

Problem Summary

The user needs a way to visually change the order or manipulate the position of elements on the screen.

Solution



Provide drag and drop capability on the page, so that the user can use a mouse gesture or keyboard key-combination to re-order items directly on the screen.

Use When

- You are displaying a list of items on a single page, and you want to give the user the ability to change the order or position of the items.
- You want to allow the user to change the order or position of items in context, without having to go to another page.

Why

- When a user is changing the order of a list, the visual context of the item's new position is often important (e.g. to put an item in alphabetical order, you need to see the items before and after it in the list while re-ordering). You can allow the user to 'visualize' the new order and position as they are performing the action by letting them perform drag action on the item directly in the list.
- Avoiding sending the user to a new page for re-ordering creates a more seamless user experience.

How

Give users visual indications about what is happening and/or possible at all the 'interesting moments.' The term 'Interesting Moments' comes from the [Yahoo Design Pattern Library](#) - these are key moments in the Drag and Drop interaction when a user needs some extra visual indication that they are able to do something, about to do something, or have just done something. Users are not used to this type of rich interaction on the web, so you need to pay particular attention to helping them to discover this functionality.

The particular visual cues you use will change slightly depending on the context of your Drag and Drop interaction. We have identified two sub-patterns that provide details about different visual cues based on the context of your application:

- **List Ordering** - A pattern for using Drag and Drop to re-order lists of items. The items are generally of a similar size and shape and organized in a standard layout (grid, horizontal list, etc).
- **Layout Preview** - A pattern for using Drag and Drop to edit more complex page layouts.

Accessibility

- **Provide keyboard-only equivalents of mouse-based actions.** Keyboard accessibility can be facilitated by providing keyboard-only equivalents of "drag and drop" functionality. Suggested keystrokes for applications are described in the sub-patterns.

- **Describe how the feature works using the keyboard.** A description of how the drag and drop operation works with the keyboard should be provided, i.e., "After selecting the object using the tab key, move the object by pressing 'Control' in combination with the relevant arrow key." Ideally, this information will be visible to all users, so that persons who are sighted but don't use a mouse will also benefit.
- **Provide additional detail.** Additional detail of how the feature works can be provided by users clicking on a nearby "i" or "?" button, i.e., "Objects replace the location of the existing item, causing it to move one cell over."
- **Make Assistive Technology (AT) users aware of object positions.** AT users must be made aware of the position of objects relative to one another. This can be done by placing objects within a borderless table, transparent to sighted users, but with column and row headings for persons using AT. They should also be told if the operation has been completed successfully. This message can be hidden from sighted users.
- **Include correct ARIA role markup.** Including correct [ARIA](#) role markup within your HTML is a way to ensure that Accessibility Technologies are able to interpret the layout and context of your DnD interface. Below is an example of ARIA markup for a draggable grid-cell. The cell is represented as an HTML DIV element, and the `aaa:grab` attribute specifies to ATs that the item is draggable.

```

<div rsf:id="scr=ignore" class="orderable" id="cell::13:"
xhtml10:role="wairole:gridcell"
aaa:selected="true"
aaa:readonly="false"
aaa:disabled="false"
aaa:grab="supported"
aaa:dropeffect="move">

<!-- Content of Grid Cell -->

</div>

```

- For more information on ARIA see the [W3C specification](#).
- Provide an alternate mechanism if necessary. Another option is to provide an alternate means of re-ordering items if a keyboard-based approach is not feasible-- this might be another page or section where a more standard 'form' interface is used.

Examples

iGoogle portal



Related Fluid Components

Reorderer & Lightbox

The Fluid [Reorderer](#) is a set of JavaScript objects that can be used by developers to create rich, accessible user interfaces that allow users to directly move around and re-arrange content on the page.

The Fluid [Lightbox](#) component is an implementation of the Fluid Reorderer which allows users to re-organize images within a collection by dragging-and-dropping thumbnails.

Together these components provide a framework for Drag and Drop that supports all the visual cues specified in this pattern, keyboard access and automatic handling of some ARIA state changes. The first release of has implementations of both sub-patterns. The Layout Preview pattern is implemented for keyboard, and the List Ordering pattern is implemented for mouse. Developers may wish to change this behaviour to provide visual cues for the sub-pattern that best matches your application context.

You can find more information about these and other Fluid components on the [Components](#) page of the [Fluid wiki](#).

Related Patterns

- [Drag and Drop - List Ordering](#)- sub-pattern
- [Drag and Drop - Layout Preview](#)- sub-pattern

References

[Yahoo Design Patterns Library: Drag and Drop Modules](#)