Interaction and Navigation Pattern Checklist

Modern user interface stacks provide a platform for a wide variety of patterns for user interaction. Even older platforms have more options than most development teams have traditionally used.

This list provides a starting point to find an appropriate interaction/navigation pattern during a design process. Typically, the list should be scanned for candidates that might work well for a particular application or function within an application. For some domains, there is only one obvious candidate. For others, there might be several, and part of the design/prototyping process should be used to try out the possibilities and choose the right one.

Many business applications will need several of these patterns, with different functions needing different patterns, and with one pattern chosen for the main "shell" navigation pattern.

Option list or menu

Option lists have been used since the beginning of software on video terminals. A menu is an example of an option list. Here are some of the typical forms of option list:

Standard menu

A standard "top of the screen" menu, such as the File | Edit | Windows | Help menu is probably the most common example of this pattern. It evolved from older examples of the pattern, such as cascading lists, in which a master list of options was displayed, and choosing an option brought up another list which was a sub-menu.

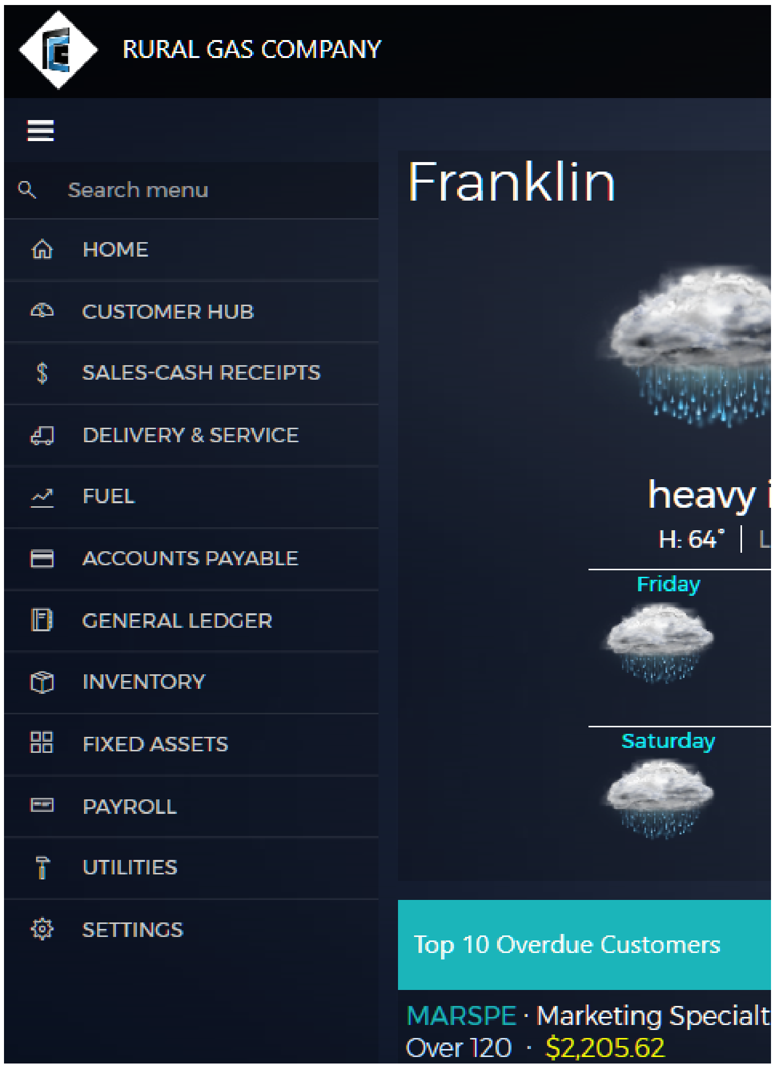
Hierarchical tree

A File | Edit | Windows | Help menu is actually hierarchical in structure, but it is sometimes supplemented or replaced with a tree-based set of options, often on the left side of the screen.

"Hamburger" menu →

The Windows 10 equivalent of the File | Edit | Windows | Help pattern is the "hamburger" menu. It is so named because of the stack of three bars in the upper left corner. By convention, each menu item has an icon and a text description.

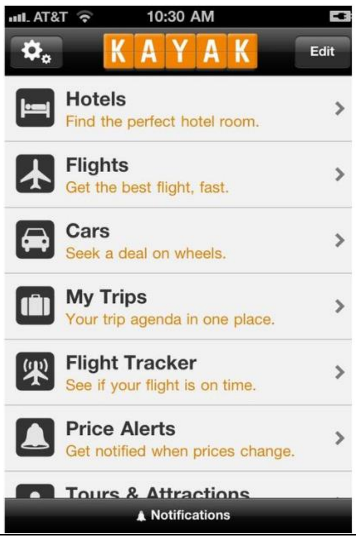
Simple hamburger menus are also used on many mobile platforms, such as Android. They work well with touch screens, and with screens of limited size. A standard capability to work with small screens is a collapse function. Touching the hamburger stack causes the menu to collapse to just icons, or if collapsed, to expand and show the full labels.



Hamburger menus on desktop applications often show all the time. By contrast, hamburger menus on mobile devices often only show when you press the hamburger icon.

Simple List

For simple applications, often on mobile platforms, no hierarchy of options is necessary. Only a simple list is needed. Here is an example →


AT&T
10:30 AM
KAYAK
Edit
Hotels
Find the perfect hotel room.
Flights
Get the best flight, fast.
Cars
Seek a deal on wheels.
My Trips
Your trip agenda in one place.
Flight Tracker
See if your flight is on time.
Price Alerts
Get notified when prices change.
Tours & Attractions
Notifications

Dynamically filtered lists

Another option list variant is the list which can be dynamically filtered by the user. This pattern is useful in cases where there are too many options for a manageable single list, but it is undesirable or technically difficult to use a hierarchical tree, as in a mobile application.

Several of the options above can have filters. It is common in desktop applications, for example, for a hamburger menu to have a search/filter capability.

Indications	***Cautions***
• App is simple • App needs to comform to expected conventions • Options loaded are highly dynamic and unpredictable • You're too lazy to go through the rest of the possibilities and decide which one really fits the users' task flow	• Respect Hick's Law: Long lists of options slow down the user • Mitigation options • Dynamic filtering • "Find" searching • Favorites • Most recently used options • Role-based subsets

Launchpad

The next most commonly seen pattern for options is the Launchpad. The desktops for Windows, Macintosh, iPad, and many other platforms are obvious examples. Other descriptions for the Launchpad pattern that you may have seen include "springboard" or "hub and spoke".

