

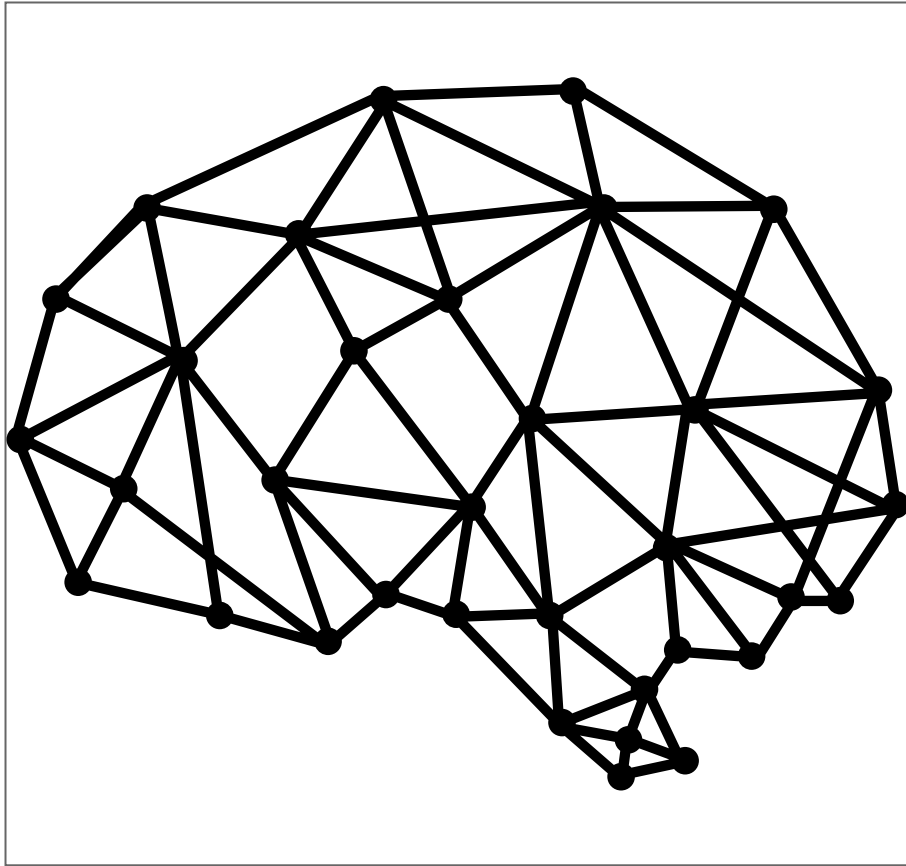
The Naming Of Things

Ben Buchanan

@200okpublic | designops.expert

The two hardest problems in computer science

- cache invalidation
- naming things
- off-by-one errors



Created by Sergey Patutin from the Noun Project

Maluma & Takete (1929)

