rethinking software design

Daniel Jackson · CSAIL · MIT

SATURN Conference · San Diego · May 4, 2016
what is software design?
### Designers & Engineers

<table>
<thead>
<tr>
<th></th>
<th>Designer</th>
<th>Engineer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elements</td>
<td>door, window, wall</td>
<td>column, beam, truss</td>
</tr>
<tr>
<td>Goals</td>
<td>comfortable, convenient, attractive</td>
<td>structural integrity, durable, sustainable</td>
</tr>
</tbody>
</table>
designers & engineers

<table>
<thead>
<tr>
<th>elements</th>
<th>designer</th>
<th>engineer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>?</td>
<td>function, object, type</td>
</tr>
</tbody>
</table>

| goals       | learnable, effective, tolerant | maintainable, correct, fast |
## The Role of Design Criteria

<table>
<thead>
<tr>
<th>Elements</th>
<th>Designer</th>
<th>Engineer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>?</td>
<td>function, object, type</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Designer</th>
<th>Engineer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>?</td>
<td>decoupling &amp; localization</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Goals</th>
<th>Designer</th>
<th>Engineer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>learnable, effective, tolerant</td>
<td>maintainable, correct, fast</td>
</tr>
</tbody>
</table>
WARNING

OBSESSIVE DETAILS AHEAD
That's quite obsessive, isn't it?
Jonathan Ive in “Objectified”

The details are not the details.
These make the design.
attributed to Charles Eames by Garrett
Someone accidentally deleted thousands of files in my company Dropbox: how can I quickly undelete them?
survey of MIT Dropbox users

correctly predicting behavior

- good knowledge
- average knowledge
- poor knowledge

Kelly Zhang

- delete shared folder results in leaving
- delete shared subfolder removes it
3
software design problems
#1

email categories
<table>
<thead>
<tr>
<th>Primary</th>
<th>Social</th>
<th>Promotions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google</td>
<td>New sign-in from Chrome on Mac - New sign-in from Cr</td>
<td>12:30 pm</td>
</tr>
<tr>
<td>Keith Muhammad at DeMont.</td>
<td>DeMontrond Auto Group - 14101 North Freeway Houston</td>
<td>12:19 pm</td>
</tr>
<tr>
<td>AT&amp;T High Speed Internet.</td>
<td>AT&amp;T High Speed Internet Service Activation - Your Account</td>
<td>10:37 am</td>
</tr>
<tr>
<td>Keith Muhammad at DeMont.</td>
<td>DeMontrond Auto Group - 14101 North Freeway Houston</td>
<td>Aug 26</td>
</tr>
<tr>
<td>betterbatonrougejobs.com</td>
<td>Job Update -- 2015-08-26 - Looking For An Advantage</td>
<td>Aug 26</td>
</tr>
</tbody>
</table>
category tab settings

Categories:
- Primary
- Social
- Promotions
- Updates
- Forums

Starred messages
- Include starred in Primary

Choose which message categories to show as inbox tabs. Other messages will appear in the Primary tab.
some reactions

Google gets it terribly, terribly wrong with Gmail tabs and makes me angry

By Mark Wilson | Published 2 years ago | Follow

71 Comments

Like many Gmail users, I greeted the news of the introduction of tabs to the interface with a degree of anticipation -- now it was just a matter of waiting for the feature to roll out so I could try it for myself. Earlier today I was randomly signed out of my Gmail account, and after signing back in and checking the settings menus, I could see that tabs were now available to me. Excitement was short-lived, however; it quickly became apparent that this new feature is a disaster.
Why the Tabs Are So Confusing

The tabs do give you a convenient way to automatically sort your inbox according to Gmail's preset categories and get notifications at a glance for when new emails come in.

However, the tabbed view introduces a new, not-so-clear element in Gmail called "categories."

The problem is, you already have labels that also match these categories. Gmail, for example, adds Social Updates, Promotions, and Forums as SmartLabels in the left menu. The "Notifications" SmartLabel corresponds to the Updates tab (I wonder why they didn't just call the tab Notifications), but **SmartLabels are not the same as Categories.** Gmail's pre-designed SmartLabel filters makes it seem like that, but they're really two different things.
Using labels

Labels help you organize your messages into categories -- work, family, to do, read later, jokes, recipes, any category you want. Labels do all the work that folders do, but with an added bonus: you can add more than one to a message.
what you can’t do

associate tabs with labels
feature available only for categories

use tabs outside inbox
tabs disappear when you filter on a label
#2 camera settings
my camera fuji x100s
image quality setting