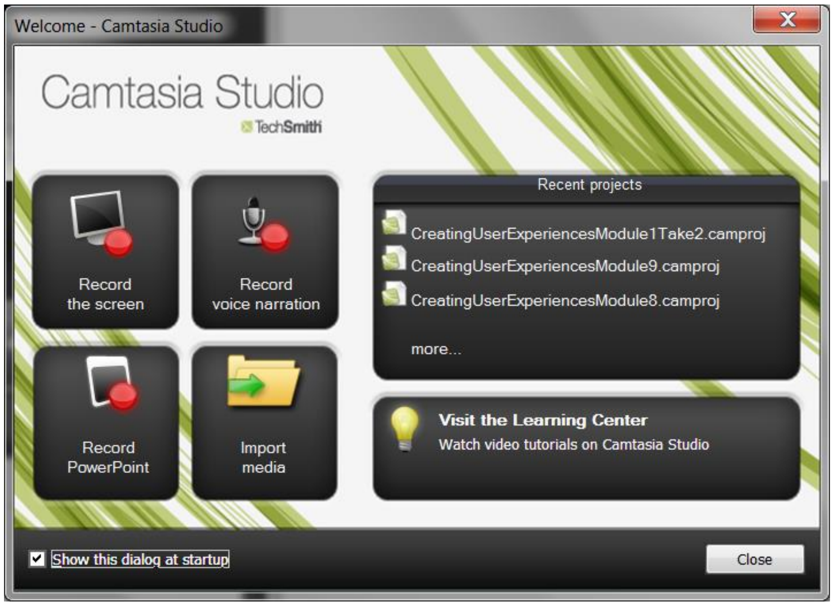
There are several variations on the Launchpad, based on the type of item being "launched".

App centric Launchpad

Some desktops, such as the Windows 8 desktop and the iPad desktop, are *app centric*, which means the only or predominant type of item is a means to launch an application. This pattern can also be seen in some business systems, where a single shell is used for multiple applications. In that case, the business system may have an initial launch pad to allow the user to choose the application they wish to use.

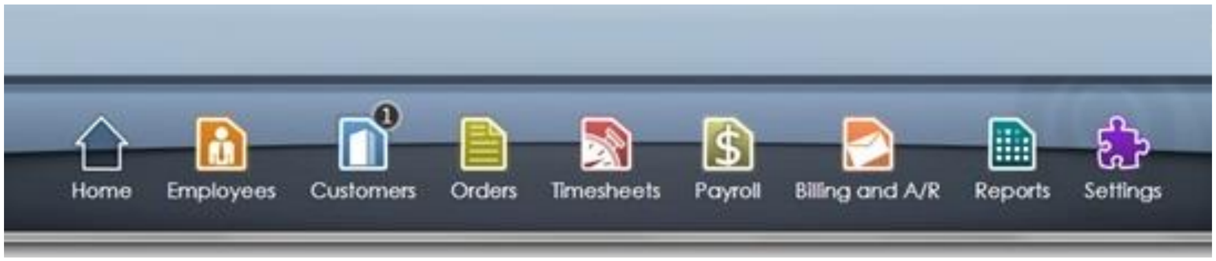
Action centric LaunchPad

Some applications use an *action centric* launch pad, in which the items launched are actions within a single application. A good example is from the screen video capture program, Camtasia:


Welcome - Camtasia Studio
Camtasia Studio
TechSmith
Record the screen
Record voice narration
Record PowerPoint
Import media
Recent projects
CreatingUserExperiencesModule1Take2.camproj
CreatingUserExperiencesModule9.camproj
CreatingUserExperiencesModule8.camproj
more...
Visit the Learning Center
Watch video tutorials on Camtasia Studio
Show this dialog at startup
Close

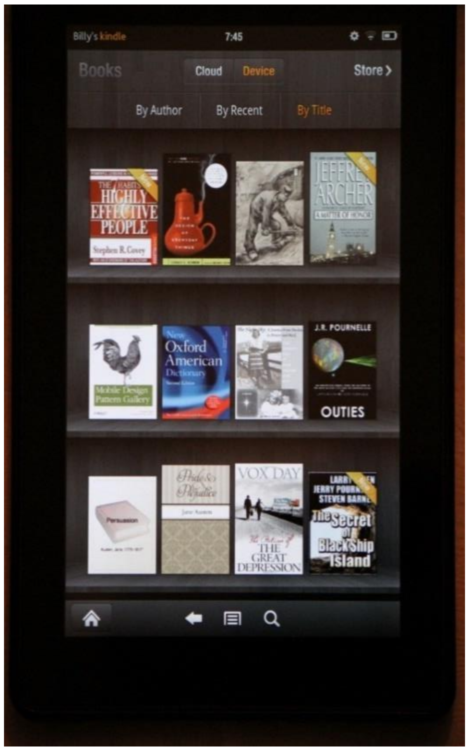
Entity centric LaunchPad

Within an application, navigation can be entity centric if the work process depends heavily upon entities. The StaffLynx app, shown as an example in the course, uses an *entity-centric* Launchpad at the bottom of the screen.



Gallery (Document/media centric Launchpad)

A cousin to the entity-centric Launchpad is a gallery, in which documents or other items are selected and "launched". A typical Windows Explorer window that displays icons for documents would be an example. Many image editing applications have a document centric Launchpad. Here is an example of a gallery to navigate books on the Kindle Fire, shown on the right: →



Indications	*Cautions*
• You want immediate familiarity • Dynamically loaded options with no categorization • Desire to conform to platform metaphor ◦ Making your app's control screen look like Windows 8 desktop	• Include enough difference that your app control screen is easy to distinguish from platform desktop • Respect Hick's Law ◦ On launchpads, that often means allowing configurability by users to include/exclude options ◦ Again, nothing wrong with a Find capability

Wizard

Most developers are quite familiar with wizards, so no additional description is warranted here.

The Wizard pattern is most often used either for relatively inexperienced users, for tasks that are done very infrequently, or for highly complex tasks that have a very linear workflow. Multiple steps, with or without branching

Common example in the web world: checkout process

Some features are present on almost all wizards, including a progress indicator and elements for back/next navigation.

Master-detail

The master-detail navigation pattern means allowing the user to navigation based on the structure and relationships in the data. For example, choosing a customer first, and then choosing an order for that customer.

Drill-down

The most common sub-type in this pattern is drill-down. The difference in drill-down from routine master detail is that the user has more control over what kind of information is displayed next. For example, they might select a customer, and then have the option of selecting one of several child collections for a customer, such as orders, support calls, or payment history.

Indications	*Cautions*
• Users are quite familiar with data structures • User path through the data structure is unpredictable	• Often coupled with search capability to manage the amount of data navigated • Often need a "Find" capability to search within currently fetched data • "Collapse all" or return to root shortcut is usually helpful

