

# concepts software

Daniel Jackson · Google, Cambridge · May 8, 2012



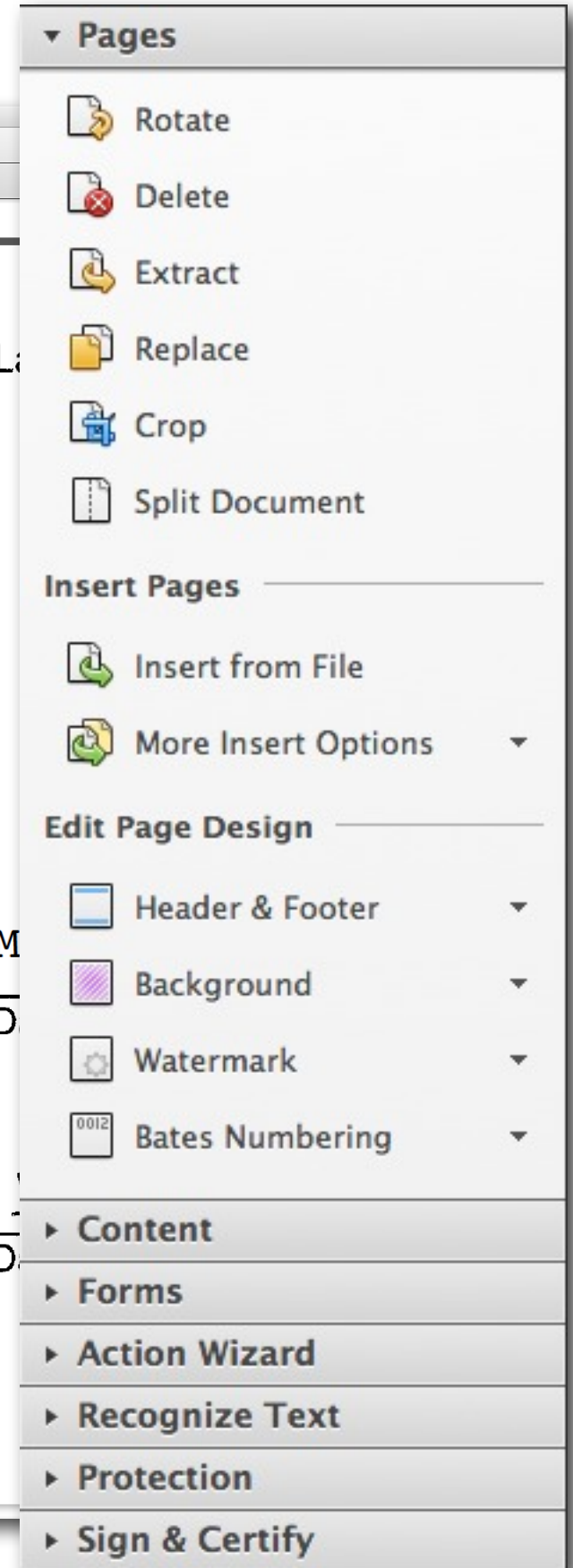
**CSAIL**

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

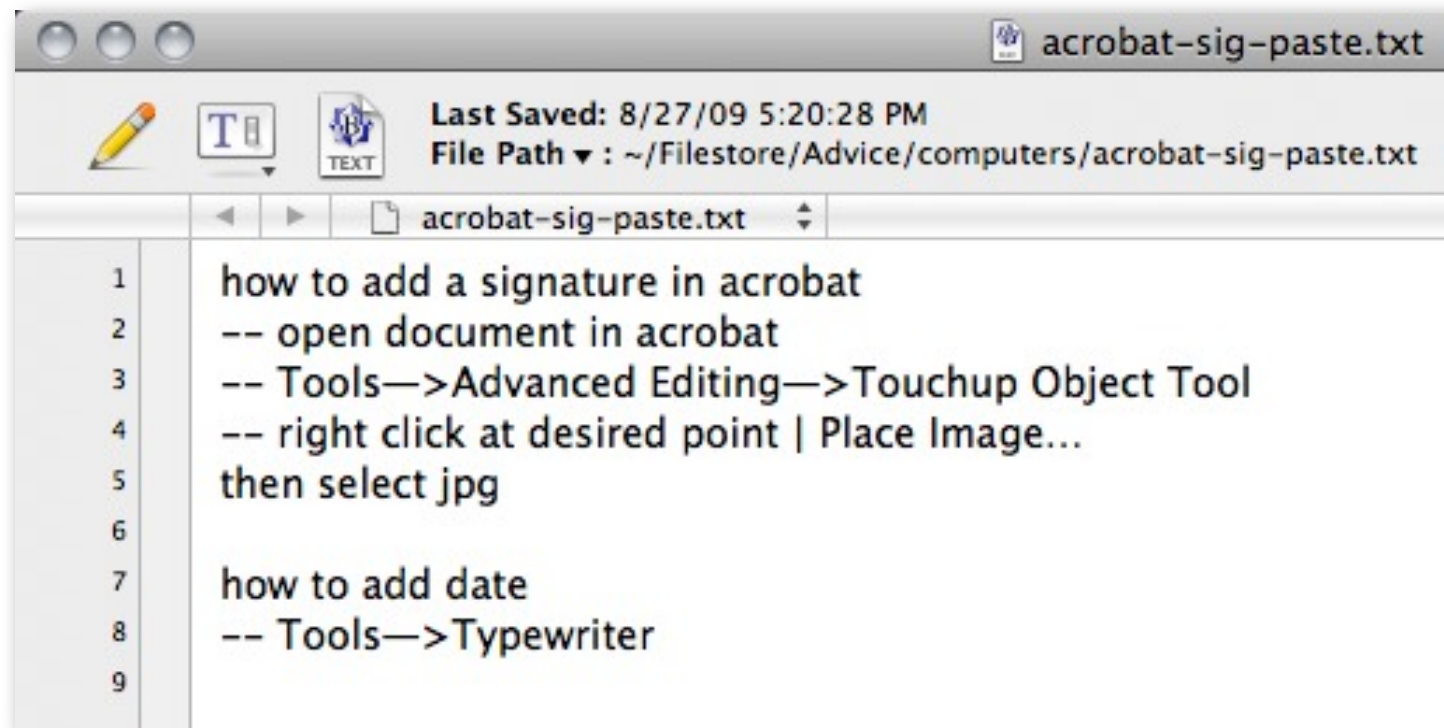
**#1**

**the good & the bad**

# adobe acrobat pro



MIT Computer Science & Artificial Intelligence Lab  
Daniel Jackson, 2012  
digital image





# adobe lightroom

The screenshot displays the Adobe Photoshop Lightroom 4 interface. The central workspace shows a black and white photograph of a computer workstation in a server room. The workstation includes a desk with a monitor, keyboard, mouse, and a tower PC. To the right, a server rack is filled with various electronic equipment. The background features a large, circular opening in a wall, possibly a doorway or a large vent.

The interface is divided into several panels:

- Library Grid:** Located on the left, it shows a grid of photo thumbnails. The current photo is selected, and its history and collections are visible.
- Presets:** A panel on the left side of the grid, showing various preset options like "Lightroom B&W Filter Presets", "Lightroom B&W Presets", etc.
- History:** A panel on the left side of the grid, showing a list of actions performed on the selected photo, such as "Export - Hard Drive", "Add Brush Stroke", "Print", "Vertical Perspective", etc.
- Collections:** A panel on the left side of the grid, showing a list of collections, including "Abbott's Dreams", "MIT Labs", "MIT Labs A", "MIT Labs A Plus", "MIT Labs AB", "MIT Labs ABC", "Museum Selection", and "Web portfolio".
- Basic Panel:** Located on the right side, it contains sliders for "Treatment" (Color/Black & White), "WB" (White Balance), "Temp", "Tint", "Tone" (Exposure, Recovery, Fill Light, Blacks), "Brightness", "Contrast", "Presence" (Clarity, Vibrance, Saturation), and "Tone Curve".
