

# Interaction Styles and Mobile Interactions

We will consider human-computer interactions from several perspectives:

1. interaction style
2. user concept of the interaction
3. interactions for mobile devices

## Human-computer Interfaces Styles

Although WIMP interfaces, which are a combination of Menus, dialog boxes, and 'point and click,' currently dominates interface designs there are other **forms of interfaces**:

- Command line interface
- Menus
- Question-answer/Dialog boxes
- Forms
- Spreadsheets
- Point and click
- Natural language
- Command Gesturing
- General Gesturing
- Direct manipulation
- Tangible interaction

We will go through the list and identifying the technique in terms of how fast it is for the user, how flexible it is for users or expressive it is for the user, how long it takes for the user to learn, and how hard it is for the programmer to implement.

**Command line interfaces** are the original interfaces for the computer and are still used. Examples of command line interfaces are UNIX operating system commands and the VI editor. They are loved by system administrators for their accuracy, speed and flexibility. If the user knows the commands then typing is faster than searching for it in menus. Consequently, some applications try to offer both; for example auto cad or Windows Excel. Users can directly create script files and verbally specify the command sequence. Some commands can be hard to visualize and searching for commands or files can be frustrating and slow. The interface is easy for programmers to implement.

**Menus** are a basis of WIMP interfaces and a favorite with inexperienced users. Novice users can easily find commands. Searching the menus, the user builds up a metaphor for the application. Using menus is slow but fun. Using toolkits, menus are not too bad to program.

**Dialogs** or “question and answer” boxes are another old interface style. They are windows that pop up asking for information, fields to be filled or buttons to be pressed. Dialogue windows have been around before WIMP interfaces, they first appeared in database entry application programs. Other examples are wizards and some help agents. They ask a sequence of questions using dialog boxes and in order to determine what the user needs. They are rather inflexible form of interface. Dialog boxes are easy to understand but not very flexible or fast. They are easy to program.

**Forms** are much like dialog boxes. They can be more sophisticated, and like dialog boxes not very flexible for the user. **Spreadsheets** are a flexible and powerful form of interface, especially if the user can specify cell types. Data entry in spreadsheets is typically slow. They are more difficult to program.

**Point and click** interfaces were made popular by the web. They represent I and P in WIMP. They were very suitable for the initial web browsers (gopher) when web pages were all text. Users knew to interpret the underscore as a link to another web page. Now, links are **hidden**, for example in images. Icons on the desktop is another example of point and click style interface. The notion of point and click is a short interaction that results in a very specific result. Because the user must move the mouse, this interface style tends to be slow for the user. It is flexible because many different kinds of UI objects can be pointed at. Short key interaction is a “point and click” interaction style without the point. Both are generally easy to implement.

The most common example of natural **language interface** are the interface of search engines. Users type words in to the search box that the software interprets and return the result for the user. They are hard to implement, but can be very flexible for the user.

**Command gesturing** interface style selects an object and uses a gesture to issue commands. In essence, it is a generalization of “point and click” interfaces. Examples are in Opera, a web browser. Swiping on smart phones is another example of command gesturing. Other examples of command gesturing are the hand cursor in Adobe. Some games, Brother in Arms, use command gesturing. Keith Ruthowski has demonstrated that a pie menu can become a form of command gesturing, and once the gestures are learned nearly as accurate and fast as text entry. Because algorithms for interpreting gestures are in its infancy, the flexibility of command gesturing is not known. They are difficult to implement. Learning a large set of gestures can take a long time.

**General gesturing** is a more general interface style than command gesturing. There does not have to be an object and the gesture does not have to represent a command. Examples of general gesturing are drawing applications and text entry by writing like on notebooks. The Wii is advancing general gesturing in games. Because this is a very new interaction style, it is unknown how easy it is to learn, but it should be more flexible. It is more difficult to implement.

Direct manipulation is closely related to command gesturing. An example of **direct manipulation** is drag and drop files into folders or trash. Drawing applications use direct manipulations. They can be slow to use but are fast to learn. They can be difficult to implement.

**Tangible interactions** refer to manipulating physical objects other than the mouse and keyboard. There are few current popular examples, but RFID and NF technology does make some tangible interactions possible. Low tech examples of tangible interactions are real buttons, switches and sliders. They can be fast or slow to use, but should be easy to learn, and can be hard to implement.

## Conceptual Interaction Models

Preece, Rogers and Sharp in *Interaction Design* propose that understanding users' conceptual models for interaction can guide HCI designers to the proper interaction techniques for their system

*The most important thing to design is the user's conceptual model. Everything else should be subordinated to making that model clear, obvious, and substantial. That is almost exactly the opposite of how most software is designed. (David Liddle, 1996, *Design of the conceptual model*, In *Bringing Design to Software*, Addison-Wesely, 17-31)*

The HCI designers' goal is to understand the interaction in terms that the users understand of them. Preece, Rogers and Sharp propose four conceptual models for interaction concepts, based on the type of activities users perform during the interaction.

