

protoscript  
simplified prototype scripting

# Prototyping the Rich Web Experience

Bill Scott  
Yahoo! Ajax Evangelist  
b.scott@yahoo.com

## stuff i've done



### Rico JavaScript for Rich Internet Applications

An open-source JavaScript library for creating rich internet applications. Rico provides full Ajax support, management and a cinematic effects library.

#### BACKGROUND

Rico is a Spanish word meaning rich. The goal of Rico is to provide a rich experience for web sites using Ajax technology. Rico is just one small part of a larger effort at Sabre Airline Solutions to create a suite of rich internet components, behaviors and effects for the web application space.

#### OPEN SOURCE

Rico is provided free and open-source (Apache 2.0 license) for either your personal or commercial use. Sabre Airline Solutions retains the copyright on the Rico code produced at Sabre.

#### BROWSER SUPPORT

Rico has been tested on IE 5.5, IE 6, Firefox 1.0x Win, Camino/Mac, Firefox 1.0x Mac. Currently there is no Safari or Mac IE 5.2 support. Support will be provided in a near future release for Safari.

#### AJAX SUPPORT

Ajax is the term that describes techniques for creating internet applications. The key ingredients to the Ajax Rico provides a very simple interface to request handlers as well as Ajax objects as Ajax response objects may be updated as they are returned.

#### DRAG AND DROP

Desktop applications have long interfaces to simplify user use the simplest interfaces for end support drag and drop. And in JavaScript object as a drags handles the rest.

#### CINEMATIC EFFECTS

When actions are no longer required to click the user only transitions can communicate Rico provides several cinematic

#### BEHAVIORS

Take some raw HTML and sprinkle in some behaviors and what do you get? Well in Rico you can get an Accordion component like those found in Macromedia Flex and Lasso. Just nest some DIVs and with one line of JavaScript turn your div panels into an accordion. And the latest behavior is the LiveGrid. LiveGrid allows you to connect an HTML table up to a stream of Ajax responses. Ajax requests are automatically called during table scrolling. The result is now HTML tables can hold an unlimited amount of data scrolled into view on the fly as needed! More behaviors are planned!

## prototyping

- Have been involved with prototyping for a number of years
- Last summer wrote a quick dragdrop proto library that was based on CSS class names and associating behavior to these classes ('dragme', 'drophere')
  - Goal was to make it easy for designers to start prototyping
  - However it was not possible to provide enough logic without resorting to writing code
  - Been planning to get back to the idea for a year.
- This got me thinking about simplifying the experience of prototyping with dhtml/ajax

## what i really want

Bill Scott  
My Portfolio  
Total Points 24

My Portfolio  
Sort by: Recent | Alpha | Points

Project One  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin tempor, lectus aliquet ornare suscipit, diam turpis suscipit, elit rhoncus feugiat purus metus gravida turpis. Nam nisi. Nulla et est in neque laoreet consequat. Quisque nisl pede. Curabitur pretium. Etiam varius. Suspendisse tempus nisl eu lectus. Integer et velit. Quisque quis magna. Nunc pellentesque pharetra ligula. Morbi semper.

Project Two  
Quisque nunc. In ac urna egestas arcu egestas imperdiet. Donec id erat id nulla molestie venenatis. Nam a quam. Nulla ornare. Fusce dignissim ultricies turpis. Fusce molestie odio ac eros. Nunc auctor massa in metus. Etiam mollis lorem in eros. Aliquam mi. Nam feugiat ante et felis.

Project Three  
Maecenas est mauris, volutpat sed, sollicitudin sed, dignissim ac, risus. Praesent rutrum gravida tortor. In in mi. Donec portitor massa ut velit. Cras mollis sollicitudin quam. Cras felis sapien, dictum et, auctor eget, volutpat at, magna. Nunc id augue non quam varius molestie. Vestibulum quis ligula. Nunc semper. Vivamus ut turpis vel lorem vulputate sagittis. Proin tincidunt elit vitae augue tempus sagittis. Ut convallis nunc ut metus. Vestibulum dictum. Sed posuere est quis arcu. Suspendisse vitae nibh pretium velit lacinia venenatis. Nullam mattis congue neque. Vestibulum mattis, mauris quis eleifend ornare, ligula lectus ullamcorper enim, vel tempor tellus eros sed dul.

view delete  
view delete  
view delete

Scratchpad

Popup

Animate ColorAnimate Fade Move Spotlight  
Close DragDrop Hide Open Popup  
ReplaceClass Script SetClass SetStyle  
Show ToggleClass ToggleOpenClose  
ToggleShowHide FetchHtml SetHtml Blur  
Click DbClick Focus Keypress Mousedown  
Mousemove Mouseout Mouseover  
Mouseup Timer