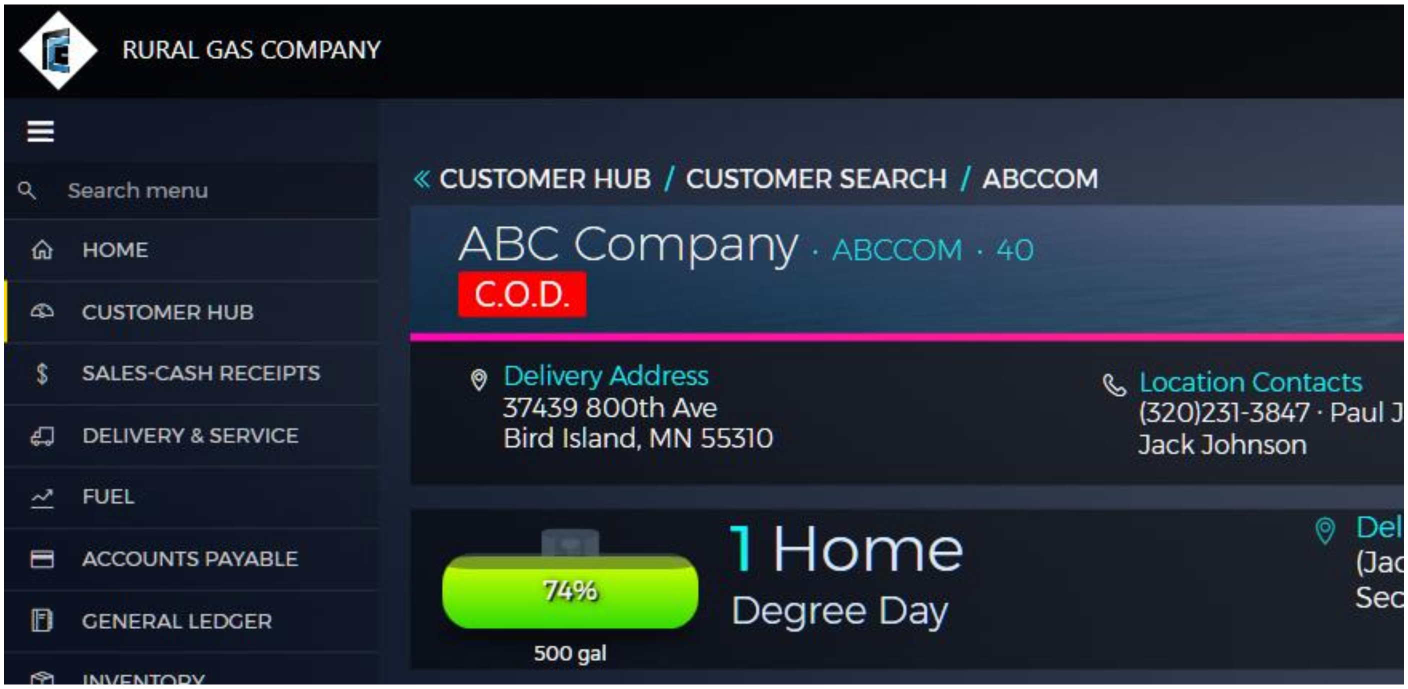
Forward / back linear nav

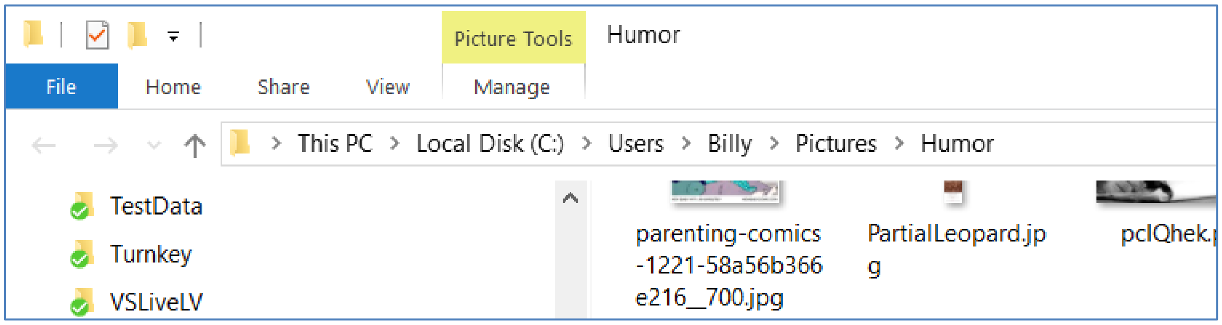
This pattern is familiar to virtually every user you have, because it is used in modern browsers. The pattern can also be helpful in your own application if it is content-centric and consumption oriented. It is usually not a good idea for applications that have a lot of data entry, and is less useful when the information presented is highly structured.

A variant of linear nav is to have a "back" button in the upper left corner of your application, which takes the user to the previously-viewed screen. This is often used in master detail or drilldown scenarios to navigate levels of the data.

Breadcrumb

Another pattern useful with drill down or other hierarchical navigation is the breadcrumb pattern. Usually shown at the top of a view, it shows the levels in the hierarchy that were used to get to the present screen. Usually the levels are clickable, allowing a user to navigate to any previous place in the hierarchy with a single click.

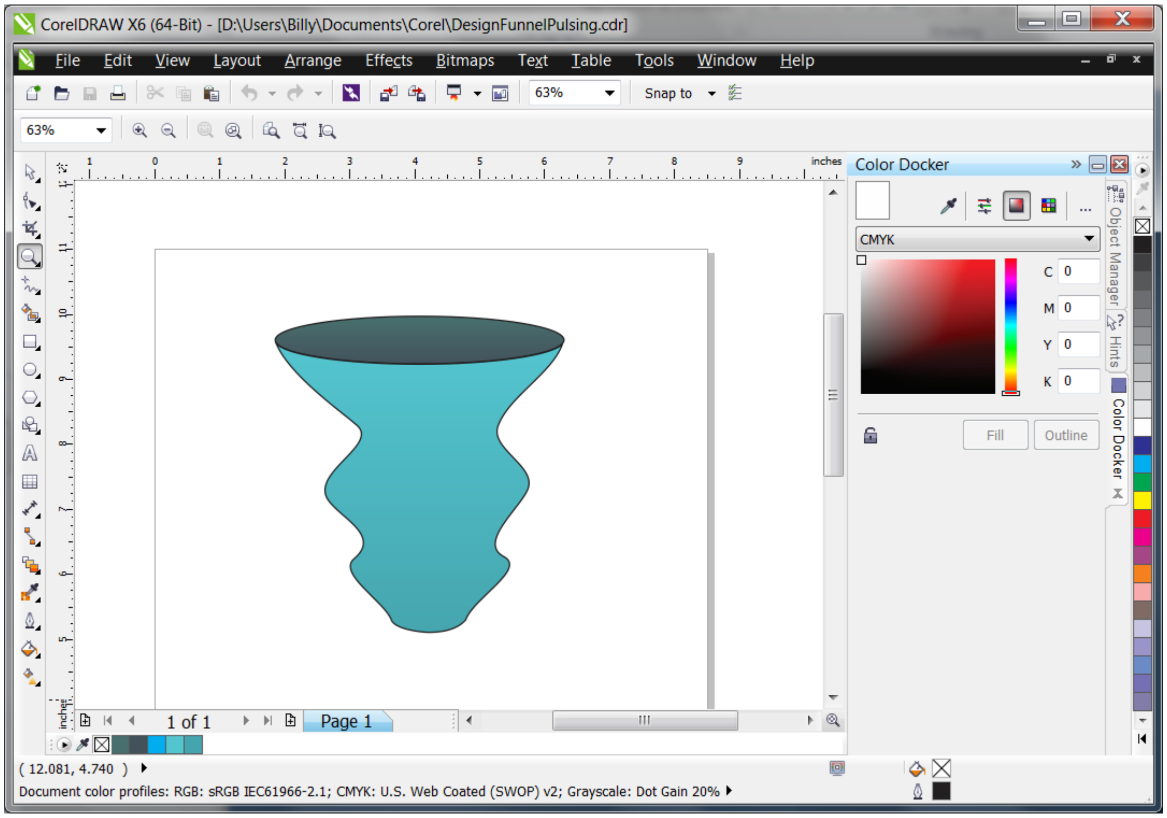



Picture Tools
Humor
File
Home
Share
View
Manage
This PC > Local Disk (C:) > Users > Billy > Pictures > Humor
TestData
Turnkey
VSLiveLV
parenting-comics-1221-58a56b366e216__700.jpg
PartialLeopard.jpg
pclQhek.