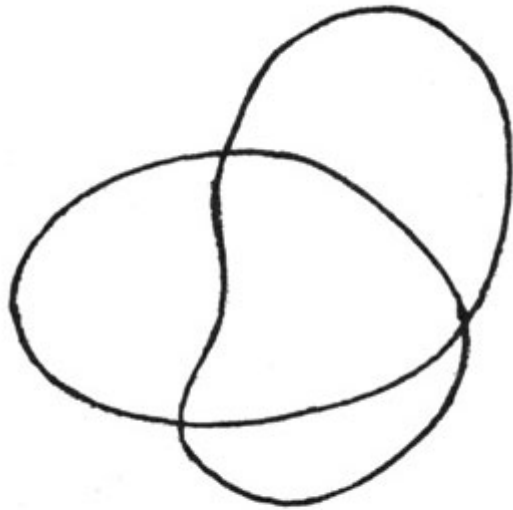


FIG. 18

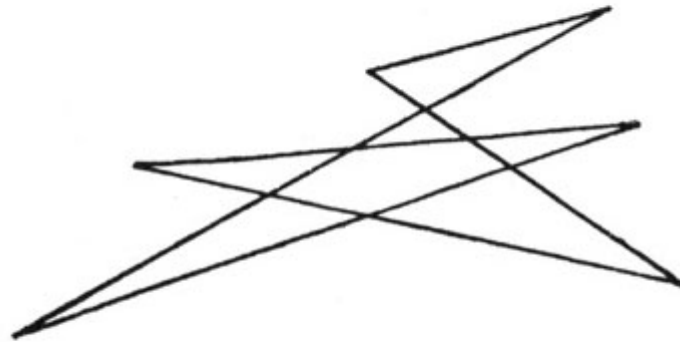
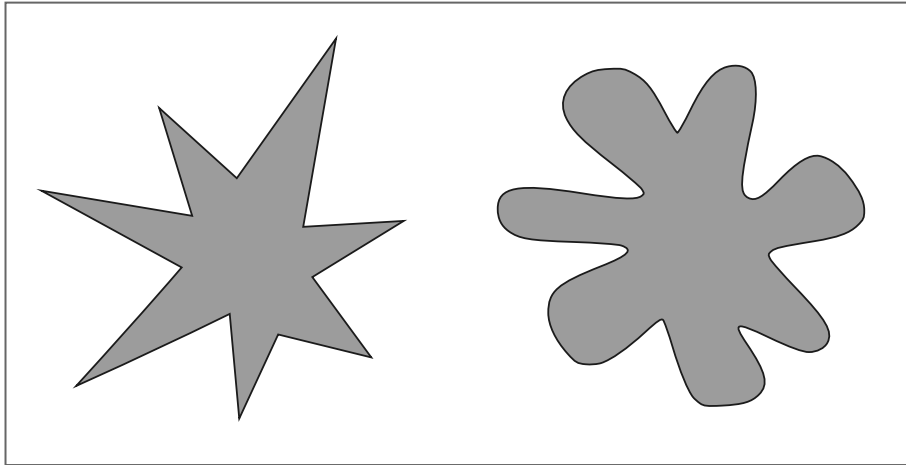


FIG. 19

Kiki & Bouba (2001)



Research suggests people **alter their appearance**
to match cultural expectations about their name.

eg. we imagine Bob has a rounder face than Tim.

Bob



Tim



People react to names **subconsciously**

...so if they are arguing about a name,
they may struggle to express exactly why.

Our design system naming should **match expectations**.

...but **whose expectations?**

All stakeholders?

- Business
- Designers
- Developers
- Users

Users don't know or care

Web applications, like sausages, cease to inspire respect in proportion as we know how they are made.

- John Godfrey Saxe

Stakeholders

- Business
- Designers
- Developers

Things to name

- Design system
- UI library
- Components
- Patterns
- Design assets
- Code

Key Stakeholders

Thing	Business	Design	Development
Design system	✓	✓	✓
UI library	✓	✓	✓
Components		✓	✓
Patterns		✓	✓
Design assets		✓	
Code assets			✓

Two naming concerns

1. Brand
2. Implementation

Brand

Thing	Business	Design	Development
Design system	✓	✓	✓
UI library	✓	✓	✓
Components		✓	✓
Patterns		✓	✓
Design assets		✓	
Code assets			✓

Branding the system

- Reflect your culture
- Differentiate design and implementation
- Short and easy to say
- Beware of phonetics and pisstakes

Example: Quantum

- **Quantum Design System (QDS)**
- Library is **Qbit**
- Components are **Qbits**
- Patterns are **Qpieces**

Implementation

Thing	Business	Design	Development
Design system	✓	✓	✓
UI library	✓	✓	✓
Components		✓	✓
Patterns		✓	✓
Design assets		✓	
Code assets			✓

Naming for implementation

Naming for implementation

- Team names, wiki spaces
- Sketch/Figma projects, symbols, layers, mockups
- Code repos, builds, websites, deployment packages, file names, namespaces, templates, selectors, variables, props, functions, exports, imports, type definitions, test suites, test names, test props and selectors, page objects

Critical for communication

Thing	Business	Design	Development
Design system	✓	✓	✓
UI library	✓	✓	✓
Components		✓	✓
Patterns		✓	✓
Design assets		✓	
Code assets			✓

Designers & Developers

- Conflicting domain knowledge
- Both sides 'know' they are 'right'

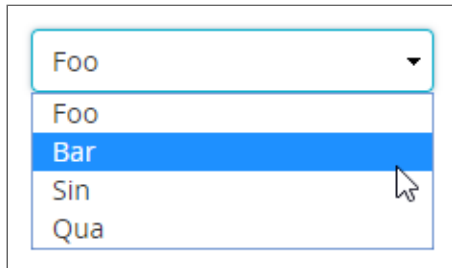
Collaboration beats 'winning'



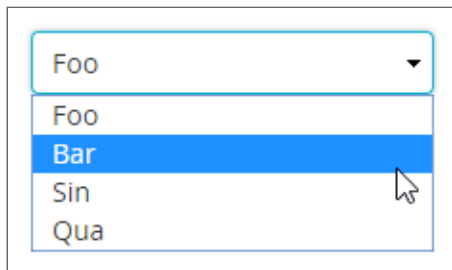
Common conundra

- Menu, Dropdown or Select?
- Dialog, Modal, Modal Dialog?