- **Instructing** – issuing commands to the system
- **Conversing** – user ask the system questions
- **Manipulating and Navigation** – users interact with virtual objects or environment
- **Exploring and Browser** – system provides structured information

I propose additional conceptual interactions that are more passive:

- **Passive Instrumental** – the system provides passive feedback to user actions
- **Passive informative** – the system provides passive information to the users.

User may interact with a system using more than one conceptual interaction model.

## Instructional Interactions

Issuing commands is an example of instructional interactions. Instructional interactions are probably the most common form of conceptual interactions. It allows the user the most control over the system. Specific examples vary from using a VCR to programming. In most cases, issuing commands to the operating system are instructional interactions. Icons, menus and

control keys are examples of improving the usability command line like instructional interaction. Instructional interactions tend to be quick and efficient.

## Conversational Interactions

The conversational Interaction are interactions like user-system dialog. Examples of systems that are primarily conversational are help-systems. Agents (such as the paper clip) use conversational interaction. Implementing conversational model may require voice recognition and text parsing or could use forms. The advantage of conversational model is that it can be more natural, but it can also be a slower interaction. For example, using automated phone based systems is a slow conversational interaction interface. Another disadvantage of conversational interaction is that the user may believe that the system is smarter then it really is, especially if the system uses an animated agent.

## Manipulating and Navigational Interactions

This model describes the interaction of manipulating virtual objects or navigating virtual worlds. Navigational interactions are popular in computer games. Manipulating interactions occur in drawing software. Navigational interactions occur even in word processors, for example zooming and using the scroll bar. **Direct manipulations** are manipulating interactions. Ben Shneiderman (1983) coined the phase and posed three properties:

- continuous representations of objects
- rapid reversible incremental actions with immediate feedback
- physical actions

Apple was the first computer company to design an operating system using direct manipulation in the desk top. Direct manipulation and navigational interactions have a lot of benefits. They are easy to learn, easy to recall, tend to have less error, give immediate feedback, and produce less user anxiety. But they have several disadvantages: the interactions are slower and the user may believe that the interaction is more than it really is. Poor metaphors such as moving the icon of a floppy to eject the floppy can confuse the user.

## Exploring and Browsing Interactions

Exploring and browsing interactions refer to searching structured information. Examples of systems using exploring interactions are Music CDs, Movie DVDs, Web, portals. Also searching for files using window explorer is an example of Exploring interactions. Not much progress has been made in this conceptual model for exploring and browsing interactions, probably because the structuring information is a non trivial task and is hard to model.

## Passive Instrumental Interactions

Passive instrumental interactions are similar to instruments, for example the speedometer in an auto dashboard. They can provide feedback to users' actions or movements, such as a GPS interface. They can also provide information to changes in the environment such as a light meter or an image in a viewfinder. Smart phones frequently used as instruments and make use of passive instrumental interactions.

## Passive Informative Interactions

Smart phones are frequently used to read books. The interaction is very passive and one way. The system is providing information to the users. The user primarily gestures to progress through the book. Viewing images is another passive informative interaction with only interactions for zooming and panning. Passive informative interaction may be a simplified Manipulating and Navigational Interactions.

We can make a table summarize interaction styles and conceptual interaction models. The related conceptual interaction model is the most common model that is supported by the interaction style. Implementation is how hard for the designer to implement. Because I made the table we should go through it and correct it.

Interface	User speed	Flexibility	Learning	Implementation	Conceptual Model
Command Line	fast	high	slow	easy	instructing
Menus	slow	medium	medium	medium	instructing
Dialog	slow	none	fast	easy	conversing
Forms	medium	not much	fast	easy	instructing, conversing, browsing
Spreadsheet	slow	high	slow	hard	instructing, conversing
Point & Click	slow – fast	none	fast	easy	instructing, manipulating, browsing
Natural Language	fast	very	slow-fast	hard	conversing
Command Gesturing	fast-medium	medium	hard	hard	instructing
General Gesturing	fast	high	slow-fast	hard	manipulating
Tangible interaction	slow-fast	low	fast	medium to hard	manipulating, browsing

# Mobile Interactions

We will explore the interactions possible in mobile web app via the technologies that enable the interactions.

- HTML
- CSS
- Twitter Bootstrap (as an example of CSS framework)
- HTML 5
- Various other JavaScript Libraries

## HTML

HTML is the lowest level technology that enables interactions on the web. Basic HTML is very to learn, and it has enhanced features. The client browser implements interpretation of the HTML so exact interaction varies with the browser.