take a rough prototype

sprinkle in some behaviors

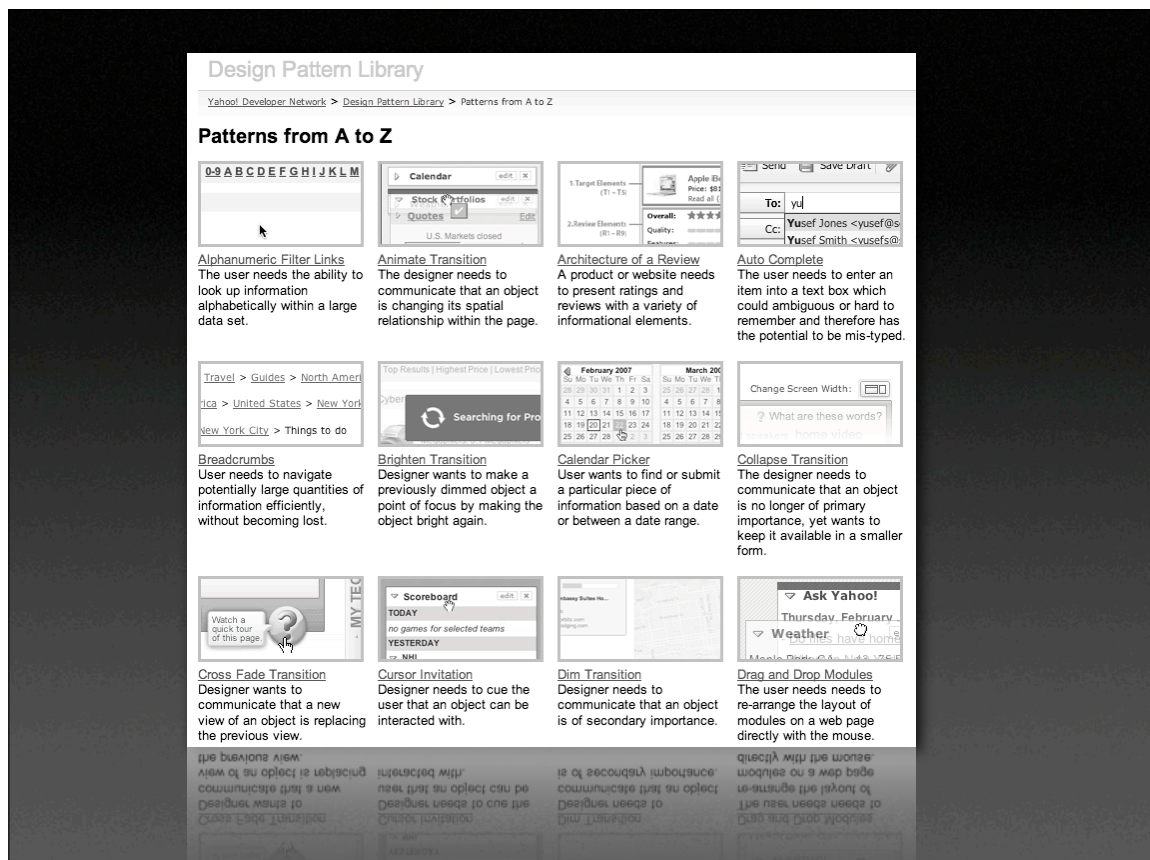
## what we will discuss

- quick overview of common design patterns (behaviors)
- quick survey of techniques to prototype a rich web experience
- dive into protoscript and discuss how it can be used to sprinkle behaviors into web pages
- look at similar techniques emerging (jQuery, css behaviors, DED | Chain, etc.

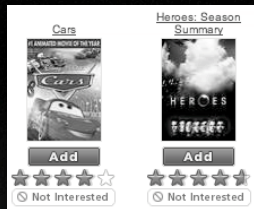
patterns  
(common interaction behaviors)

# ria patterns

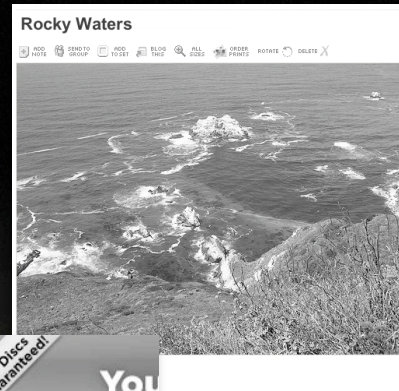
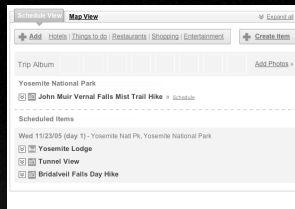
invitation, cursor invitation, hover invitation, tooltip invitation, page invitation, drop invitation, available, selected, auto complete, live suggest, refining search, dynamic filtering, live search, on-demand scrolling, deferred content loading, fresh content, on-demand content, periodic refresh, resizable modules, scrolling modules, hover detail, narrowing actions, narrowing choices, auto form fill, in-place drill down, dependent choices, look before you leap, drag and drop, drag and drop modules, drag and drop objects, persistent portals, in page edit, inline text edit, inline custom edit, direct state edit, rating an object, inline custom edit, inline tag edit, popup custom edit, grid cell edit, slide out custom edit, module configure, module flip configure, module inline configure, module slide out configure, module faceplate, expandable content, in-context tools, in-context links, in-context hover menu, silent submit, remembered collection, auto save, remembered preferences, hover spy, inline assistant, inline validation, validate then suggest, indicator: busy indicator, progress indicator, inline status, in-context busy, cursor busy, opacity fade, high contrast, balloon error tip, dynamic goal, opacity focus, detail zoom, lightbox, transition, brighten, cross-fade, dim, expand, fade-in, fade-out, flip, move, self-heal, collapse, slide, animate.



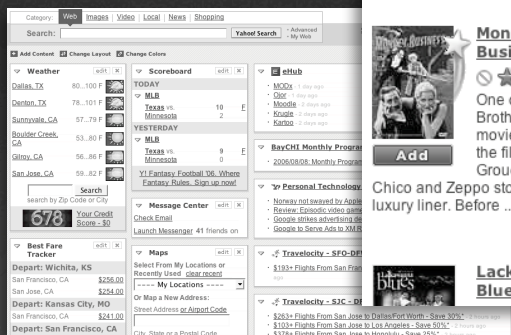
# ria pattern examples



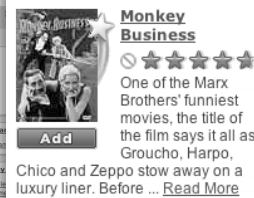
pattern. in-page action.



pattern. Bill, the following movies were chosen based on your interest in: [Animal Crackers](#) [Something the Lord Made](#) [X2: X-Men United](#)



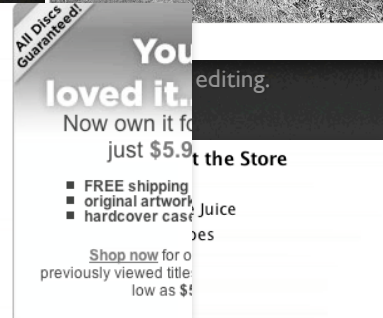
pattern. drag & drop.



Chico and Zeppo stow away on a luxury liner. Before ... [Read More](#)

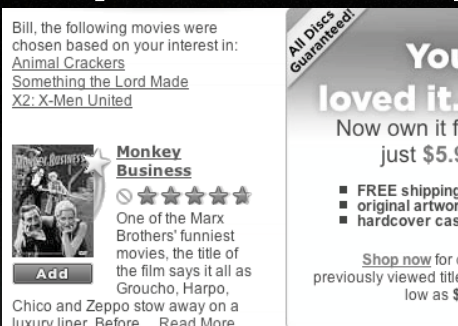


Rate Your

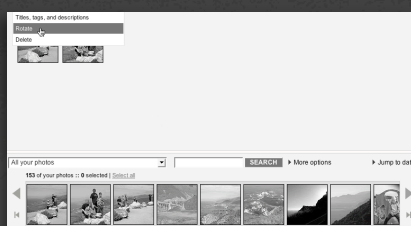


pattern. hover details.