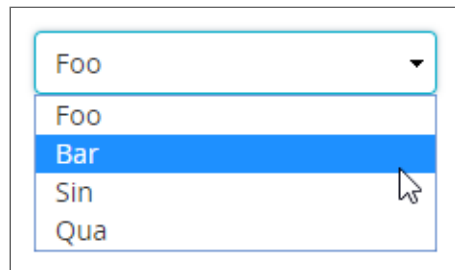
Menu? Dropdown? Picker?



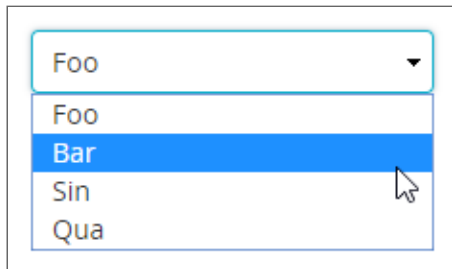
Menu?



Menu?



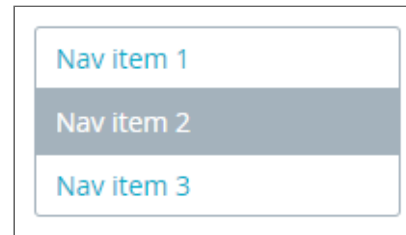
Select



Dropdown



Nav

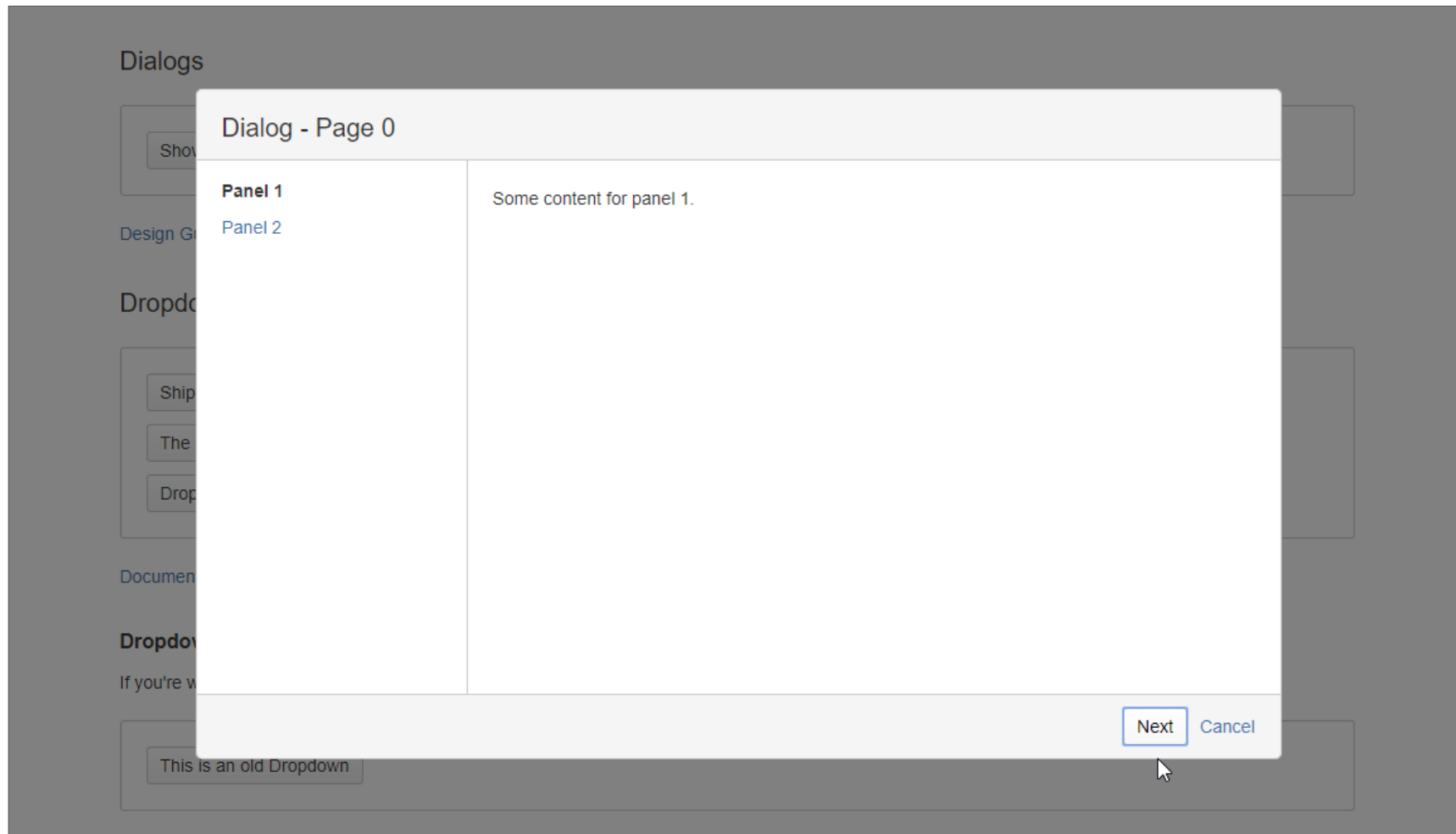


The conversation

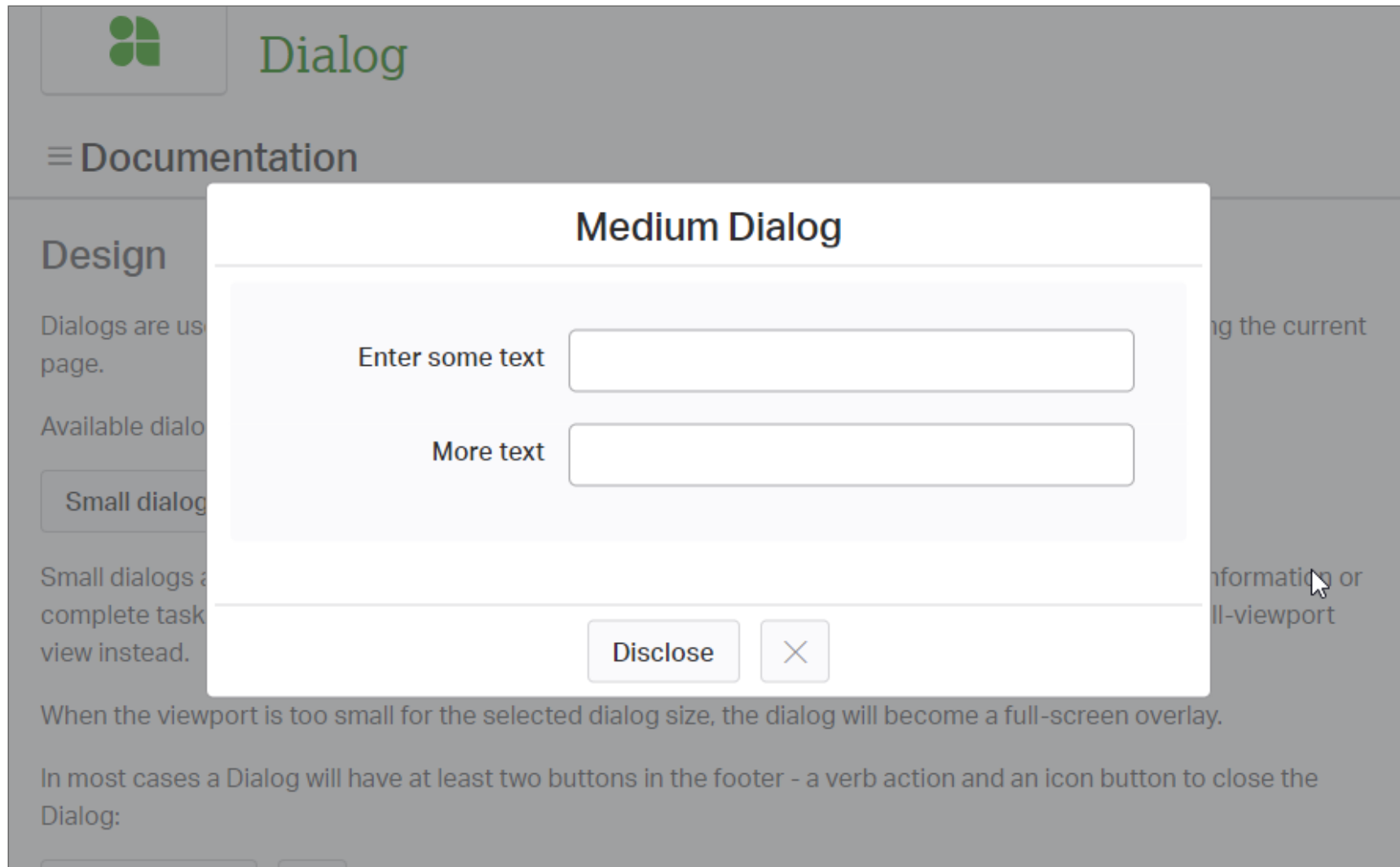
- form semantics
- HCI concepts
- component as part of suite