- Tone Curve Panel:** Located on the right side, it shows a graph for adjusting the tone curve, with a "Point Curve" dropdown set to "Medium Contrast".
- HSL / Color / B&W Panel:** Located on the right side, it shows sliders for "Black & White Mix" and "HSL" (Red, Orange, Yellow, Green, Aqua, Blue, Purple, Magenta).
- Split Toning Panel:** Located on the right side, it shows sliders for "Split Toning" and "Detail".
- Lens Corrections Panel:** Located on the right side, it shows sliders for "Profile" and "Manual" adjustments, including "Transform" (Distortion, Vertical, Horizontal, Rotate, Scale), "Lens Vignetting" (Amount, Midpoint), and "Defringe".
- Effects Panel:** Located on the right side, it shows a "Camera Calibration" section.

The top of the interface shows the "Library | Develop | Map | Book | Slideshow | Print | Web" menu. The bottom of the interface shows "Copy..." and "Paste" buttons.

# hypothesis

weak concepts	strong concepts
hard to use	intuitive, predictable
a mess to maintain	decoupling & localization
unreliable & buggy	more dependable

# #2

## how to do it?



# what we're already doing

thinking & sketching  
simulating features

normal design practice  
copying good ideas

evaluating products  
user feedback

discarding failed designs  
"refactoring"

*To design something really well, you have to get it. You have to really grok what it's all about. It takes a passionate commitment to really thoroughly understand something, chew it up, not just quickly swallow it. Most people don't take the time to do that. --Steve Jobs*

**what we're not doing**

**being explicit**

focusing: what are the concepts?

relating: how are they related to each other?