Indications	***Cautions***
• Users are quite familiar with the hierarchy involved • User path through the data structure is unpredictable	• Some users can become confused with too many levels • The breadcrumb "names" should be clear to the user

Edit/production view

Most developers see this pattern several hours a day, in the form of Visual Studio. A wide variety of applications use it, with the common thread being that the user is involved in some sort of editing, design, or production task. Editors, spreadsheets, and programs that do visual layout are all in this genre. A typical example would be CorelDraw:


CorelDRAW X6 (64-Bit) - [D:\Users\Billy\Documents\Corel\DesignFunnelPulsing.cdr]
File Edit View Layout Arrange Effects Bitmaps Text Table Tools Window Help
63%
Snap to
Color Docker
CMYK
C 0
M 0
Y 0
K 0
Fill
Outline
Object Manager
Hints
Color Docker
1 of 1
Page 1
inches

(12.081, 4.740)

Document color profiles: RGB: sRGB IEC61966-2.1; CMYK: U.S. Web Coated (SWOP) v2; Grayscale: Dot Gain 20%

Indications	***Cautions***
• User needs detailed control over layout • Including choice of elements to lay out • Task flow extremely hard to predict	• Users must tolerate long learning process • Requires lots of design for the “panes”

Spreadsheets

It's worthwhile to single out spreadsheets from this category. The spreadsheet or data grid pattern is commonly used in business applications, and sometimes over-used.

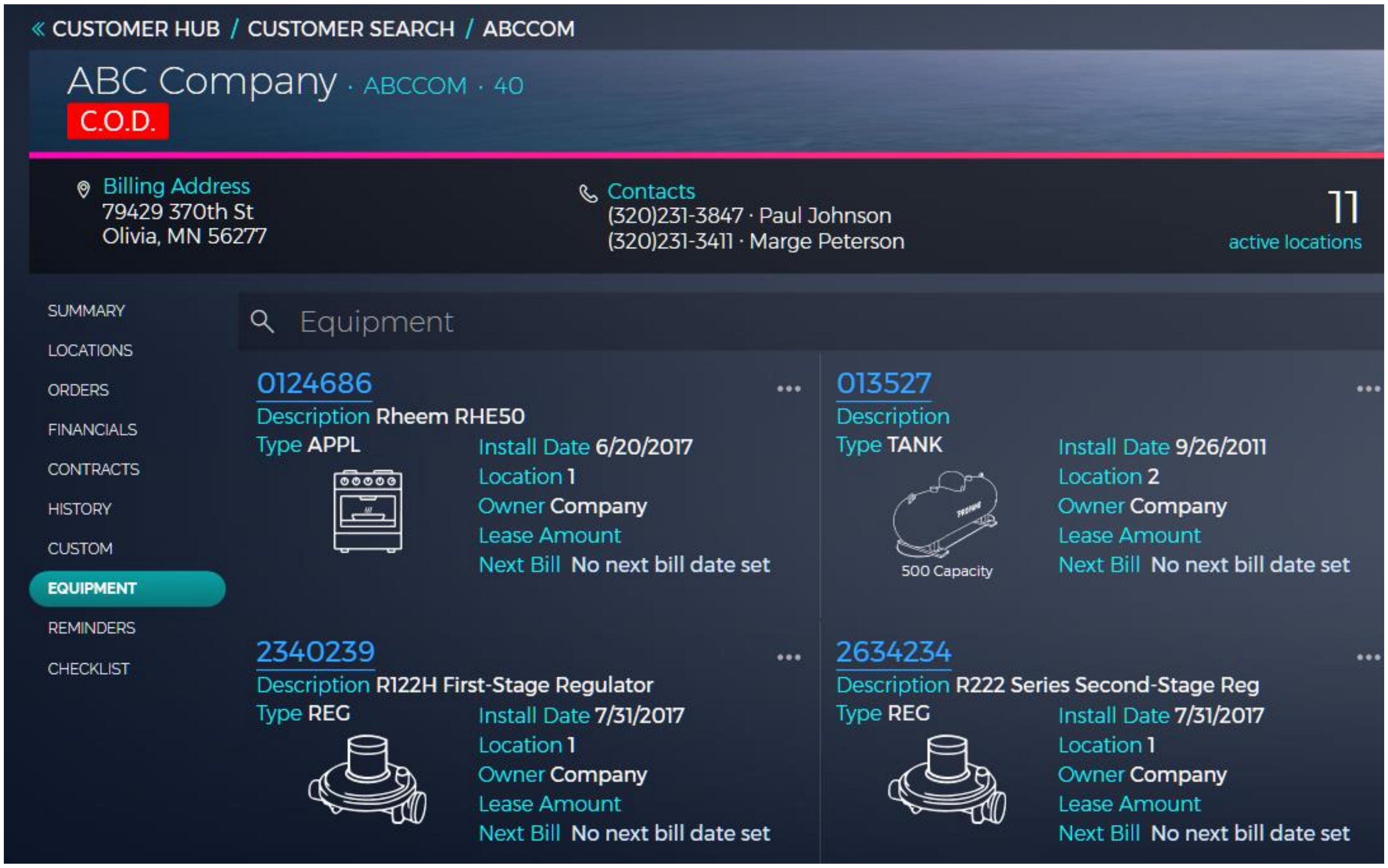
The chief characteristics of the spreadsheet pattern are a flat data structure (rows and columns only), and cell-based editing.

Indications	***Cautions***
• You need flexible sort capability • You need users to have some control over what they view via drag/drop of columns • Your users are accountants • You are too lazy to create a better way to show and change data	• Don't do one out of laziness • Consider a card-view for detail • Modern UI includes data templating capability that works better for the majority of scenarios • If you do use spreadsheet navigation in a XAML app, you can do data templating of cells

Context switchers

The context switcher pattern allows some means of navigation between a number of different views. The maximum number of views for this pattern to work well is 4-10, depending on screen size, type of user, etc.

The most common context switcher used in business application is a tab control. Each view has a tab, and the navigation is accomplished by clicking on the tab. Tabs are often at the top, but for many layouts, tabs on the side work better, as shown below:



ABC Company · ABCCOM · 40
C.O.D.

Billing Address
79429 370th St
Olivia, MN 56277

Contacts
(320)231-3847 · Paul Johnson
(320)231-3411 · Marge Peterson

11
active locations

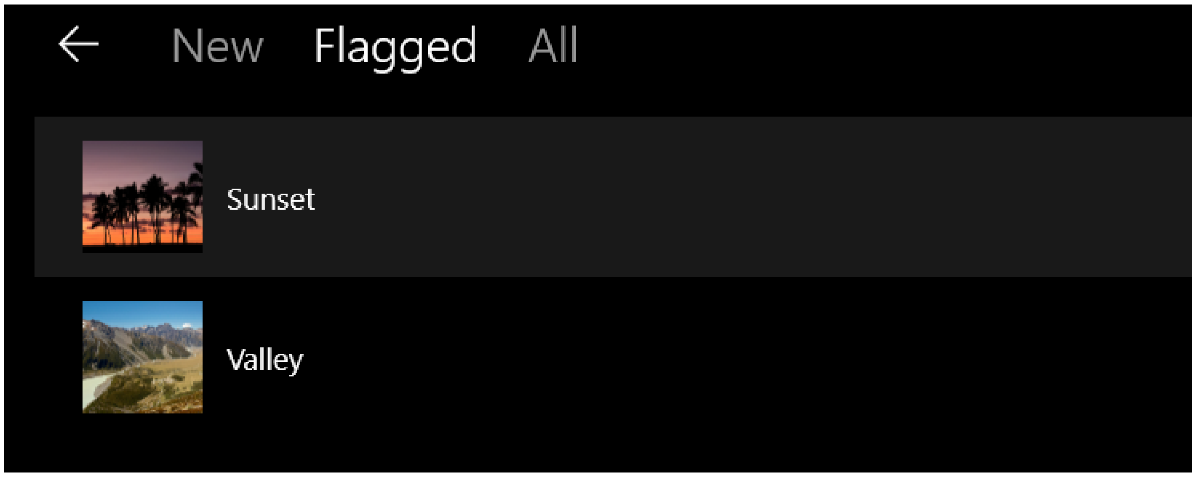
0124686
Description Rheem RHE50
Type APPL
Install Date 6/20/2017
Location 1
Owner Company
Lease Amount
Next Bill No next bill date set