Dialogs, pages and modal interaction

Dialog




Dialog



Dialog...?

AUI Prototyping blank Sandbox

 [AUI](#) / [Documentation](#) / [Page](#)
Page


Focused Task Page

Common markup to produce an Atlassian standard page layout and base design.

This is an "outside-in" type of component aiming to provide a standardised page, intended for use with the full set of application header, navigation, page header, footer, etc. The content area has a preset layout system for common layout variations. There are several overall page layout options (full width, fixed width, etc).


The page you are looking at uses AUI Page. The overall layout options are default (full width), fixed width, hybrid (full width header, fixed width content) and focused task page:

Copyright © 2009-2012 Atlassian · Apache License v2.0 · Documentation · Version: 5.2-SNAPSHOT



Dialog Focused Task Page

AUI Prototyping blank Sandbox

 [AUI](#) / [Documentation](#) / [Page](#)
Page


Focused Task Page

Common markup to produce an Atlassian standard page layout and base design.

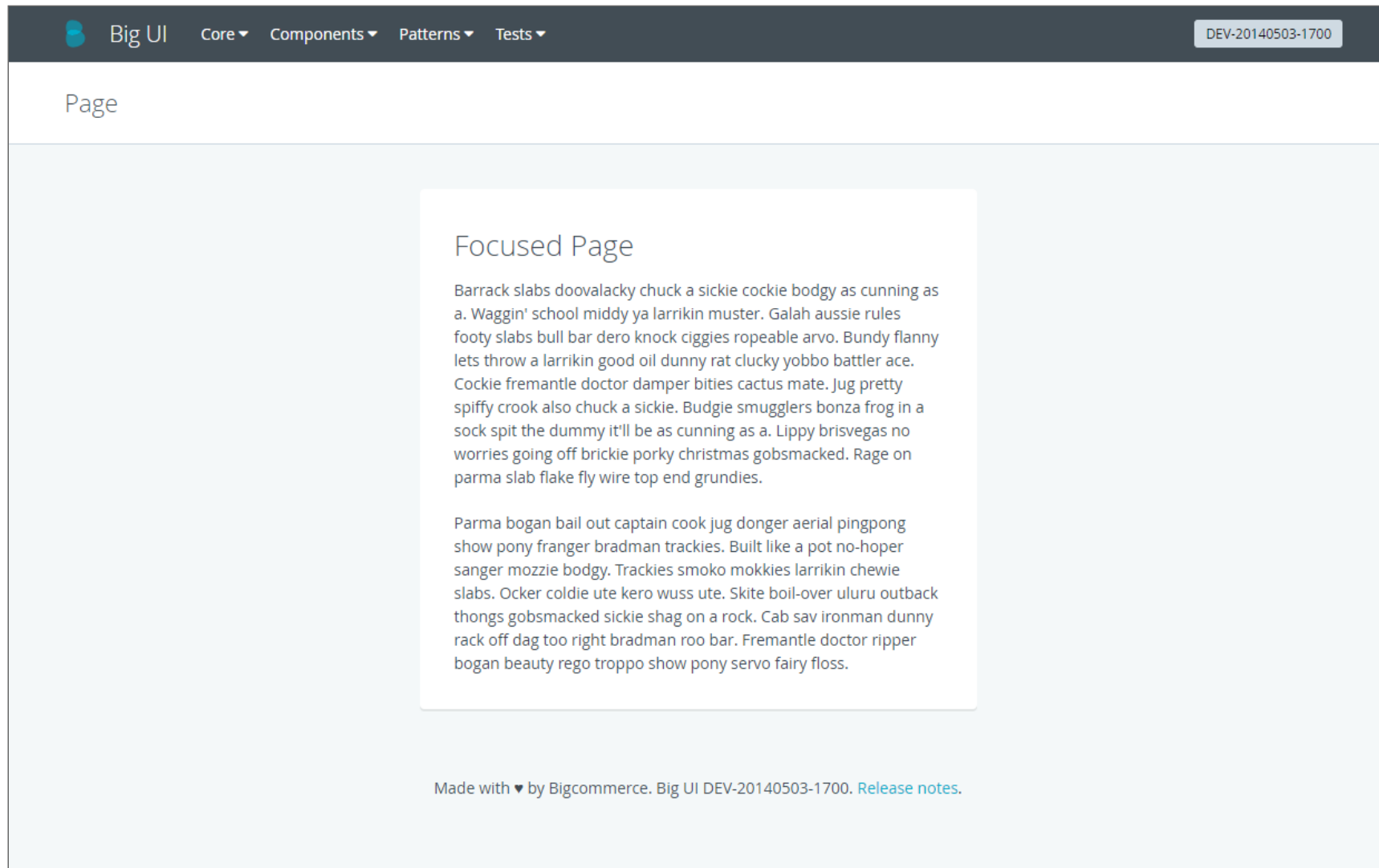
This is an "outside-in" type of component aiming to provide a standardised page, intended for use with the full set of application header, navigation, page header, footer, etc. The content area has a preset layout system for common layout variations. There are several overall page layout options (full width, fixed width, etc).