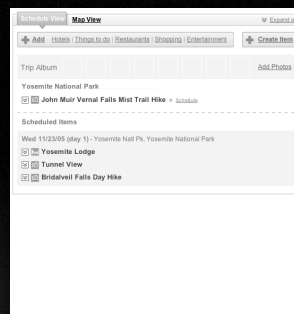
# ria pattern examples



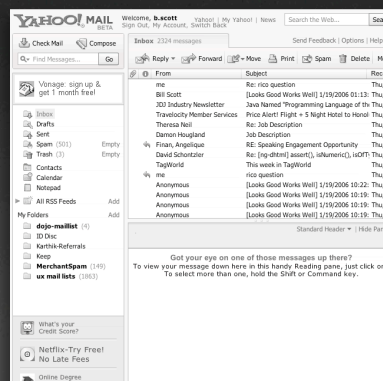
pattern. hover details.



pattern. lightweight popup + lightbox.

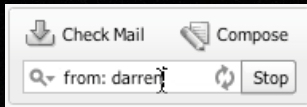


pattern. in-context expand.

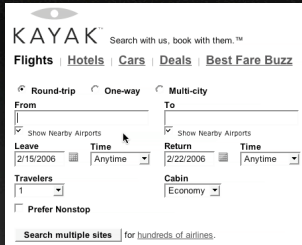


pattern. on-demand scrolling.

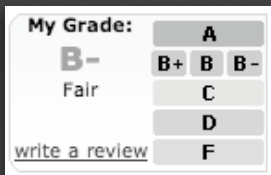
# ria pattern examples



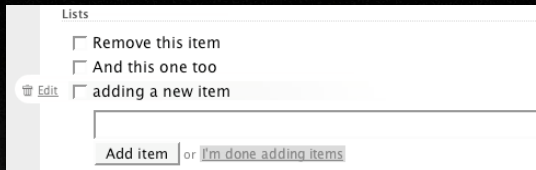
pattern. busy indicator.



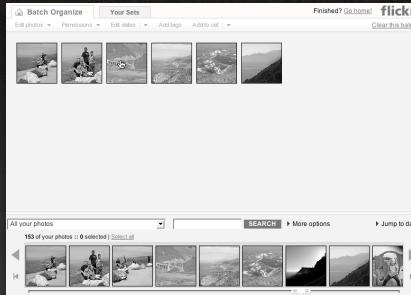
pattern. auto complete.



pattern. hover invitation.



pattern. fade transition + self-healing transition.



pattern. drop invitation.



pattern. live previews.

Date: Nov-05				
Drag Over Invalid Target	Drag Over Parent Container	Drop Accepted	Drop Rejected	Drop On Parent Container
 CSS Move cursor	 CSS Move cursor	 Normal Cursor	 Normal Cursor	 Normal Cursor
 Reduced Opacity & Invalid Badge	 Reduced Opacity	 Modules animates into the area just below insertion bar  Module comes to rest in new area  Modules slide up in a self-healing transition to close hole	 Modules animates back to the home area  Module comes back to rest at full opacity	 Modules animates back to the home area  Module comes back to rest at full opacity
 it will drop	 No insertion bar, just a gap	 No insertion bar, just a gap & original hole frame of animation	 Insertion bar is removed as first frame of animation	 Insertion bar is removed as first frame of animation

## interesting moments for drag drop

	Page Generation	Mouse Hover	Drag Initiated	Drag over Valid	Drag over Invalid	Drag over Original	Drop Accepted	Drop Rejected	Drop on Original
Page Content	Hint	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Cursor	Normal	Move Cursor	Move Cursor	Move Cursor	Move Cursor	Move Cursor	Normal	Normal	Normal
Drag Object	Normal	Normal	Reduced Opacity & Tracking	Reduced Opacity & Tracking	Reduced Opacity & Tracking + Invalid Badge	Reduced Opacity & Tracking	2. Modules animates into the area just below insertion bar 3. Module comes to rest in new area 4. Modules slide up in a self-healing transition to close hole	Normal Opacity + Zoom Back to Original	Normal Opacity + Zoom Back to Original
Orig Location	Normal	Normal	Hole Opens	Hole Remains	Hole Remains	Hole Remains	Hole Remains	Hole refilled with drag object	Hole refilled with drag object
Drop Target	Normal	Normal	Normal	Insertion Bar	N/A	N/A	1. Insertion Bar Removed	N/A	N/A
Drop Target	Normal	Normal	Normal	Insertion Bar	N/A	N/A	1. Insertion Bar Removed	N/A	N/A

non-dhtml  
prototyping techniques

## challenge of interesting moments



*Wireframing AJAX is a bitch. The best our agency has come up with is the Chuck Jones approach: draw the key frames. Chuck Jones had an advantage: he knew what Bugs Bunny was going to do. We have to determine all the things a user might do, and wireframe the blessed moments of each possibility.*  
- Jeffrey Zeldman

<http://www.alistapart.com/articles/web3point0/>

- Blessed Moments = Microstates = Interesting Moments
  - behavior is not static
  - Multiple states need to be captured
  - Happens within a context
  - Not suited for wireframes

## using keyframes

AJAX Status Interaction

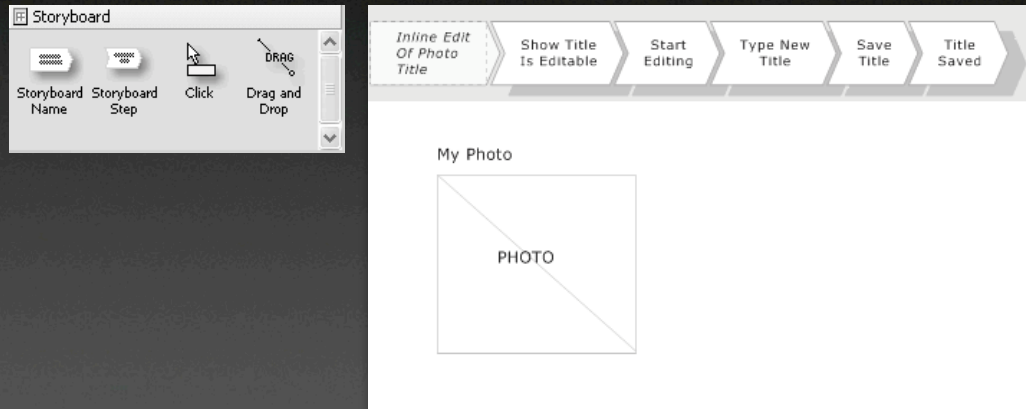
1 On Page Load

2a On Click display message (listen and subscribe buttons have same)