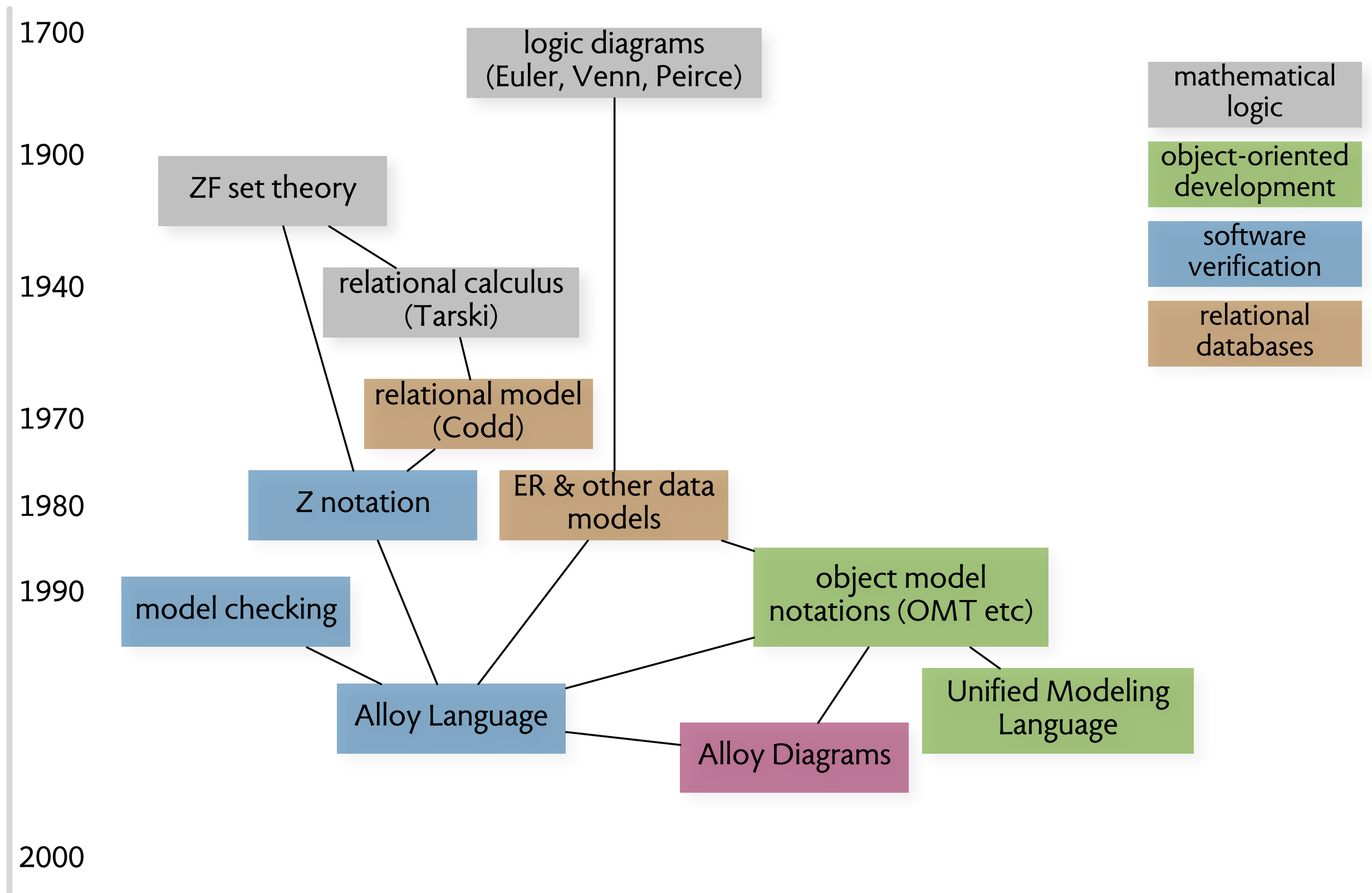
analyzing: what properties do they have?