- FINE
- NORMAL
- FINE+RAW
- NORMAL+RAW
- RAW
aspect ratio
image size setting
non-standard ratio + raw?
what you can't do

non-standard aspect ratio + raw
even though raw images get nice nondestructive crop!
#3 fonts & styles
what's a font?
what you can’t do

define a style that italicizes
   Arno Regular to Arno Italic
   Futura Book to Futura Book Oblique
   Magma Light to Magma Light Italic
what kind of problems are these?

minor nitpicks?
in all cases, can’t do useful things

coding bugs?
code seems to meet a coherent spec

user interface flaws?
interfaces are clear and faithful to function
getting to the essence of an app
what characterizes an app?

concepts!

Apple Mail
- Email Address
- Message
- Folder or Label

Microsoft Word
- Paragraph
- Format
- Style

Twitter
- Tweet
- Hashtag
- Following

Photoshop
- Pixel Map
- Layer/Mask
- Adjustment
concepts define classes

text editor
  line
  buffer

word processor
  paragraph
  format
  style

desktop publishing app
  stylesheet
  text flow
  page template

Just upgraded to the new Pages and can't find a way to link text boxes anymore like
http://www.macobserver.com/tmo/article/pages-linking-text-boxes
Am I missing something, or is it really not possible anymore?
where are Word’s concepts from?

Charles Simonyi: brought key concepts to Word from Xerox PARC
rich concepts have long journeys

Ginn & Co, since 1868

Apple Pages, 2005

Bravo, 1974

Microsoft Word, 1983
kinds of concept

even these were invented
given concepts
- electoral vote
- reservation
- social security number
- calendar event

analogical, or new
instrumental concepts
- hashtag
- label
- friend
- follower

solve a hard design problem
enabling concepts
- relative reference
- public key
- layer mask
- conference call
how to sketch a concept
how would you explain this?
the operational principle
a way to explain a concept

an archetypal scenario
separates essential from accidental aspects
shows how purpose is fulfilled
by combination of user & system actions

“if you pull a tab out, then when that time slot comes around, the light will go on”

Michael Polanyi
“if you change a style’s format, then all paragraphs of that style will change format accordingly”

“if you select some files and they belong to a folder with keyboard focus, then pressing delete will move the files to the trash”

“if you pull a tab out, then when that time slot comes around, the light will go on”

“if you tag a photo, then all friends of the person tagged will be able to see the photo”
purposes, principles & misfits

concept: trash

**purpose**: allow undo of deletions

**operational principle**: if you delete a file, it moves to a special folder; you can restore from there, but emptying it removes contents for good (and makes space on disk)

**misfit**: if you delete a file on an external drive, you cannot reclaim the space until you empty the trash, but then you’ll lose the ability to restore files deleted from the main drive

**misfit**: if you delete an old file and change your mind, you may not be able to find it again in the trash (if there are many deleted files and you forgot the file’s name)
how to model a concept
example word styles
There is no problem in computer science that cannot be solved by introducing another level of indirection.  
David Wheeler

operations
add  
delete  
modify  
update  
apply
Concept dependences

\[ \langle c, c' \rangle \in \text{depends} \iff \forall a: \text{apps} \cdot c \in \text{concepts}(a) \Rightarrow c' \in \text{concepts}(a) \]
how to reuse a concept
other instantiations style

Powerpoint schemes

Indesign swatches
non-instantiations style

Apple color picker
swatch-color mapping fixed

Write 2 text editor
no para-style mapping
## generic concept parts

<table>
<thead>
<tr>
<th>part</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td><strong>Style</strong></td>
</tr>
<tr>
<td>purpose</td>
<td>make it easy to maintain consistent format across set of elements</td>
</tr>
<tr>
<td>sample uses</td>
<td>Text formatting in word processors/layout apps (Word, Indesign, Pages, etc); rules in CSS; color themes in Powerpoint.</td>
</tr>
<tr>
<td>model</td>
<td><em>base, add-ons, variants</em></td>
</tr>
<tr>
<td>operations</td>
<td>add, delete, modify, apply, update</td>
</tr>
<tr>
<td>related to</td>
<td><strong>StyleBuffer, Stencil, Master</strong></td>
</tr>
<tr>
<td>issues</td>
<td>optional rules problem</td>
</tr>
</tbody>
</table>
concept selection