The best resource for HTML is W3 School.

<http://www.w3schools.com/html/default.asp> Views and Layouts

## Views and Layout Tools

The goal of HTML is to provide a syntax that is independent of the browser and window dimension, but this can be a challenge. Consequently, HTML only offers the most primitive layout tools. The layout basically flow from the top of the page to the bottom of the page. Some of the layout tools are

- Paragraph tag – example of section of the page
- image tag – example of non-text layout
- Table tags – advance layout for displaying data
- Layout tags – such as header, nav, section, aside, article and footer. An advance layout for conveying semantic information
- IFrame tag – layout that enable displaying another a page within another page

The original intention of table tags was to display data as in a spreadsheet. It was not designed to be a layout tool, but in early HTML is the only tool to control layout. Using tables for layout is an awkward syntax, so older HTML editors and IDE focused on making the table tags usable for web designers to express layout. The main fault with tables for expressing layout is that the layouts are not responsive to different window dimensions.

## Interactions

The basic HTML interactions consist of links and forms:

- Links
- Button
- Text Fields
- Radio Buttons
- Check Boxes
- Browser Back
- Browser Closed
- Browser URL field

The original HTML interaction technique was traversing a link. The link concept of a was developed by Nelson (1965) as extension of Bush's (1945) Memex description. Original the link was a powerful tool to relate documents. Later the link was used to initiate an user generated event for the browser to detect.

When messaging boards were developed for HTML, forms tags were developed. The form input tags include text fields, radio buttons and check boxes. The typical interaction is for user to use input tags to define the input data and then click submit which generates a POST request to the server. The form data is mapped in the body of the request.

We should also include the interactions that the browser offers which include back button, the browser address field and the closing button.

## CSS

In original HTML, styling were expressed by attributes in the element. This made it hard to maintain large website styles. To change the style one had to search and edit all the element tags. Cascading Style Sheet syntax was develop so that style can be expressed not in the the tags, and so that the styling can all be in one place. Cascading comes from the priority that styles definition based on the location of the rules: user defined, inline, in the page, or in a separate file.

CSS can express more than just the style. It can express animation or changes in style for common events such as hovering, clicking etc. A good resource for CSS is

<http://www.w3schools.com/css/>

## Twitter Bootstrap

Twitter Bootstrap is an example of a css framework. In essence, Twitter Bootstrap is a style sheet with some JavaScript code. Twitter Bootstrap has a very complete style set and was one the first frameworks to be "mobile first." CSS frameworks provide consistent styling for all the HTML tags.

The best resources for Twitter Bootstrap is at the official website and W3 Schools.

<http://getbootstrap.com/>

<http://www.w3schools.com/bootstrap/default.asp>

## Views and Layouts

Besides styling for the HTML, Twitter Bootstrap offers advance views and layouts.

- grids
- Jumbotron

The grid is used to define the responsiveness of the layout, meaning how elements should be laid out for different window widths. A row is a horizontal layout and columns divide the row. The column class specification defines the size of the width and the window width break points. Twitter Bootstrap is “mobile first” design, meaning that generally each column is a single row in a small device and the break points define the number of column size for larger window width.

Jumbotrom is a large display. The name references the Sony's original 1985 giant display. They are popular for home page titles and images.

## Interactions

Besides interactions provided by HTML, Twitter offers some advance interaction widgets.

- Navigation Bar (Navbar) – menu bar can be located at the top (static or fix) or on the side
- drop-down menu – set of links dropping down from menu items or buttons
- Notices – panels that conditional display
- Modal – a window that overlays and holds focus until the user response
- Accordions – collapsing panels
- progress bar

Navbar enable a menu similar to what user are familiar with in desktop applications. In essence, the navbar help to enable the idea of web apps. The combination of navbar and drop-downs can make the web app functionality as vast as a desktop application.

Modal windows grab focus and force the user to respond. They are good for alerting the user before they might delete a database entry. But modal's should be used with caution. If you find yourself writing modal with just one button, I suggest reconsidering your design. A notice might be better or at least have a check box with “do not show again.” Modal's or overlays are also used to show expanded views. Accordions are good for concealing and revealing detail information in a list.

## HTML 5

HTML 5 our proposed standards to extend HTML functionality. Many of the proposed standards have already been implemented by major browsers. Some of these standards add additional interaction technique via JavaScript.