**#3**

**an approach**

# alloy: a notation



# semantic concepts

## atom

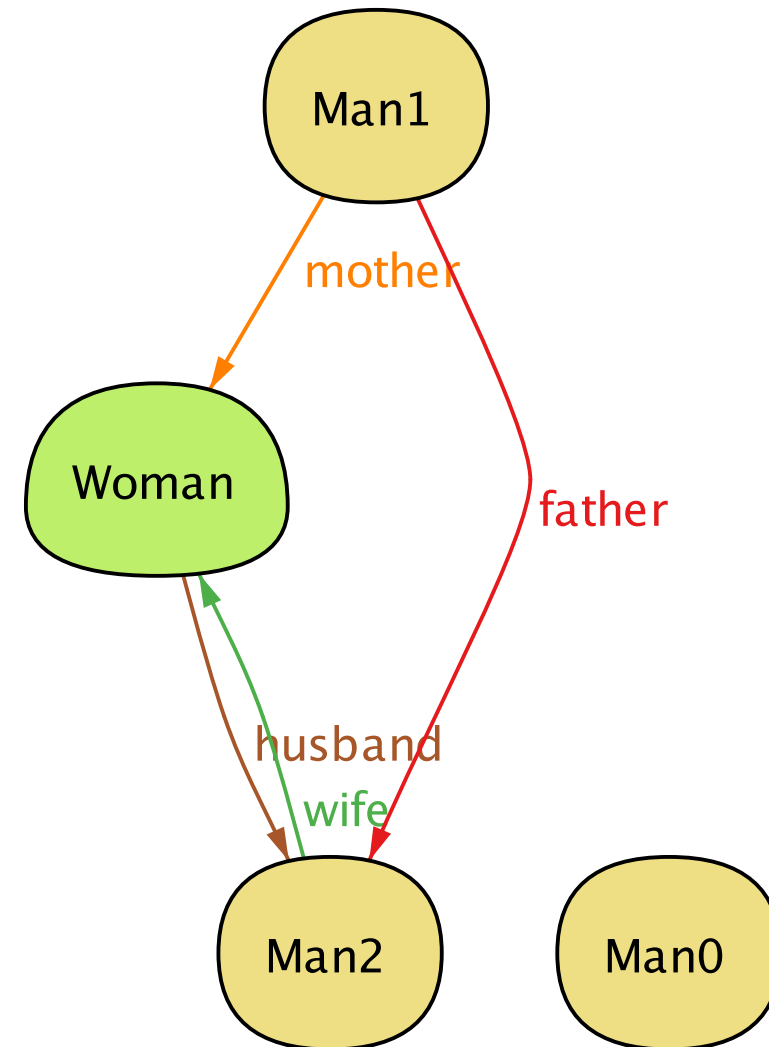
indivisible  
immutable  
uninterpreted

## relation

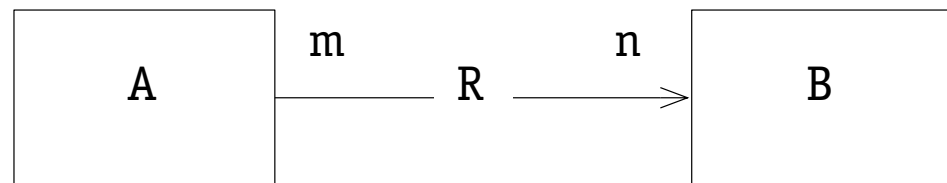
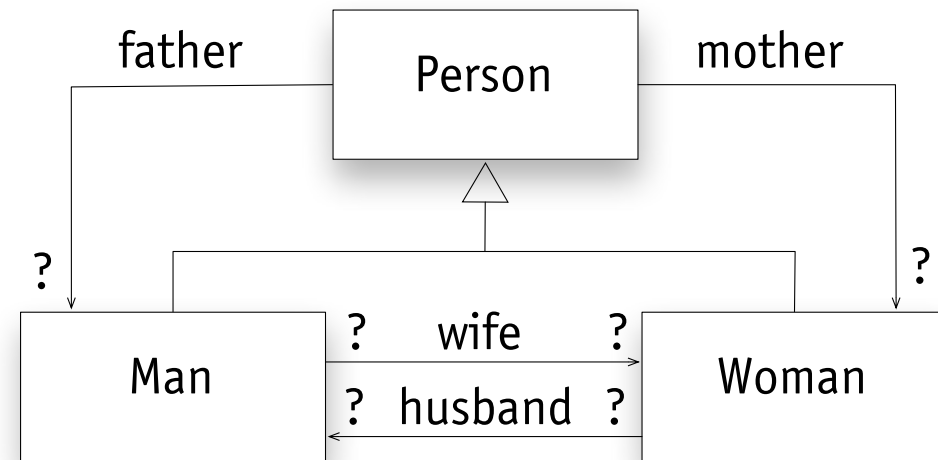
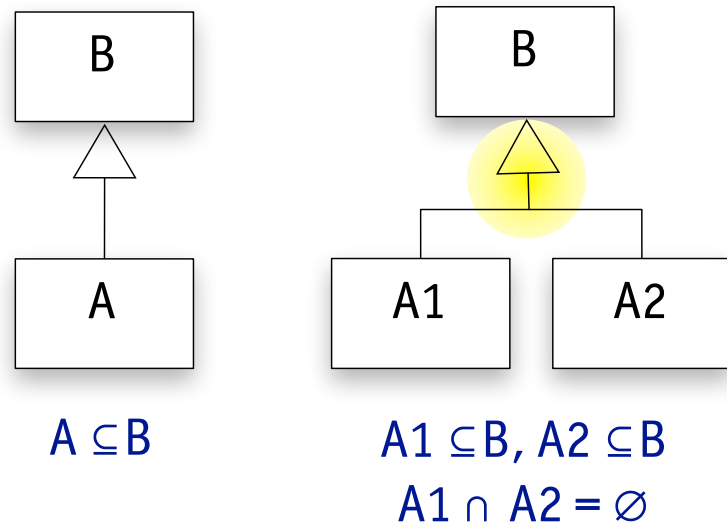
collection of atom tuples

## set

collection of atoms  
(ie, a unary relation)