2b Fade (50% opacity)

## prototyping with visio

- Changes in state placed in different layers
- Storyboard steps toggle visibility of associated layers
- Each storyline serves as a use case in the wireframe



Source: [http://www.boxesandarrows.com/view/storyboarding\\_rich\\_internet\\_applications\\_with\\_visio](http://www.boxesandarrows.com/view/storyboarding_rich_internet_applications_with_visio)

## prototyping with keynote/powerpoint

- Slides make major steps
- Builds used to bring in transitions, animations

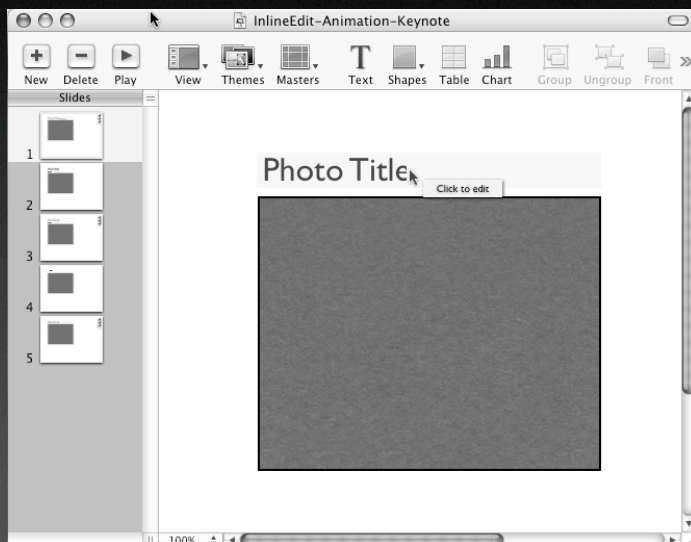
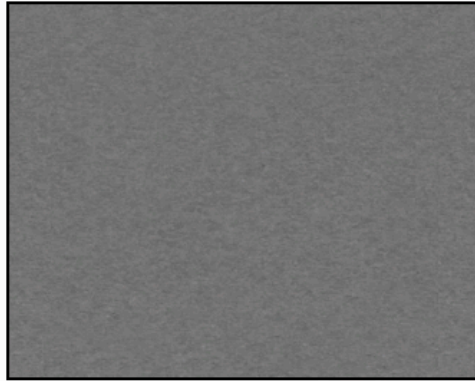
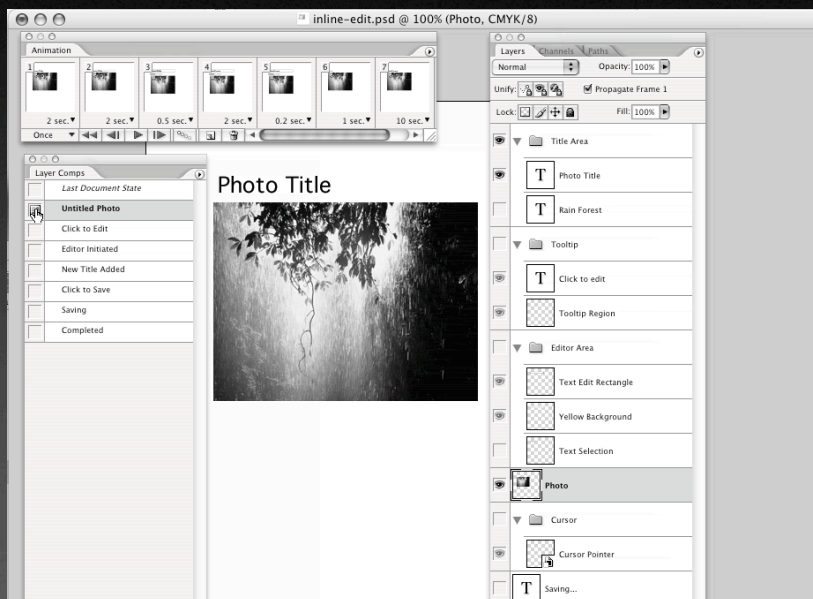


Photo Title



## prototyping with photoshop

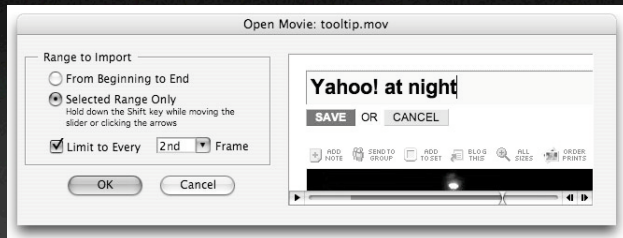
- Easy to create from a series of images
- Images can be created by hand or from other apps



## animations from screencasts

Start with screencast (SnapZPro/SnagIt)

Turn movie into animated GIF (Adobe ImageReady)



## other approaches

Flash

Flex

iRise ([www.irise.com](http://www.irise.com)) \$\$\$

Axure ([www.axure.com](http://www.axure.com))

Interactive PDFs

Fireworks + PDF

Powerpoint

DHTML

# dhtml prototyping

## prototyping with DHTML

- Benefits
  - Can start simple and layer in complexity
  - Works well across different teams (design to code)
  - Technology the same or similar to final implementation
  - Most capable to approximate final result
  - Living example
- Disadvantages
  - Requires technical savvy
  - Bugs can suck down time
  - Too much time spent 'messing with code'
  - Hard to extract into documentation

## dhtml prototyping skills/techniques

- Keep structure, style and logic separate
  - Avoid adding onclick handlers in the HTML (unobtrusive)
  - Avoid any direct styles or html style tags (font, etc.)
- Use style class switching for visual changes
- Use CSS to simulate most mouse hover behavior
- Use innerHTML for content changes
- Understand when to use a class vs. id
- Use CSS query libraries for referencing interface parts
  - CSS selector syntax allows non-intrusive behavior injection

## dhtml prototyping skills/techniques

- Use anchors as basic navigation/action hooks
- Use background images in CSS with style switching
- Use a grid layout system
  - YUI Grid
- Use Firebug
- Understand CSS Selectors
  - Ext, jQuery, Dojo, Mootools, Prototype, cssQuery()
- Use JSON syntax (options)

## underlying desire for protokit

- address the conceptual phase of interface development  
-- think “interaction theme”  
*(free from solving production issues)*
- express an interaction as a configuration  
*(no code; declarative approach)*
- dynamically insert interaction themes (behaviors) to any web page  
*(allows experimentation)*
- create a gui tool for configuring & exploring interactions

## protoscript approach

- JSON used to express interaction theme to express relationship between dom elements, behaviors, attributes, and callbacks
- jQuery CSS Selector syntax for dom elements
- Chain dependent behavior as well as parallel behavior with JSON syntax
- Simple to add behaviors (class with 2 methods)
- Focus is on prototyping

# Demo

<http://protoscript.com/demos.php>

## simple example

Protoscript is a simplified scripting language for creating Ajax style prototypes for the Web. With Protoscript it's easy to bring interface elements to life. Simply connect them to behaviors and events to create complex interactions.