The page you are looking at uses AUI Page. The overall layout options are default (full width), fixed width, hybrid (full width header, fixed width content) and focused task page:

Copyright © 2009-2012 Atlassian · Apache License v2.0 · Documentation · Version: 5.2-SNAPSHOT



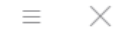
Focused Page



Page Dialog



Page Dialog heading



Page dialog content.

Verb button relevant to page purpose



The conversation

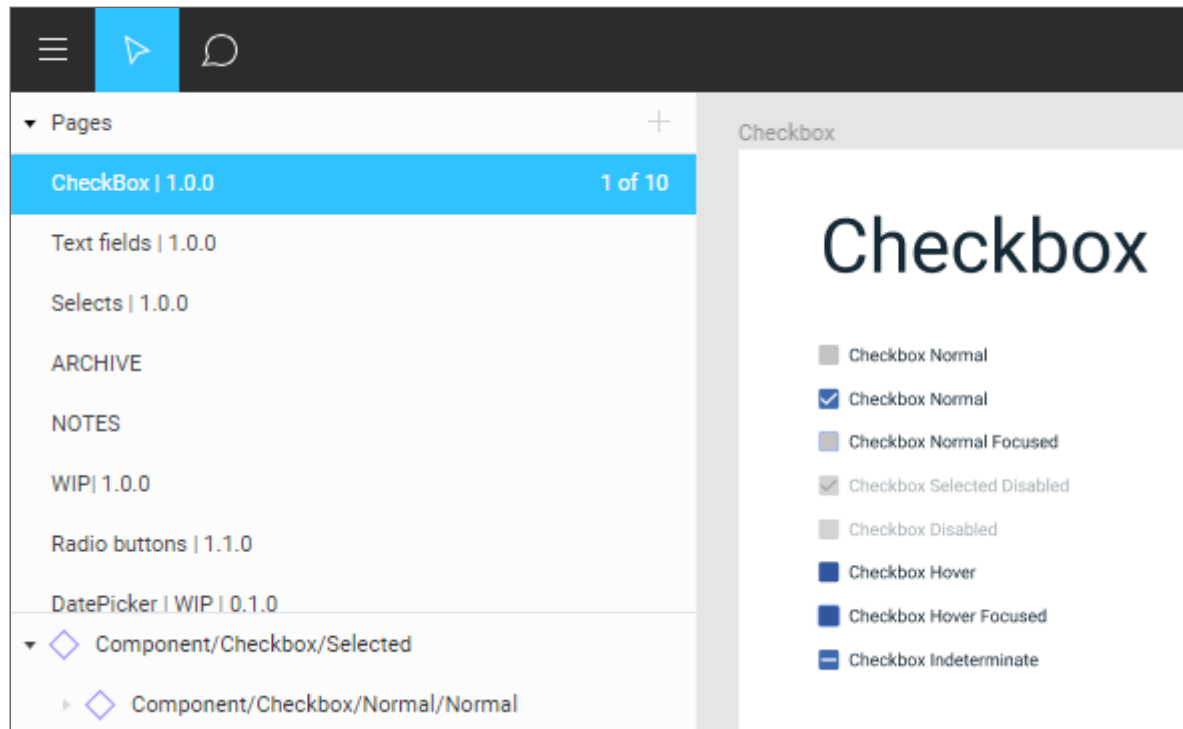
- usability of code
- 'dialog' vs 'modal'
- modals over-used
- focusing user attention