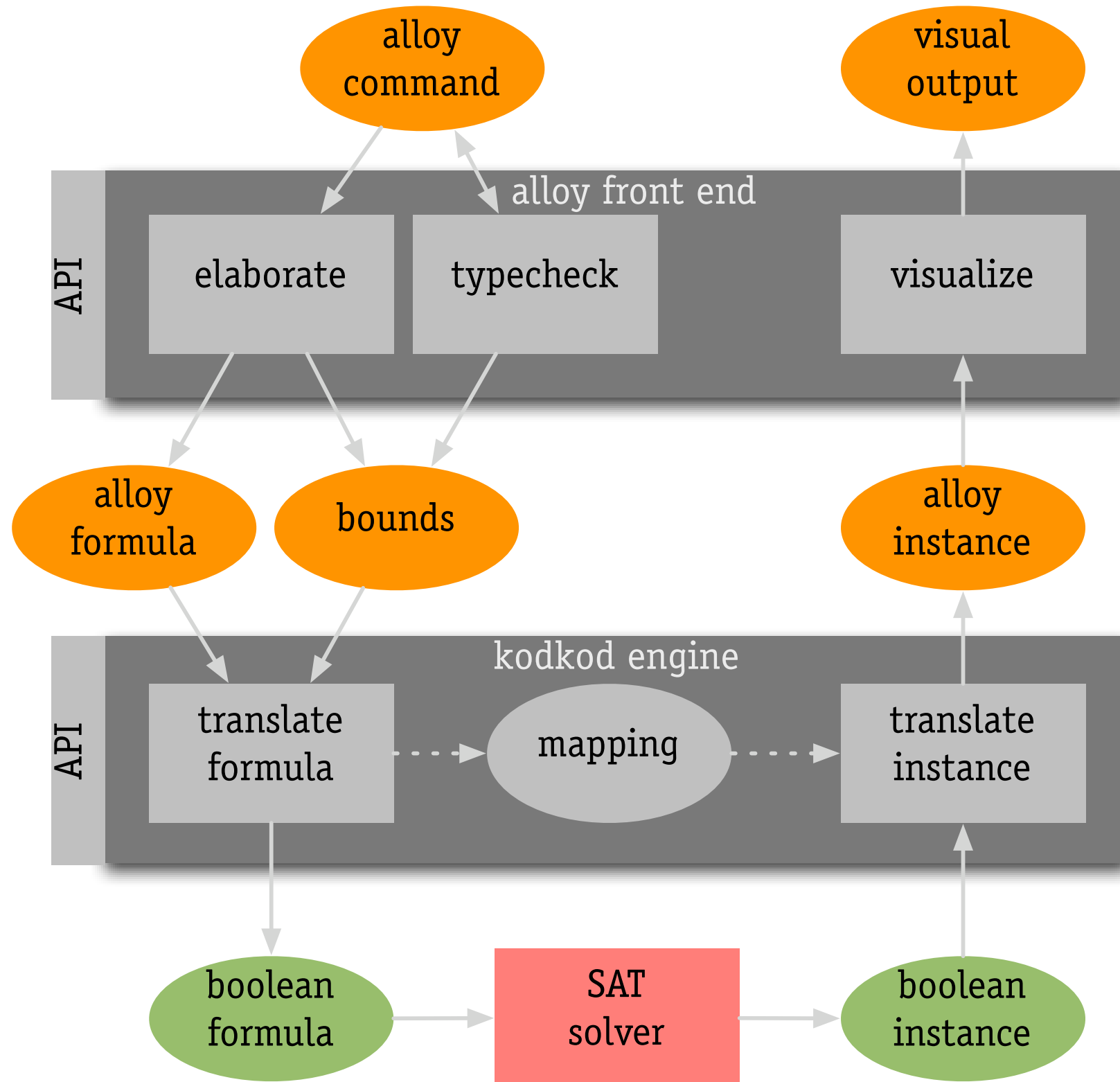
# graphical syntax



- R maps **m** A's to each B
- R maps each A to **n** B's

+ one or more  
\* zero or more  
! exactly one  
? at most one  
omitted = \*

# the alloy analyzer: a model finder





# i'm my own grandpa

Alloy Analyzer 4.1.10 (build date: 2009/03/19 02:02 EDT)

Executing "Run ownGrandpa for 4 Person"  
Solver=minisatprover(jni) Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20  
1257 vars. 76 primary vars. 1985 clauses. 130ms.  
Instance found. Predicate is consistent. 11ms.

```
/*
 * An Alloy model of the song "I Am My Own Grandpa"
 * by Dwight B. Latham and Moe Jaffe
 *
 * The challenge is to produce a man who is his own grandfather
 * without resorting to incest or time travel. Executing the predicate
 * "ownGrandpa" will demonstrate how such a thing can occur.
 *
 * The full song lyrics, which describe an isomorphic solution,
 * are included at the end of this file.
 *
 * model author: Daniel Jackson
 */

abstract sig Person {
  father: lone Man,
  mother: lone Woman
}

sig Man extends Person { wife: lone Woman }
sig Woman extends Person { husband: lone Man }

fact Biology { no p: Person | p in p.^(mother+father) }
fact Terminology { wife = ~husband }

fact SocialConvention {
  no wife & *(mother+father).mother
  no husband & *(mother+father).father
}

fun grandpas [p: Person]: set Person {
  let parent = mother + father + father.wife + mother.husband |
  p.parent.parent & Man
}

run {mother.husband = father} for 4 Person
pred ownGrandpa [m: Man] { m in grandpas[m] }
run ownGrandpa for 4 Person
```