013527
Description
Type TANK
Install Date 9/26/2011
Location 2
Owner Company
Lease Amount
Next Bill No next bill date set

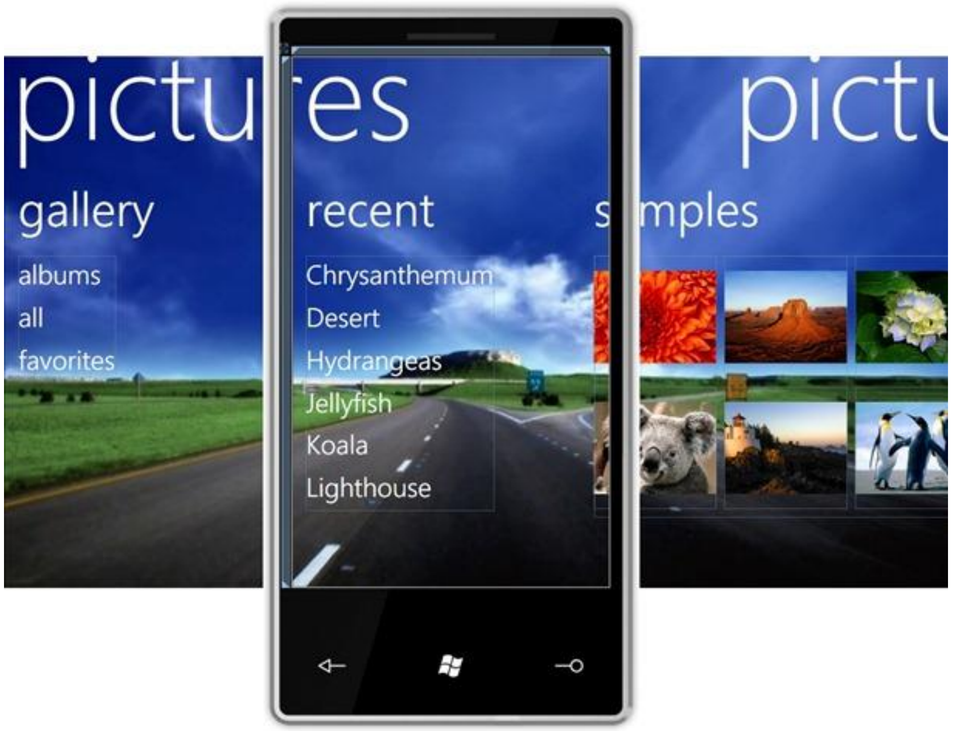
2340239
Description R122H First-Stage Regulator
Type REG
Install Date 7/31/2017
Location 1
Owner Company
Lease Amount
Next Bill No next bill date set

2634234
Description R222 Series Second-Stage Reg
Type REG
Install Date 7/31/2017
Location 1
Owner Company
Lease Amount
Next Bill No next bill date set

A variant of the tab control for Windows 10 / UWP is the Pivot. It has mostly the same dynamics as a classic tab control, but has better touch capabilities and built-in interaction animations.



Other context switchers include a panorama, in which the views are “swished” in a cycle, and a carousel, in which bits of all the views are shown, but one view is in front and predominant. Here is a panorama-based app on the Windows Phone:



The panorama pattern was once common on touch based tablet and phone apps, but has fallen out of favor in the last few years. It works well up to five or six “panels”, or perhaps a bit more on a tablet. Too many panels cause the user to need to swipe too much for the later panels. A variation on the pattern puts context switching capability as part of an app header (similar to a Pivot), so that every panel has a place to use for immediate access.

Indications	***Cautions***
• Application has several categories of functionality that are mostly unrelated • Or application has too much data for one screen and must categorize data elements • You wish to use most of the screen for the mode of functionality the user wants right now	• For data entry apps, changes on hidden views should be easy to find • Don't use as an excuse to cram too much data into an application

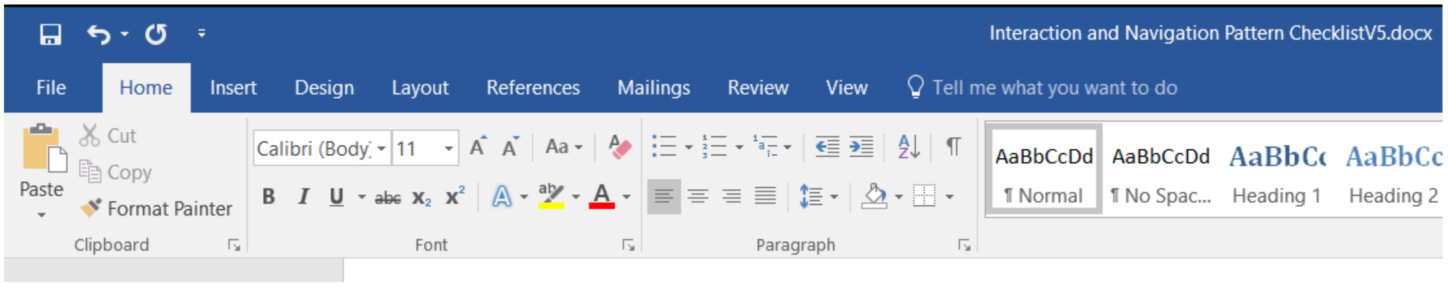
Toolbar / Ribbon

Options on a toolbar or ribbon are usually action oriented, and are primarily for interaction, not navigation. I have seen exceptions in which they were used for navigation; I can't say I was impressed with most of those designs.

There are exceptions to that, however. For example, Powerpoint uses its toolbar for navigation to master views. In that case, since the app requires very minimal navigation/context switching, doing the small amount needed on the toolbar works reasonably well.

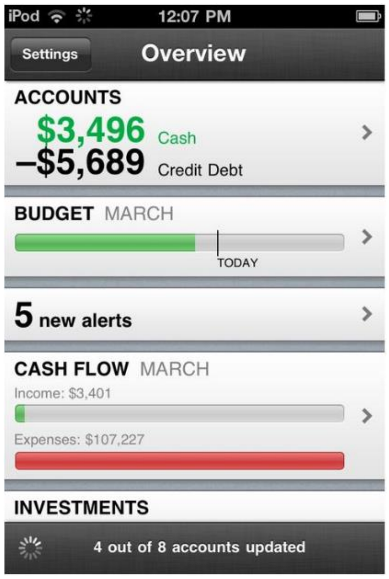
Ribbons and toolbars are often associated with a contextual element, such as a document or view. When you use one, it should be absolutely clear which document or view the ribbon or toolbar affects.