- slides in Keynote
- photos in Adobe Lightroom
- messages in Apple Mail
- objects in OS X Finder
- thumbnails in Preview
- notes in Evernote
subtlety selection scope

subset of selection in scope

subset of selection out of scope
Adobe Lightroom: brightest thumbnail is the “active photo”
subtlety continuous selection

Photoshop: outline shown with “marching ants”

Photoshop: selection shown in Quick Mask mode
subtlety folder selection

Google Drive: selecting folder = selecting children

CrashPlan: selecting folder = selecting all future children
concept selection

Purposes: apply action in aggregate to many items at once
<table>
<thead>
<tr>
<th>instantiate</th>
<th>organize</th>
<th>relate</th>
<th>resource</th>
<th>save</th>
<th>communicate</th>
<th>personalize</th>
</tr>
</thead>
<tbody>
<tr>
<td>stylesheet</td>
<td>selection</td>
<td>friend</td>
<td>access token</td>
<td>history</td>
<td>message</td>
<td>account</td>
</tr>
<tr>
<td>master</td>
<td>folder</td>
<td>clique</td>
<td>notification</td>
<td>buffer</td>
<td>posting</td>
<td>karma</td>
</tr>
<tr>
<td>stencil</td>
<td>group</td>
<td>invitation</td>
<td>reservation</td>
<td>cursor</td>
<td></td>
<td>OOBA</td>
</tr>
<tr>
<td>style buffer</td>
<td>label</td>
<td></td>
<td>REST</td>
<td>sync</td>
<td></td>
<td>rating</td>
</tr>
<tr>
<td></td>
<td>layer</td>
<td></td>
<td>cart</td>
<td></td>
<td></td>
<td>status</td>
</tr>
<tr>
<td></td>
<td>stack</td>
<td></td>
<td>subscription</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>alias</td>
<td></td>
<td>purchase order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>preset</td>
<td></td>
<td>RMA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cursor</td>
<td></td>
<td>coupon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>filter</td>
<td></td>
<td>catalog</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>property</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>metadata</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
how to evaluate a concept
the fundamental principle

in a well-designed system
each concept is motivated by one purpose
the ideal mapping

purposes

P1

concepts

C1

P2

C2
4 bad smells

unfulfilled purpose

P1 → C1

unmotivated concept

P1 → C1

overloaded concept

P1 ← C1

P2

P1

C1

redundant concepts

P1 ← C1

P2

C2

C2
Adobe Indesign: an unfulfilled purpose

allow
typeface
independent
styling

subfamily
unfulfilled purposes (more)

user (Apple Mail, Gmail)
‘identify parties to communication’
weak search, no authentication

slide hierarchy (Powerpoint)
‘structure slides in a tree’
sections provide just one level

eject-after-import (Lightroom)
‘prevent accidental writing’
feature removed (and reinstated!)

binder (Preview, Acrobat)
‘maintain composite PDF doc’
can insert pages, but forgets source
unmotivated concepts

You never poke me on Facebook anymore.
unmotivated concepts (more)

buffer (text editors)
Apple got rid of it

stash (Git)
addresses branching misfits

boxing (Java)
addresses primitives vs objects misfit

null (Javascript)
when undefined is too vague
redundant concepts

classify messages

label

category

category in Gmail

a redundant concept
redundant concepts in acrobat

text object, text box, document text (Acrobat)
all subtly different (Acrobat 10)
merged into one (Acrobat 11)
overloaded concepts

No one can serve two masters. Either you will hate the one and love the other, or you will be devoted to the one and despise the other. [Matthew 6:24]

3 forms of overloading:
- **piggybacking** new purpose hacked onto old concept
- **false convergence** two purposes looked the same
- **emergent purpose** users found second purpose for concept
piggybacking fuji camera
new purpose hacked onto old concept

image size
aspect ratio piggybacked
on JPEG dimensions
piggybacking epson driver

result: can’t create custom size for front loading
also, page size presets in Lightroom hold feed setting
false convergence
two purposes looked the same

---

**Performance Review Form 2016**

**Performance Summary**

Evaluate and discuss the employee’s performance. Base your evaluation on the position requirements, MIT Core Competencies, achievement of the goals established during the past year, and your assessment of the employee’s accomplishments.