(grandpa) Run ownGrandpa for 4 Person

Projection: none

Viz Dot XML Tree Theme Magic Layout Evaluator Next

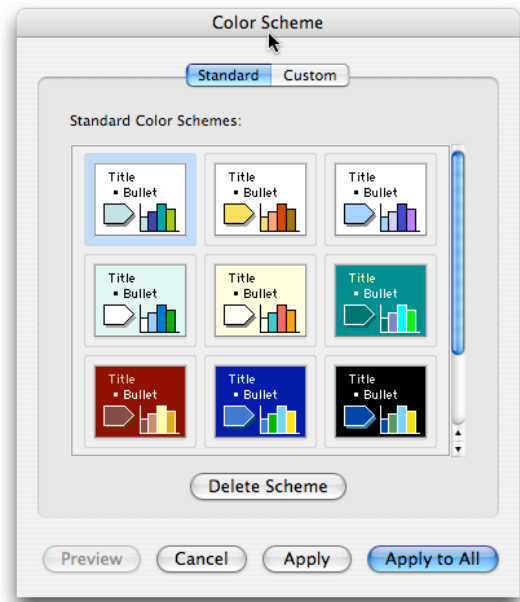
husband: 2  
mother: 2  
wife: 2

```
graph TD
  Man1((Man1))
  Man0((Man0 (m)))
  Woman0((Woman0))
  Woman1((Woman1))
  Man1 -- wife --> Woman0
  Man1 -- husband --> Woman1
  Man0 -- mother --> Woman0
  Man0 -- wife --> Woman1
  Woman0 -- mother --> Man1
  Woman0 -- wife --> Man0
  Woman1 -- mother --> Man0
  Woman1 -- wife --> Man1
```

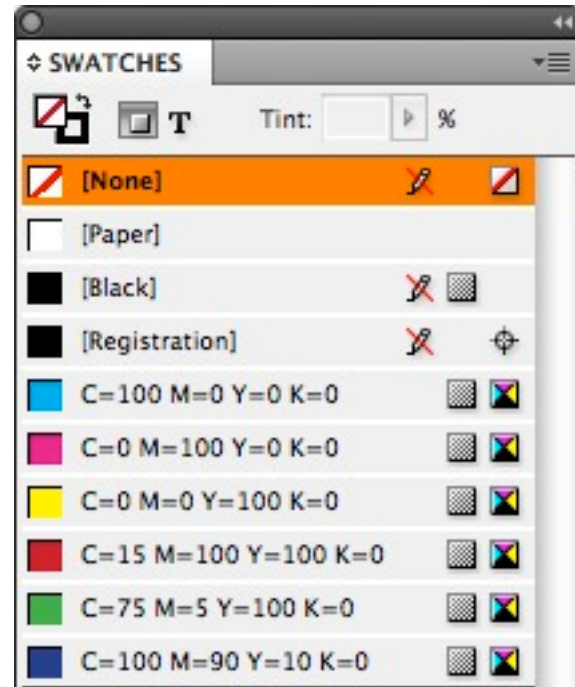
**#4**

**some generic concepts**

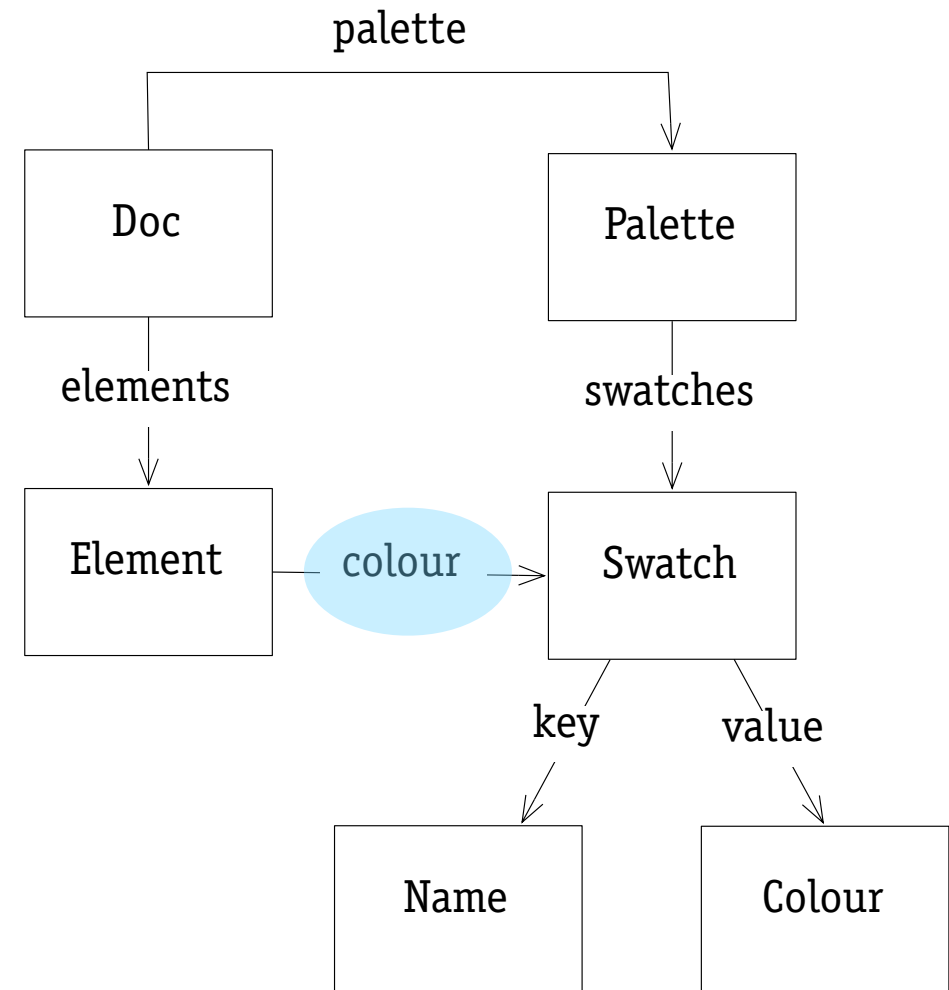
# "style"