Here's an example that fades & closes the image when the user clicks on it. (It's live, so go ahead and try it!)

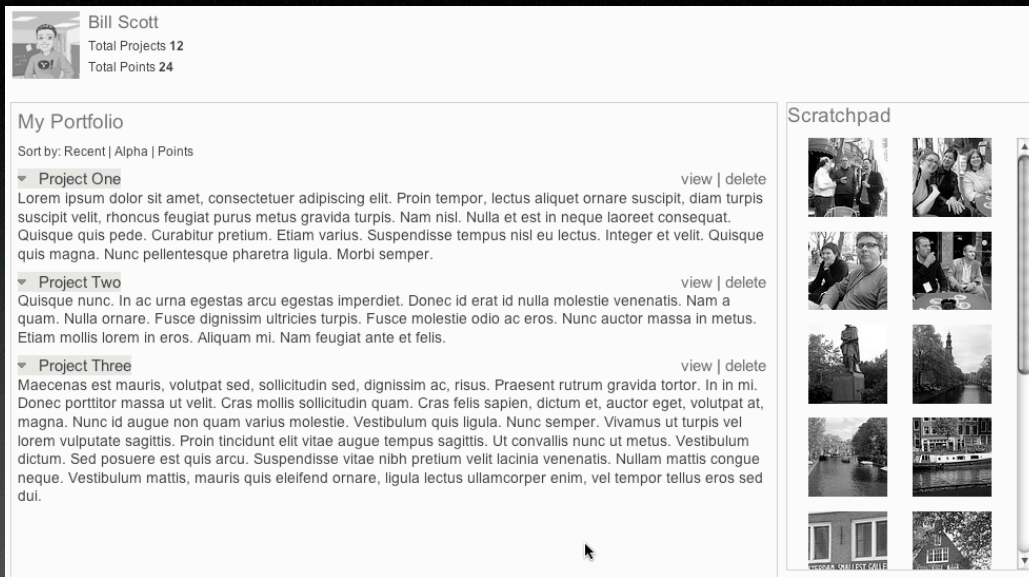


```
$proto('img#avatar', {  
  Click: {  
    onClick: {  
      Fade: {  
        opacity: {to: 0},  
        onComplete: {Close : {} }  
      }  
    }  
  }  
});
```

The protoscript above says: *for an image with id **avatar**, fade it out when the user clicks on it and close it when the fade completes.*

Simple Click, Fade, Close  
(self-healing delete)

# sprinkling in behaviors



All behaviors added with protoscript

## anatomy of an interaction

### DOM elements

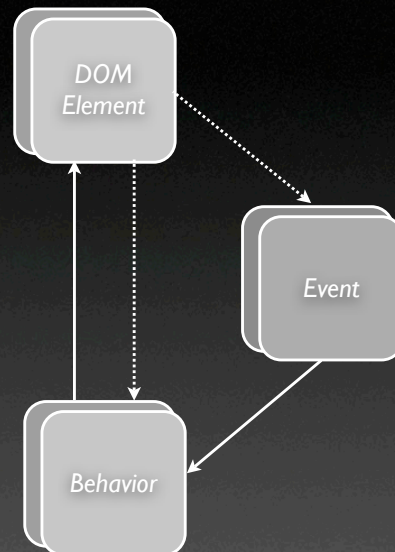
Target an element or elements for attaching events & behaviors.

### Events

Events are the glue between the user and behaviors. They start behaviors.

### Behaviors

Blocks of logic, patterns, actions that operate on elements.



# anatomy of self-healing delete

## DOM elements

[X]

<LI>[X] oldmovie.mov</LI>

## Events

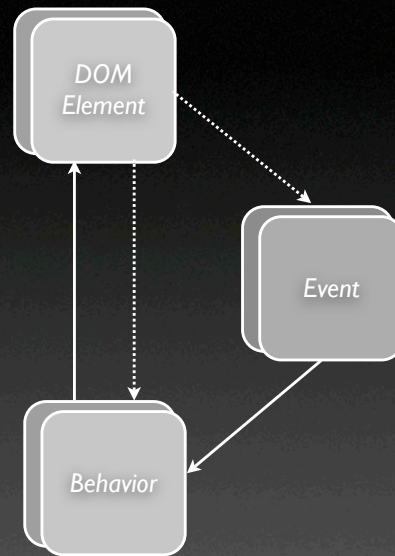
Click

onClick (callback)

## Behaviors

Fade

Close



## protoscript code

```
<div id="multiple" class="example-live">
  <p>Files</p>
  <ul>
    <li><span><strong>[X]</strong></span><span>mystuff.tmp</span></li>
    <li><span><strong>[X]</strong></span><span>bigfile.tmp</span></li>
    <li><span><strong>[X]</strong></span><span>oldmovie.mov</span></li>
    <li><span><strong>[X]</strong></span><span>scrap.txt</span></li>
  </ul>
</div>
```

Files

- [X] mystuff.tmp
- [X] bigfile.tmp
- [X] oldmovie.mov
- [X] scrap.txt

```
$proto('span:contains([X])', {
  SetStyle: {cursor:'pointer'},
  Click: {
    onClick: {
      Fade: {
        target: '#multiple li',
        opacity: {to: 0},
        onComplete: {Close : {} }
      }
    }
  },
});
```

Element

Behavior

Event

## selecting dom elements (targeting)

Element

```
<div id="multiple" class="example-live">
  <p>Files</p>
  <ul>
    <li><span><strong>[X]</strong></span><span>mystuff.tmp</span></li>
    <li><span><strong>[X]</strong></span><span>bigfile.tmp</span></li>
    <li><span><strong>[X]</strong></span><span>oldmovie.mov</span></li>
    <li><span><strong>[X]</strong></span><span>scrap.txt</span></li>
  </ul>
</div>
```