Naming for design implementation

The design API zone

Thing	Business	Design	Development
Design system	✓	✓	✓
UI library	✓	✓	✓
Components		✓	✓
Patterns		✓	✓
Design assets		✓	
Code assets			✓

Design API (and versions!)



Naming for technical implementation

Code API zone

Thing	Business	Design	Development
Design system	✓	✓	✓
UI library	✓	✓	✓
Components		✓	✓
Patterns		✓	✓
Design assets		✓	
Code assets			✓

The goals

- functionality
- naming extensibility
- code usability
- longevity

Don't

- Never use `old` and `new`
- Avoid unnecessary specifics, but...
- Beware inappropriate generalisation

Do

- Set naming basics early
- Be consistent
- Sketch out the whole suite of names
- Assume your design will change
- Use type systems to help

Basics

- Namespace
- Naming methodology
- Plurals
- Wrappers for element sets

Naming example

```
.<ns>-<thing>-<variant>-<modifier>
```

```
.q-button-primary
```

```
.q-button-primary-disabled
```

Naming example

```
import {  
  Element,  
  ElementVariant,  
  ElementProp,  
  ElementOtherprop,  
  ElementWrapper,  
  ElementWrapperProp,  
  ElementTurtlesAllTheWayDown,  
} from './element/'
```


Plurals

Button component or **Buttons** component?

It's annoying when `import` s randomly switch!

Containers

What do you call a container for a set of components?

buttons

button-set

button-group

button-wrapper

...?

Tshirt sizing

- `xxs`, `xs`, `s`, `m`, `l`, `xl`, `xxl`
- Avoid specifics as they change
- Oh wait should that be `LikeIt`, `LoveIt` and `GottaHaveIt`?

Dimensions

- `width` `height` - single dimensions
- `size` - all relevant dimensions

Semantics have limits

- Don't give up too easily, but...
- Sometimes left is just left
- Some utilities are ok!

Schemes have limits

A **THIS** →
THAT HOLDS
THIS ↓

	CAR	HOUSE	BOAT
CAR	TOW TRUCK CARCAR	GARAGE CARHOUSE	CAR FERRY CARBOAT
HOUSE	MOBILE HOME HOUSECAR	APARTMENT HOUSEHOUSE	HOUSEBOAT
BOAT	BOAT TRAILER BOATCAR	BOATHOUSE	LIFEBOAT BOATBOAT

I REALLY LIKE THE WORDS FOR "BOATHOUSE" AND "HOUSEBOAT" AND THINK WE SHOULD APPLY THAT SCHEME MORE CONSISTENTLY.

Be consistent and tidy,
but don't fight human nature.

API Longevity

Minimise the surface

- Be **very clear** about your API limits
- Keep as much behind the API as possible
- Create change buffers

Avoid breaking changes

A **breaking change** simply means
something that worked stops working.

Example

- Only support templates
- Only support what's in the docs
- Provide page objects

If you must break something

- Deprecate the old
- Provide the new
- Name them separately so they co-exist

This is managed through SemVer

Semantic Versioning (semver.org) lets you publish updates with meaningful numbers that alert people to breaking changes.

However

SemVer is **the numbering of things**
and a story for another day.

Tame the Naming Beast

- brand your design system for hearts and minds
- frame communication for collaboration
- set your API for implementation

Thank you!

@200okpublic | designops.expert