Powerpoint



Indesign

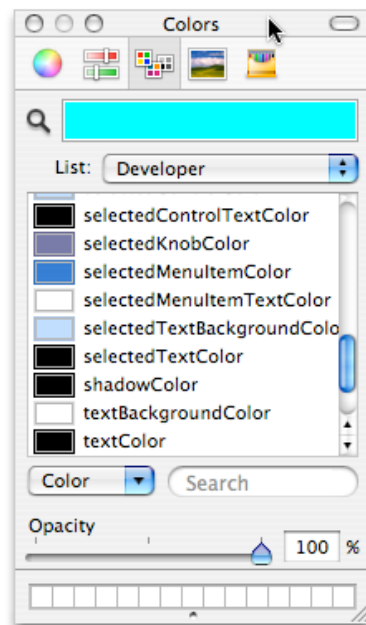


There is no problem in computer science that cannot be solved by introducing another level of indirection. --David Wheeler

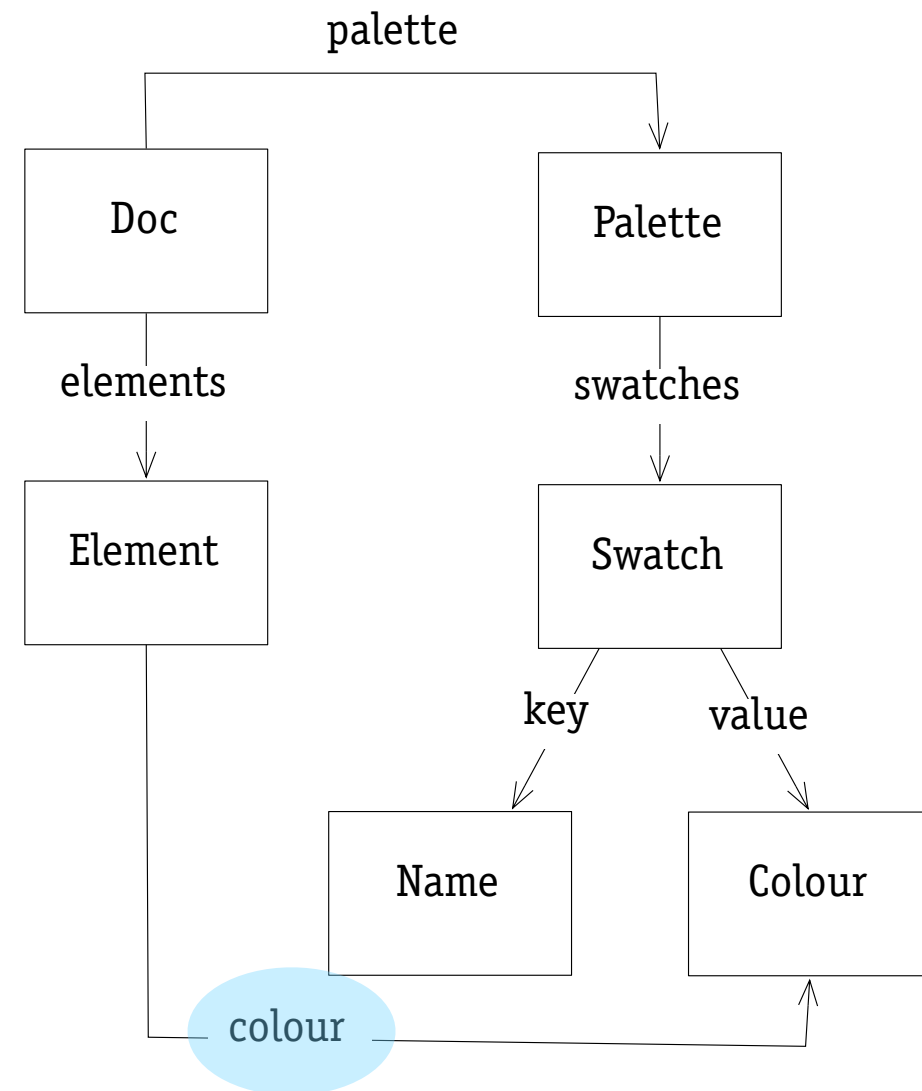
## rough edges

- › Indesign: can't tell whether you assigned color or swatch
- › CSS: formatting rules aren't independent