Files

[X] mystuff.tmp  
[X] bigfile.tmp  
[X] oldmovie.mov  
[X] scrap.txt

```
$proto('span:contains([X])', {
  SetStyle: {cursor:'pointer'},
  Click: {
    onClick: {
      Fade: {
        target: '#multiple li',
        opacity: {to: 0},
        onComplete: {Close : {} }
      }
    }
  },
});
```

Behavior

Event

## chaining of target scope

Element

```
<div id="multiple" class="example-live">
  <p>Files</p>
  <ul>
    <li><span><strong>[X]</strong></span><span>mystuff.tmp</span></li>
    <li><span><strong>[X]</strong></span><span>bigfile.tmp</span></li>
    <li><span><strong>[X]</strong></span><span>oldmovie.mov</span></li>
    <li><span><strong>[X]</strong></span><span>scrap.txt</span></li>
  </ul>
</div>
```

Files

[X] mystuff.tmp  
[X] bigfile.tmp  
[X] oldmovie.mov  
[X] scrap.txt

```
$proto('span:contains([X])', {
  SetStyle: {cursor:'pointer'},
  Click: {
    onClick: {
      Fade: {
        target: '#multiple li',
        opacity: {to: 0},
        onComplete: {Close : {} }
      }
    }
  },
});
```

Behavior

Event

## selector syntax

Element

Consult jQuery docs:

[http://docs.jquery.com/DOM/Traversing/Selectors#CSS\\_Selectors](http://docs.jquery.com/DOM/Traversing/Selectors#CSS_Selectors)

If you are setting up your own prototypes, stick with class and identifier

`.my-class-name`

`#my-id-name`

Other useful examples

`a:contains(Click Here)`

`.my-class-name #my-id-under-there`

`#my-specific-container .my-items`

Consult CSS-1, CSS-2, CSS-3 and XPath

Behavior

Event

## behaviors

Behavior

Animate. ColorAnimate. Fade.

Move. Spotlight. Close.

DragDrop. Hide. Open.

Popup. ReplaceClass. Script.

SetClass. SetStyle. Show.

ToggleClass. ToggleOpenClose.

ToggleShowHide. FetchHtml.

SetHtml.

Event

Element

## behavior demos

Behavior

Site has a demo page for each behavior

<http://protoscript.com/demos.php>

Site has a wiki with documentation page for each behavior

<http://docs.protoscript.com/>



Event

Element

## behavior attributes

Behavior

Behaviors may supply attributes for configuring the behavior

Where possible if an underlying YUI component or utility has a config object, this is exposed

**EXAMPLE FROM Popup BEHAVIOR (bold denotes from Panel cfg)**

```
close: true,
draggable: true,
underlay: 'shadow',
model: false,
visible: true,
effect: null,
monitorresize: true,
x: null,
y: null,
xy: null,
context: null,
fixedcenter: true,
width: '300px',
height: null,
zIndex: null,
constraintviewport: false,
iframe: false,
id: 'popup_panel',
hd: 'Panel Title',
bd: 'This is the body',
ft: null
```

Event

Element

## plug-in behavior

Behavior

Protoscript core is independent of any single toolkit

selector is a plugin. *default set uses jQuery*

behavior set is plugged in. *default set uses YUI.*

```
ProtoScript.Core.registerBehaviorSet($, YAHOO.protoscript);  
(yui-proto.js)
```

You are free to create your own behavior sets or plug in other selector technologies

Example: Mootools selector & behaviors

```
ProtoScript.Core.registerBehaviorSet($ES, MooBehaviors);  
(moo-proto.js)
```

You can also add behaviors to the default set

Event

Element

## peek under the hood of a behavior

Behavior

```
YAHOO.protoscript.Show = function (behaviorName, behaviorCfg) {  
    this.init(behaviorName, behaviorCfg);  
};  
YAHOO.protoscript.Show.prototype = {  
    defaultCfg : {  
    },  
  
    init: function(behaviorName, behaviorCfg) {  
        this.name = behaviorName;  
        this.cfg = ProtoScript.Core.applyConfig(this.defaultCfg, behaviorCfg);  
    },  
  
    action: function(currScope) {  
        YAHOO.util.Dom.setStyle(currScope.elms, 'visibility', 'visible');  
        if(this.cfg.onShow) {  
            ProtoScript.Core.callBehaviors(this.cfg.onShow, currScope);  
        }  
    }  
};
```

'Show' behavior

Event

Element

## adding a behavior (to default YUI set)

Behavior

1. Add behavior to `YAHOO.prototype` namespace
2. In `init` store name & apply config parameters
3. In `action` put your behavior logic

The action performs the behavior

`action(currScope)`

`currScope.elems`: list of elems resolved from the current target scope

`currScope.idx`: if multiple elems, the index into the active element for this action

`currScope.pseudoElems`: some behaviors set dynamic elements (like `$drag`, `$drop` in `DragDrop` behavior).

4. For any interesting moments, call

`ProtoScript.Core.callBehaviors` to process the callback

5. Add attributes as needed (in your `defaultCfg`)

Event

6. Start using it in your scripts

Element

## events

Event

Blur. Click. Dblclick. Focus.  
Keypress. Mousedown.  
Mousemove. Mouseout.  
Mouseover. Mouseup. Timer.

Element

Behavior

## events are actually behaviors

Event

```
YAHOO.protoscript.Click = function(behaviorName, behaviorCfg) {
    this.init(behaviorName, behaviorCfg);
};
YAHOO.protoscript.Click.prototype = {
    defaultCfg : {},
    init: function(behaviorName, behaviorCfg) {
        this.name = behaviorName;
        this.cfg = ProtoScript.Core.applyConfig(this.defaultCfg, behaviorCfg);
    },
    action: function(currScope) { YAHOO.protoscript.Trigger(this.cfg,
currScope, 'click');}
};
```

'Click' behavior

Element

Behavior

## callbacks

Event

onComplete, onTween, onStart, onClose,  
onStartDrag, onDrag, onDragEnter, onDragOver,  
onDragDrop, onDragOut, onEndDrag, onMouseUp,  
onInvalidDrop, onBlur, onClick, onDbClick, onFocus,  
onKeyPress, onMousedown, onMousemove,  
onMouseout, onMouseover, onMouseup, onTimer,  
onHide, onOpen, onReplaceClass, onSetClass,  
onSetStyle, onShow, onToggleClass,  
onToggleOpenClose, onToggleShowHide,  
onSuccess, onFailure, onSetHtml.