As with other patterns encompassing long lists of options, some sort of search or filter function can make a toolbar more effective. The Office ribbon has been dramatically improved in recent versions with the additions of the "Tell me what you want to do" entry box just above the ribbon.



Dashboard

The dashboard usually combines data visualization with some navigation ability. Drill down on the data is one common navigation option, but a dashboard might also contain an option list (of work items, for instance). Here are a couple of visual examples:



iPod 12:07 PM

Settings **Overview**

ACCOUNTS
$3,496 Cash
–$5,689 Credit Debt

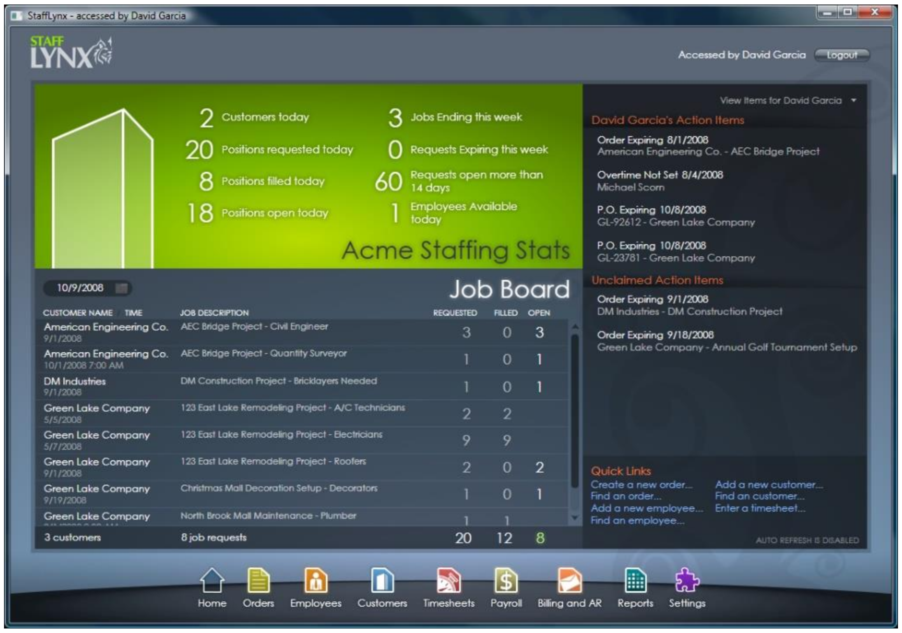
BUDGET MARCH
TODAY

5 new alerts

CASH FLOW MARCH
Income: $3,401
Expenses: $107,227

INVESTMENTS

4 out of 8 accounts updated



StaffLynx - accessed by David Garcia

Accessed by David Garcia Logout

2	Customers today
20	Positions requested today
8	Positions filled today
18	Positions open today
3	Jobs Ending this week
0	Requests Expiring this week
60	Requests open more than 14 days
1	Employees Available today

Acme Staffing Stats

Job Board

10/9/2008

CUSTOMER NAME / TIME	JOB DESCRIPTION	REQUESTED	FILLED	OPEN
American Engineering Co. 9/1/2008	AEC Bridge Project - Civil Engineer	3	0	3
American Engineering Co. 10/1/2008 7:00 AM	AEC Bridge Project - Quantity Surveyor	1	0	1
DM Industries 9/1/2008	DM Construction Project - Bricklayers Needed	1	0	1
Green Lake Company 5/5/2008	123 East Lake Remodeling Project - A/C Technicians	2	2	
Green Lake Company 5/7/2008	123 East Lake Remodeling Project - Electricians	9	9	
Green Lake Company 9/1/2008	123 East Lake Remodeling Project - Roofers	2	0	2
Green Lake Company 9/19/2008	Christmas Mall Decoration Setup - Decorators	1	0	1
Green Lake Company	North Brook Mall Maintenance - Plumber	1	1	
3 customers	8 job requests	20	12	8

View Items for David Garcia

David Garcia's Action Items

Order Expiring 8/1/2008
American Engineering Co. - AEC Bridge Project

Overtime Not Set 8/4/2008
Michael Scorn

P.O. Expiring 10/8/2008
GL-92612 - Green Lake Company

P.O. Expiring 10/8/2008
GL-23781 - Green Lake Company

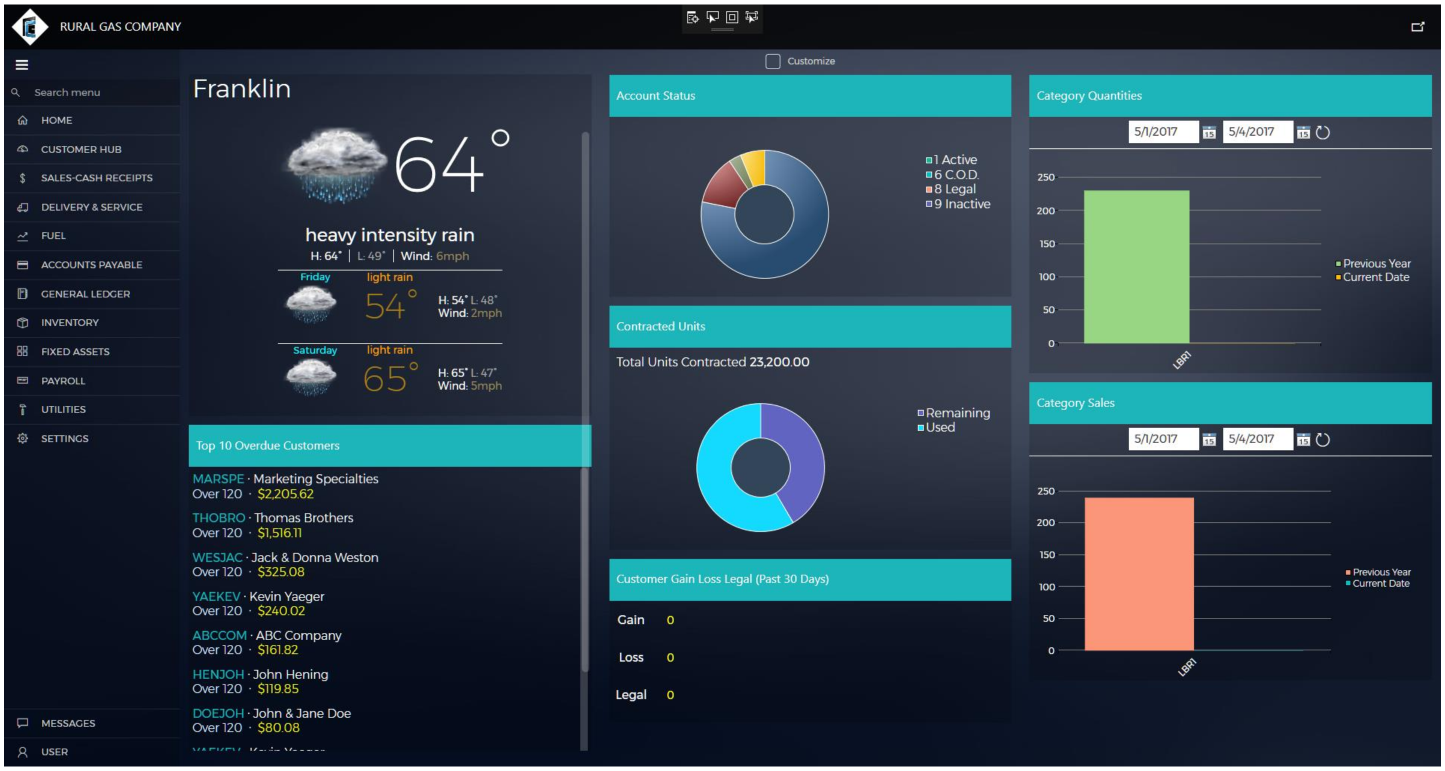
Unclaimed Action Items

Order Expiring 9/1/2008
DM Industries - DM Construction Project

Order Expiring 9/18/2008
Green Lake Company - Annual Golf Tournament Setup

Quick Links
Create a new order...
Find an order...
Add a new employee...
Find an employee...
Add a new customer...
Find an customer...
Enter a timesheet...