## Interactions

The API listed below add interactions techniques to most modern browsers. Google Chrome browser supports all of the API:

- Geolocation – <http://dev.w3.org/geo/api/spec-source.html>
- Multimedia – <http://www.w3.org/html/wg/drafts/html/master/embedded-content.html#media-elements>
- Canvas – <http://www.w3.org/html/wg/drafts/html/master/scripting-1.html#the-canvas-element>
- SVG – <http://www.w3.org/Graphics/SVG/>
- Motion Sensors – <http://w3c.github.io/deviceorientation/spec-source-orientation.html>
- Form Virtual Keyboards
  - <http://www.w3.org/html/wg/drafts/html/master/Overview.html>
- Touch Events – <http://www.w3.org/TR/touch-events/>
- CSS 3 Transitions – <http://www.w3.org/TR/css3-transitions/>
- CSS 3 Animations – <http://dev.w3.org/csswg/css-animations/>
- WebGL – <https://www.khronos.org/webgl/>
- HTML Media Capture – <http://www.w3.org/TR/html-media-capture/>
- Web Speech API – <https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html>
- Vibration API – <http://www.w3.org/TR/vibration/>

The Geolocation API gives access to the GPS even when the browser is offline. Multimedia provides audio and video tags. Motion Sensor gives access to the device's accelerometers and consequently can be used to implement the compass and level. Form Virtual Keyboards give mobile devices different keyboards depending on the text field attributes. Touch Events give access to continuous X-Y page and screen coordinates, so can be used to implement drawing and gesturing. HTML Media Capture gives access to the camera in the device so that photos videos can be captures. Web Speech API provides speech recognition and Vibration API vibrates the device, but these are implemented only in Chrome.

Most but not all these APIs are implemented by the major modern browsers, you should check the implementation status at

<http://mobilehtml5.org/>

The above APIs give web apps nearly the same functionality of native apps. Some of the interactions are:

- GPS Location
- Touch

- Gesturing
- Orientation
- Photo
- Vibration
- 2D drawing
- 3D rendering

## Other JavaScript Libraries

There are many JavaScript libraries that build on the base features of HTML and HTML5 to provide advance interaction techniques. Below is only a short list of libraries that you may find useful.

### Google Maps API

Google Maps API is a convenient JavaScript library for displaying maps and map icons. The displays offer user control including locating icons.

The best resources are Google Developers W3 Schools websites:

- <https://developers.google.com/maps/>
- <http://www.w3schools.com/googleAPI/default.asp>

### JQuery and JQueryUI

JQueryUI is an advance library built on top of the very popular JQuery library. More than 500 plugins widgets (for example autocomplete) offer advance user interface interactions. Below is short list of useful resources:

- <http://www.w3schools.com/jquery/default.asp>
- <http://jquery.com/>
- <http://plugins.jquery.com/>
- <http://jqueryui.com/>
- <http://learn.jquery.com/>

## Interactions

We should list interaction techniques available on the smart phones and associate their constraints and opportunities.

Interactions	Opportunities	Constraints/Difficulties
Viewing	Any where and time	Small screen Low resolution

Touch	Basic input	Space for only a few buttons, Small buttons
Long Touch	Context menu	User are unaware, requires time
Gesturing	More expression to touch	Small spaceLimited gestures
Keying	Text input	Small keyboard, Error prone, slow
Spinner	Alternative to text input	Only a few selections
Auto Completion	Assist text input	Error prone, Complex use
GPS location	location,documentation	Low resolution, 30 meters, Slow
Orientation	Alternative inputprovide direction	Noisy, User imprecision
Microphone	Alternative text inputOther inputs	Poor quality, Transcription hard and error prone
Speaker	Alternative outputfeedback	Poor quality, Inappropriate use in public
Time	Documentation	Little use
GPS Motion	Alternative input, Direction, Area	Imprecise, Slow
Accelerometer Activities	Alternative inputMeasure activity	Small vocabulary, Imprecise
Photo	Documentation,Vast informationAlternative inputAlternative text input	Hard to interpret, Large storage space, Slow
Vibration	Low noise outputDoes not require view	Small vocabularyImpreciseUnnoticed
WiFi	Vast information	Slow, Not always available, Small screenLinks hard to touch
Bluetooth	Locale communicationTransfer information	Public, Complicate connect protocol, Insecure
Bluetooth Devices	Many opportunities	More than one device, Complicated communications
NF	Tangible interfaces	More secure, range 1 meter

Others		
--------	--	--

In general mobile apps frequently use:

- Viewing
- Touch
- Spinners

They avoid the using:

- Keying
- Audio

When the opportunity arises, they should make smart use of:

- Gestures
- GPS location
- Orientation
- Time
- Photos
- Vibrations
- WiFi
- Bluetooth

New opportunities for interactions techniques are provided by:

- GPS
- Accelerometers
- Photos
- Vibrations
- Bluetooth
- Bluetooth devices
- Near Field