Element

Behavior

# callbacks are interesting moments

## Interesting Moments

Simply making an object draggable rarely suffices to achieve the desired interaction behavior for a drag-and-drop implementation. In most cases, Drag and Drop requires that you write code to respond to the interesting moments in the interaction: when the drag event starts, when the dragged object enters another object, and so on. The Drag and Drop Utility provides methods that fire during each of the interesting moments of the interaction. You customize your implementation by supplying the code for these methods (all of which are members of the `YAHOO.util.DD`, `YAHOO.util.DDProxy`, and `YAHOO.util.DDTarget`).

Moment	Description
onMouseDown	Provides access to the mousedown event. The mousedown does not always result in a drag operation.
startDrag	Occurs after a mouse down and the drag threshold has been met. The drag threshold default is either 3 pixels of mouse movement or 1 full second of holding the mousedown.
onDrag	Occurs every mousemove event while dragging.
onDragEnter	Occurs when the dragged object first interacts with another targettable drag and drop object.
onDragOver	Fires every mousemove event while over a drag and drop object.
onDragOut	Fires when a dragged object is no longer over an object that had the onDragEnter fire.
onDragDrop	Fires when the dragged objects is dropped on another.
onInvalidDrop	Fires when the dragged objects is dropped in a location that contains no drop targets.
endDrag	Fires on the mouseup event after a drag has been initiated (startDrag fired).
onMouseUp	Fires on the mouseup event whether or not a drag was initiated.

# callbacks are invoked in behaviors

```
YAHOO.protoscript.Open = function (behaviorName, behaviorCfg) {
    this.init(behaviorName, behaviorCfg);
};
YAHOO.protoscript.Open.prototype = {
    defaultCfg : {
    },

    init: function(behaviorName, behaviorCfg) {
        this.name = behaviorName;
        this.cfg = ProtoScript.Core.applyConfig(
            this.defaultCfg, behaviorCfg);
    },

    action: function(currScope) {
        YAHOO.util.Dom.setStyle(currScope.elms, 'display', 'block');
        if(this.cfg.onOpen) {
            ProtoScript.Core.callBehaviors(this.cfg.onOpen, currScope);
        }
    }
};
```

'Open' behavior invoking onOpen

Event

Element

Behavior

# Bookmarklet\*

\*humble beginnings

**Burritophile** It's all in your hands.

Sign Up NowLogin

Search

»

Blog

User Reviews

Editorial Reviews

Scoring

Directory (recent)

San Francisco (M)

Mountain View (M)

San Jose (M)

Los Angeles (M)

San Diego (M)

Chicago (M)

Boston (M)

more cities...

Suggest

Write A Review

Articles

About

Press

Store

Links

RSS

## Directory

Browse our directory of burrito places (recently added)

**USA > California > Sunnyvale** (show map)

Sort by: A to Z | [High Scores](#)

- [Burrito Factory](#) (no reviews) 1671 Hollenbeck Ave
- [Chavez Supermarket Taqueria](#) (7.2) 666 N Fair Oaks A
- [El Camino Mexican Restaurant](#) (7.0) 510 E El Camino
- [Garcia's Taqueria](#) (5.7) 738 S Fair Oaks Ave
- [Hot and Mild Taqueria](#) (6.5) 848 East Evelyn Ave
- [Las Islitas](#) (5.3) 848 E Evelyn Ave
- [Little Michoacan](#) (6.5) 305 N Fair Oaks
- [Senor Jalapeno](#) (6.0) 415 N Mary Ave # 117
- [Tacos Jalapa](#) (6.5) 154 W El Camino Real
- [Taqueria Latina](#) (8.0) 195 E Maude Ave
- [Tres Potrillos Taqueria](#) (no reviews) 670 S Fair Oaks Ave
- [Tres Potrillos Taqueria](#) (no reviews) 670 N Fair Oaks Ave
- [Una Mas \(Lawrence Expy\)](#) (5.5) 548 Lawrence Expy
- [Una Mas \(Sunnyvale Saratoga Rd\)](#) (no reviews) 725 Sunny

Help us improve the directory.  
Send your suggestions to [editors\[at\]burritophile.com](mailto:editors[at]burritophile.com).

Latest User Reviews

[La Borinquena Mexicatessen](#)  
Oakland, CA


Protoscriber

For elements matching:  
 Find

Do the following:

```
SetHtml: {  
  html: '<span class="small">&nbsp;&nbsp;&nbsp;[X]&nbsp;&nbsp;&nbsp;</span>'  
},  
SetStyle: {  
  cursor: 'move'  
},  
Click: {  
  onClick: {  
    target: 'ul.bulleted li',  
    Fade: {  
      duration: 1,  
      onComplete: {  
        close: { }  
      }  
    }  
  }  
}
```

Add Behaviors


[home](#)
[demos](#)
[docs](#)
[download](#)
[forum](#)

Protoscript is a simplified scripting language for creating Ajax-style prototypes for the Web. With Protoscript it's easy to...

**Protoscripter**

For elements matching:

Do the following:  

```
Spotlight: {}
```

Protoscript is for prototyping. It's simple syntax makes it easy to sprinkle behaviors onto any web page.

**Easy to query for interface elements**  
 Uses jQuery. Plug-in architecture allows different selector plugins.

**Simple to express complex dependent behaviors**  
 Just nest behaviors within behaviors or callbacks.

**Add new behaviors & events with plug-in architecture**  
 Uses YUI library. Can be extended to use other Ajax frameworks.

You can play with the current version of protoscript with the ProtoScripter Bookmarklet.

For Firefox, drag this link [ProtoScripter](#) to your bookmarks toolbar. On IE right-click it and choose Add Favorite...

## adding protoscript to your page

### Download

- The protoscript library: [proto.js](#)
- The default behavior set (jQuery + YUI): [yui\\_proto.js](#)
- jQuery library: [jquery-1.1.3.1.js](#)

### Default YUI + jQuery Behavior Set

Include the following in your page:

```
<!-- YUI Library -->
<script type="text/javascript" src="http://yui.yahooapis.com/2.3.0/build/utilities/utilities.js"></script>
<script type="text/javascript" src="http://yui.yahooapis.com/2.3.0/build/container/container-min.js"></script>

<!-- jQuery Library -->
<script type="text/javascript" src="scripts/jquery-1.1.3.1.js"></script>

<!-- Default Behavior Set - YUI -->
<script type="text/javascript" src="scripts/yui_proto.js"></script>

<!-- YUI CSS -->
<link rel="stylesheet" type="text/css" href="http://yui.yahooapis.com/2.3.0/build/reset-fonts-grids/reset-fonts-grids.css">
<link rel="stylesheet" type="text/css" href="http://yui.yahooapis.com/2.3.0/build/base/base-min.css">
<link rel="stylesheet" type="text/css" href="http://yui.yahooapis.com/2.3.0/build/container/assets/container.css">
```

### Protoscript library

Include the following in your page:

```
<script type="text/javascript" src="scripts/proto.js"></script>
```

### Code Repository - Subversion

The code repository for protoscript is located at [googlecode](#)

### Protoscripter Bookmarklet

You can play with the current version of protoscript with the ProtoScripter Bookmarklet.