AUTO REFRESH IS DISABLED



Franklin

64°
heavy intensity rain
H: 64° | L: 49° | Wind: 6mph

Friday light rain 54° H: 54° L: 48° Wind: 2mph

Saturday light rain 65° H: 65° L: 47° Wind: 5mph

Top 10 Overdue Customers

MARSPE · Marketing Specialties
Over 120 · $2,205.62

THOBRO · Thomas Brothers
Over 120 · $1,516.11

WESJAC · Jack & Donna Weston
Over 120 · $325.08

YAEKEV · Kevin Yaeger
Over 120 · $240.02

ABCCOM · ABC Company
Over 120 · $161.82

HENJOH · John Hening
Over 120 · $119.85

DOEJOH · John & Jane Doe
Over 120 · $80.08

Account Status

Contracted Units

Total Units Contracted 23,200.00

Customer Gain Loss Legal (Past 30 Days)

Gain	0
Loss	0
Legal	0

Category Quantities

5/1/2017 5/4/2017

Category Sales

5/1/2017 5/4/2017

Dashboards may be static, with a predetermined layout, or dynamic, in which the users have some control over which panes appear and how they are laid out in relation to one another. Dynamic dashboards make an application more flexible to a wider range of user, but require more design and development. When implementing a dynamic dashboard, it's a good idea to have a default configuration that is useful to a large subset of your users.

Queue

One of the most underused navigation patterns is the queue. This navigation pattern can have a dramatic impact on user productivity if the workflow is a good match for a queue.

There are two main variations of queues. Either the next item to work on is determined automatically for user, or user chooses from a small list over which they have limited or no control. I usually refer to these types as “push queues” and “pull queues”, respectively.

Examples of good scenarios for a queue:

- A pharmacist approving prescriptions for other workers to pack and distribute.
- A customer service representative handling issues filed by customers
- A salesperson working with prospective deals or customers

Indications	***Cautions***
• Highly structured and repetitive workflow • Multiple users working with same set of pending items	• Build in forgiveness ◦ User gets in rhythm and it’s easy to finish/accept an item just as realization hits that something is incorrect or questionable • Don’t use confirmation instead ◦ “Are you sure?” will quickly become ignored

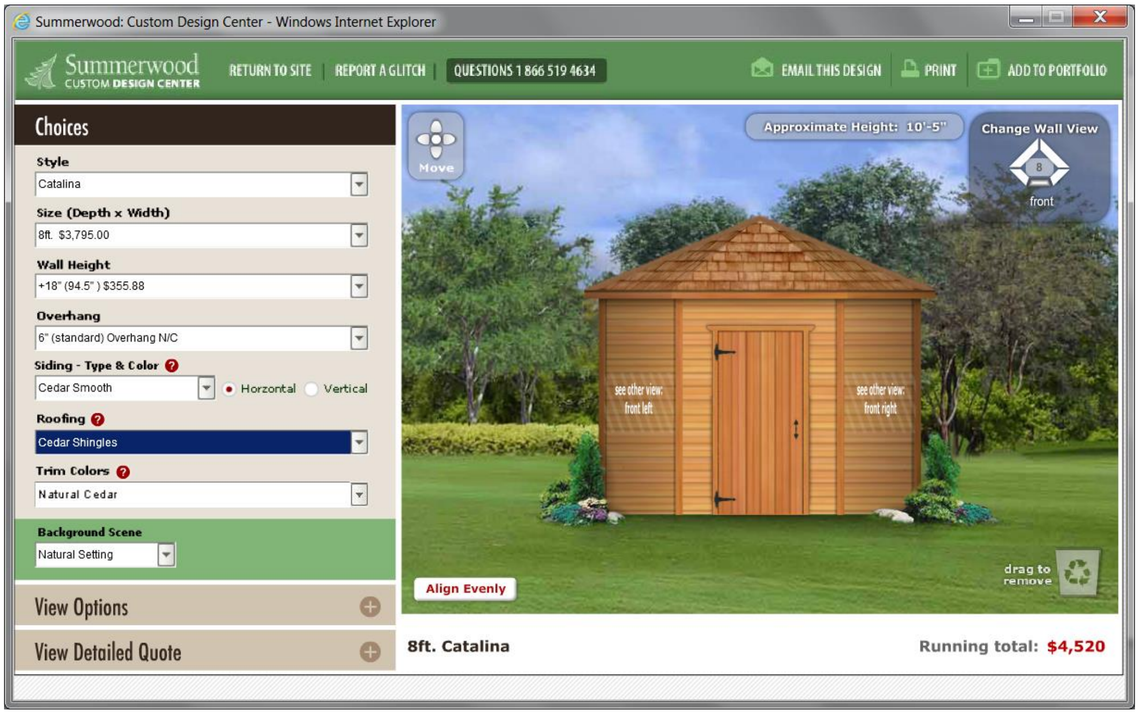
Queuing architecture

The queue navigation/interaction pattern usually has wider impact on overall architecture of the application than other patterns. For example, the choice of queuing technology can influence what user can do, and therefore the queuing technology might need to be chosen based on how this pattern is implemented in the application.

The architect should be familiar with architectural patterns such as a “dispatch queue”. Items tend to move from one queue to another in a workflow. The brute force way to do that is some code to do direct routing from one queue to another. A better architecture for all but the simplest cases is to have a central queue where every item goes when its last queue was finished with it. There, a central rules engine can decide what queue it goes to next, and can even spawn additional queue items if needed (as when an approved order needs to route multiple item shipment work items).

Configurator

A configurator features multiple, independent options affecting a visualizable object. Here is a typical screen→



Style: Catalina
Size (Depth x Width): 8ft. $3,795.00
Wall Height: +18" (94.5") $355.88
Overhang: 6" (standard) Overhang N/C
Siding - Type & Color: Cedar Smooth [✓ Horizontal] Vertical
Roofing: Cedar Shingles
Trim Colors: Natural Cedar
Background Scene: Natural Setting

8ft. Catalina Running total: $4,520

A configurator is conceptually similar to an edit/production view. However, where edit/production views are typically aimed as users who will endure a substantial training time, a configurator is usually aimed at first time or casual users. Thus it must be considerable simpler than an edit/production view pattern, and it usually has far fewer options for the user to access. It also often dispenses with any drag/drop or direct editing in the designer view, whereas an edit/production view will usually include those features.