**Goal Setting**

State and discuss the expectations and goals for the upcoming review period. Give examples of how these goals can be met (e.g., training). How will you support the employee to accomplish these goals? Reference here

---

evaluation & goal setting
incompatible purposes
false convergence

two purposes looked the same

filter incoming posts
control access to my posts

**distinct purposes**

2011: Facebook added
subscribe/follow
emergent purpose
users find second purpose for concept

To: Daniel Jackson <dnj@mit.edu>
Re: Catch me if you can in real life!
initial purpose: summarize content

To: csail-related@lists.csail.mit.edu
Re: [csail-related] turn off the lights?
emergent purpose: show sender
if you bcc a list, subject reveals to-address
thanks to Shriram Krishnamurthi

To: Daniel Jackson <dnj@mit.edu>
your trip reservation
emergent purpose: group by conversation
can’t label reservations from Expedia by trip
thanks to Eunsuk Kang
gitless: a case study
This is Git. It tracks collaborative work on projects through a beautiful distributed graph theory tree model.

Cool. How do we use it?

No idea. Just memorize these shell commands and type them in sync up. If you get errors, save your work elsewhere, delete the project, and download a fresh copy.
gitless: a reworking of git

Gitless: a version control system

About Gitless

Gitless is an experimental version control system built on top of Git. Many people complain that Git is hard to use. We think the problem lies deeper than the user interface, in the concepts underlying Git. Gitless is an experiment to see what happens if you put a simple veneer on an app that changes the underlying concepts. Because Gitless is implemented on top of Git (could be considered what Git pros call a "porcelain" of Git), you can always fall back on Git. And of course your coworkers you share a repo with need never know that you're not a Git aficionado.

Check out the documentation to get started. If you are a novice user that never used any version control system the documentation should be enough to get you started. If you are a Git pro looking to see what's different from your beloved Git you'll be able to spot the differences by glancing through the Gitless vs. Git section.

Download

- Mac OS X Binary (.tar.gz)
- Linux Binary (.tar.gz)
- Source Code (.tar.gz)

For installation instructions see the readme file. After installation, you should be able to execute the gl command. The current Gitless version is 0.7 which was released on 4/2015 (release notes).
example: branch

**concept:** branch

**purpose:** support independent line of development

**operational principle:** when you switch branches, your working directory is synchronized with the new branch, and you can make and commit changes which will be invisible on other branches; when you’re done, you can merge the branch into the master branch...

**misfit:** can’t switch branches with uncommitted changes; can stash, but only if no conflicts...

**fix:** give branch its own working directory; when you switch branches, the working directory changes too, and the working directory associated with the previous branch is preserved
results of a user study

I enjoyed using Gitless
I found Gitless to be easier to learn than Git
I found Gitless to be easier to use than Git
I would continue using Gitless if I could

Task completion time (minutes)

<table>
<thead>
<tr>
<th>Git</th>
<th>Gitless</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Percentage

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Disagree Somewhat</th>
<th>Neutral</th>
<th>Agree Somewhat</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>9%</td>
<td>0%</td>
<td>9%</td>
<td>18%</td>
<td>18%</td>
<td>9%</td>
<td>64%</td>
</tr>
<tr>
<td>0%</td>
<td>9%</td>
<td>5%</td>
<td>27%</td>
<td>27%</td>
<td>9%</td>
<td>64%</td>
</tr>
<tr>
<td>18%</td>
<td>18%</td>
<td>9%</td>
<td>36%</td>
<td>9%</td>
<td>27%</td>
<td>73%</td>
</tr>
<tr>
<td>36%</td>
<td>18%</td>
<td>5%</td>
<td>27%</td>
<td>18%</td>
<td>9%</td>
<td>55%</td>
</tr>
</tbody>
</table>
conclusions
<table>
<thead>
<tr>
<th>elements</th>
<th>designer</th>
<th>engineer</th>
</tr>
</thead>
<tbody>
<tr>
<td>concepts</td>
<td>purposes</td>
<td>function, object, type</td>
</tr>
<tr>
<td>criteria</td>
<td>1:1 = concept:purpose</td>
<td>decoupling &amp; localization</td>
</tr>
<tr>
<td>goals</td>
<td>learnable, effective, tolerant</td>
<td>maintainable, correct, fast</td>
</tr>
</tbody>
</table>
But how to choose increment?

How to structure design step?
a software design approach

- users
  - needs of structured
  - motivate
  - built by
  - evaluate
  - criteria

- purposes
  - drive
  - related by
  - suggest
  - patterns

- increments
  - organize
  - dependences

- developers
  - concepts

- concepts
  - needs of structured
  - motivate
  - built by
  - evaluate
  - criteria
  - patterns
a common view of software design

UI design
soft & human
about presentation

programming
hard & technical
about content
a better view of software design

conceptual design:
essential concepts & behavior

representation design:
organization & performance