For Firefox, drag this link [ProtoScripter](#) to your bookmarks toolbar. On IE right-click it and choose Add Favorite...

<http://protoscript.com/downloads.php>

## syntax

### **\$proto(selector, interaction)**

where:

selector is the jQuery selector syntax

interaction is a configuration object (JSON) describing the interaction

```
$proto('span:contains([X])', {  
  SetStyle: {cursor:'pointer'},  
  Click: {  
    onClick: {  
      Fade: {  
        target: '#multiple li',  
        opacity: {to: 0},  
        onComplete: {Close : {} }  
      }  
    }  
  }  
});
```

## syntax

### **\$proto(config)**

single config object defines an interaction

convenient for passing interactions around or saving to config files

```
$proto(  
{  
  'span:contains([X])': {  
    SetStyle: {cursor:'pointer'},  
    Click: {  
      onClick: {  
        Fade: {  
          target: '#multiple li',  
          opacity: {to: 0},  
          onComplete: {Close : {} }  
        }  
      }  
    }  
  }  
});
```

## syntax

### **\$proto(config)**

single config object defines an interaction

convenient for passing interactions around or saving to config files

```
var cfg = {'span:contains([X])': {
  SetStyle: {cursor:'pointer'},
  Click: {
    onClick: {
      Fade: {
        target: '#multiple li',
        opacity: {to: 0},
        onComplete: {Close : {} }
      }
    }
  }
};
$proto(cfg);
```

## 0.1 beta

- Live at <http://protoscript.com>
  - Started it last month
  - reference implementation YUI + jQuery
  - 31 behaviors (including events)
  - 40+ live examples
  - full wiki site with all behaviors documented ([docs.protoscript.com](http://docs.protoscript.com))
- Open source
  - Code is at Google Code. Send me email if you want to participate will get you acces to SVN trunk
  - BSD license
- Bookmarklet
  - Just hack to start experimenting with GUI
  - Combined with firebug, works ok

## next steps

- 'You' write lots of behaviors
  - Wiki for docs
  - Google code for new behaviors & behavior sets
- New GUI tool
  - Most likely extend Firebug
  - Combine 'inspector' with behavior wizard to inject behaviors
  - Persist configurations for pages; allow re-load
- Flesh out behavior self-publishing
  - Will add spec on how to make behavior publish itself to a GUI tool

## useful approaches (inspiration)

- Responder
  - <http://www.boxpop.net/responder>
  - given a style class name, associate behavior with those objects
- jQuery
- LivePipe's Event.Behavior
  - [http://livepipe.net/projects/event\\_behavior/](http://livepipe.net/projects/event_behavior/)
- Dustin Diaz, DED Chain
  - <http://dedchain.dustindiaz.com/>

## useful approaches (inspiration)

- Transclusions
  - including html from somewhere else on your page
  - <http://ajaxian.com/archives/purple-include-transclusions-you-know-you-want-them>
- DOMDom
  - nice way to express HTML for Dom insertion
  - <http://www.zachleat.com/web/2007/07/07/domdom-easy-dom-element-creation/>
  - (merge with protokit)
- Magic DOM
  - <http://amix.dk/blog/viewEntry/19199>
- Jester: JavaScriptian REST
  - <http://giantrobots.thoughtbot.com/2007/4/2/jester-javascriptian-rest>
- JavaScript Templating Library: MJT
  - <http://mjtemplate.org/>

## useful approaches (inspiration)

- moz-behaviors (dean.edwards.name)
  - <http://dean.edwards.name/moz-behaviors/intro/>
  - uses CSS Linking mechanisms
  - <http://www.w3.org/TR/becss>
- ClassAnim
  - <http://blog.mozmonkey.com/2007/classanim-hoverhijax-keeping-presentation-out-of-your-javascript/>
- JDA Emulator: Spring for JS
  - Adds properties to the HTML tags themselves
- Ben Nolan's Behavior Library
  - <http://www.ccs.neu.edu/home/dherman/javascript/behavior/>
  - <http://www.ccs.neu.edu/home/dherman/javascript/behavior/example.html>
  - Ben Nolan, Dave Herman, Simon Wilison
- Prototype Behavior Library (built on prototype.js)

## more resources

### Articles

Web 3.0 article: [www.alistapart.com/articles/web3point0/](http://www.alistapart.com/articles/web3point0/)

[looksgoodworkswell.blogspot.com/2005/11/visio-wireframe-toolkit-for-download.html](http://looksgoodworkswell.blogspot.com/2005/11/visio-wireframe-toolkit-for-download.html)

[looksgoodworkswell.blogspot.com/2005/11/animating-interactions-with-photoshop.html](http://looksgoodworkswell.blogspot.com/2005/11/animating-interactions-with-photoshop.html)

Prototyping with PowerPoint: [blogs.msdn.com/jensenh/archive/2006/02/20/535444.aspx](http://blogs.msdn.com/jensenh/archive/2006/02/20/535444.aspx)

Visio - the interaction designer's nail gun: [http://www.guui.com/issues/01\\_06.php](http://www.guui.com/issues/01_06.php)

PDF Prototyping: [http://www.gotomedia.com/gotoreport/may2005/news\\_0505\\_usable1.html](http://www.gotomedia.com/gotoreport/may2005/news_0505_usable1.html)

### Products

[www.axure.com/demo.aspx](http://www.axure.com/demo.aspx)

iRise Application Simulator: [irise.com](http://irise.com)

### Visio Stencils

[www.bboxesandarrows.com/view/storyboarding\\_rich\\_internet\\_applications\\_with\\_visio](http://www.bboxesandarrows.com/view/storyboarding_rich_internet_applications_with_visio)

Visio Stencil Library: [swipr.com](http://swipr.com)

Wireframe Stencils: <http://iainstitute.org/tools/>

Garrett Dimon's Stencils: <http://www.garrettdimon.com/resources/templates-stencils-for-visio-omnigraffle>

<http://protoscript.com>  
<http://looksgoodworkswell.com>