Timeline

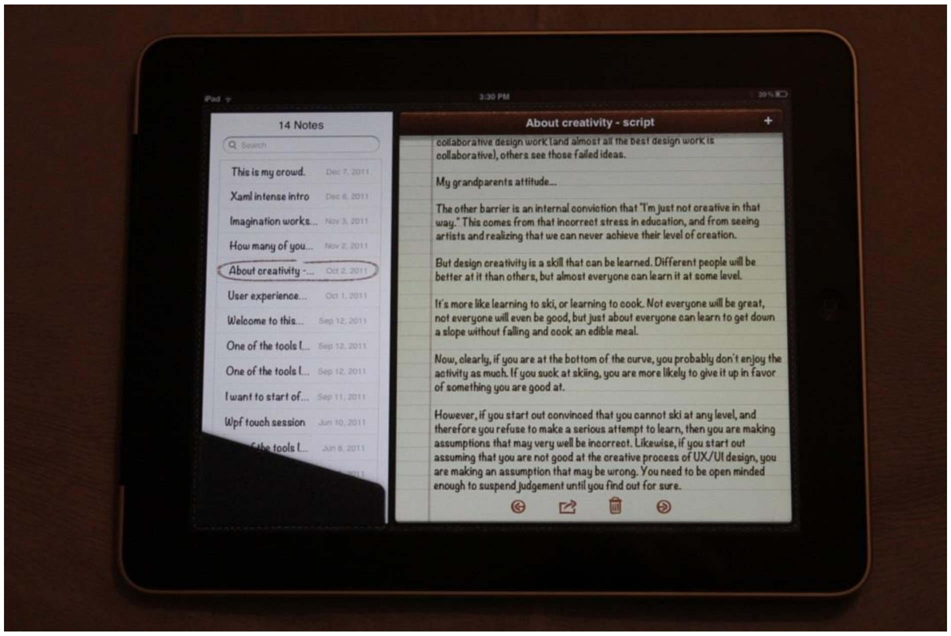
Anyone who has used Outlook or a similar program is familiar with a timeline. Whenever the items being manipulated have a significant time-sensitive aspect, a timeline should be considered as a navigation pattern.

Timelines can be difficult to implement. Most platforms offer third party “controls” that do much of the work.

Don’t forget that timelines can have a horizontal layout. Since Outlook has a vertical layout for the timeline, it’s easy to fall into the default of using that layout for all timelines. But depending on the nature of the data, a horizontal timeline can work better, especially on touch-based systems. Video editing software is an example where a horizontal timeline is usually used.

Metaphor

With the Metaphor interaction pattern, the interface resembles something familiar in the real world. When done well, that can make an interface intuitively obvious to first-time users. Here is the notepad app on the original iPad:



This is my crowd.	Dec 7, 2011
Xaml intense intro	Dec 6, 2011
Imagination works...	Nov 3, 2011
How many of you...	Nov 2, 2011
About creativity -...	Oct 2, 2011
User experience...	Oct 1, 2011
Welcome to this...	Sep 12, 2011
One of the tools I...	Sep 12, 2011
One of the tools I...	Sep 12, 2011
I want to start of...	Sep 11, 2011
Wpf touch session	Jun 10, 2011
the tools I...	Jun 8, 2011

About creativity - script

collaborative design work (and almost all the best design work is collaborative), others see those failed ideas.

My grandparents attitude...

The other barrier is an internal conviction that "I'm just not creative in that way." This comes from that incorrect stress in education, and from seeing artists and realizing that we can never achieve their level of creation.

But design creativity is a skill that can be learned. Different people will be better at it than others, but almost everyone can learn it at some level.

It's more like learning to ski, or learning to cook. Not everyone will be great, not everyone will even be good, but just about everyone can learn to get down a slope without falling and cook an edible meal.

Now, clearly, if you are at the bottom of the curve, you probably don't enjoy the activity as much. If you suck at skiing, you are more likely to give it up in favor of something you are good at.

However, if you start out convinced that you cannot ski at any level, and therefore you refuse to make a serious attempt to learn, then you are making assumptions that may very well be incorrect. Likewise, if you start out assuming that you are not good at the creative process of UX/UI design, you are making an assumption that may be wrong. You need to be open minded enough to suspend judgement until you find out for sure.

This picture could be mistaken at first glance for a real notebook.

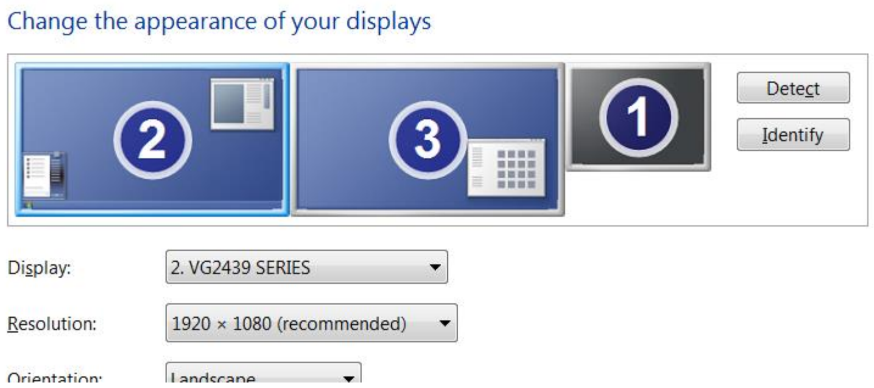
It's possible to go overboard in a metaphor, and include unnecessary details from the real-world equivalent. This is sometimes called a "skeuomorph". However, the essence of using this navigation pattern is that the metaphor makes the interaction intuitive to a first time user. You should normally only include enough detail for the metaphor to be obvious and intuitive.

Skeuomorphic metaphor designs have fallen out of favor in recent years. If you use one today, your users may consider it obsolete or even cheesy.

Map

The Map pattern applies when working with items laid out in the real world. Certainly geographical layout applies, and the various Internet mapping options show this pattern in use. But it can be used in more restricted ways too. For example, an application might use the layout of hospital rooms on a floor to allow the nurses to navigate among patients, or a warehouse app might use a map of the warehouse to allow the use to select the desired location of an item.

You have used a non-geographical example of the Map pattern in working with Windows:



Change the appearance of your displays

Display:	2. VG2439 SERIES
Resolution:	1920 × 1080 (recommended)
Orientation:	Landscape

The appearance of the screens is associated with both the size and position of the real-world monitors that show them. This makes it easy for the user to just drag things around until the virtual arrangement of screens matches the real-world arrangement of screens.

Viewport

Geographical mapping applications on the Internet, such as Google Maps and Bing Maps, show another navigation/interaction pattern than can be an excellent fit for certain less common scenarios – the Viewport. The Viewport pattern typical has these characteristics:

- Interaction area is larger than the screen
- User can pan to reposition the “viewport” (screen) onto the section they want
- Zooming often also supported

Besides maps, certain types of data visualization can effectively use viewports, with the user panning to find the desired data and then zooming to drill down for detail.

This pattern works very well in a touch environment, because of the ready access of gestures for panning and zooming.

Combining interaction/navigation patterns

These navigation patterns are not mutually exclusive.

One app may use one pattern at one interaction point, and another one at another interaction point.

Example: Launchpad when first open app to choose main functional area, then another pattern such as a menu or option list for the next stage of interaction. Or a menu of options to get to a functional area, and then a breadcrumb within the functional area.

*To inquire about a class for your organization on design principles and design processes,
contact Billy: billyhollis@live.com*