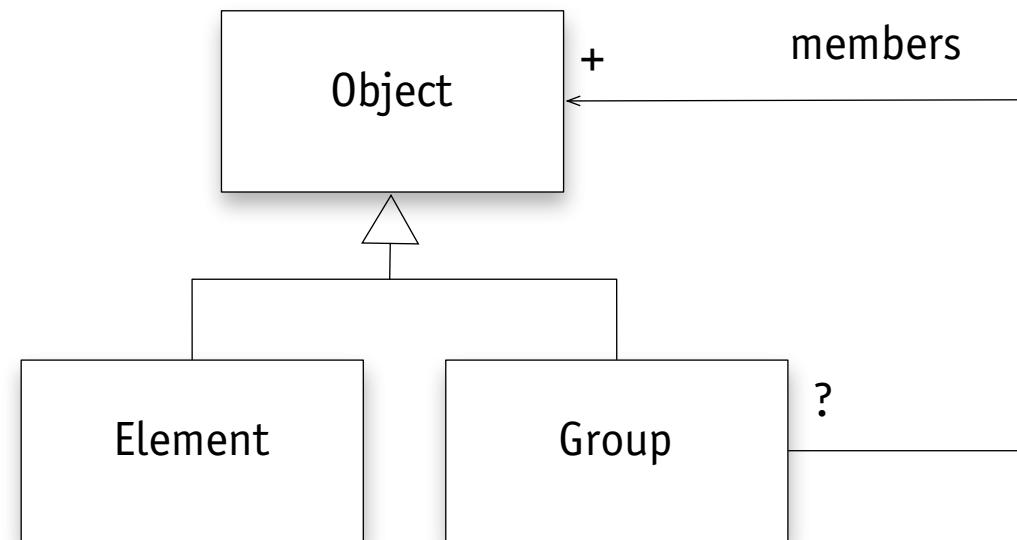
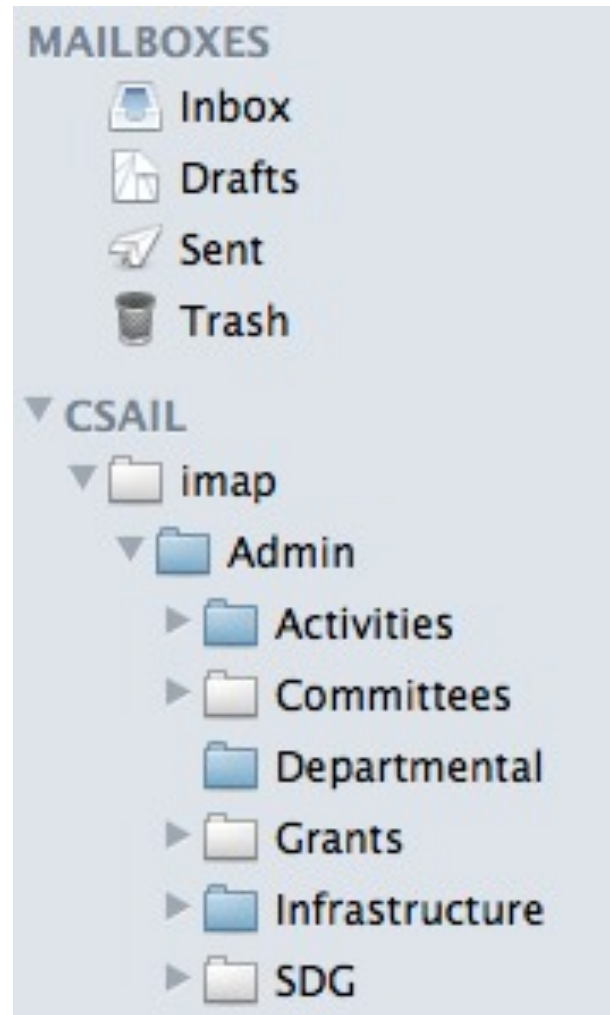
# "pseudo style"



**Keynote**



# “composite”



## rough edges

- › Lightroom: “collection sets”
- › IMAP vs Apple Mail: folder holding message *and* folder?
- › Google docs: collections a bit scary?



# "approval"

ALL | READER PICKS | NYT PICKS | Newest ▾ | Comments

 **Carolyn Egell** · Valley Lee, Md.

As always, the battle is set between the haves, and the rest. The haves have just lost out, and the movement back towards the greater good has begun. Let's hope the American public sees the sense in what the French and Greeks have done yesterday, by reclaiming their governments to represent the people for a change. The stock markets will likely be unhappy.

May 7, 2012 at 3:23 p.m. · RECOMMEND  8

7

Ok, I've rolled my own theme, I've made a custom jQuery UI pack (progress bar, date picker, slider) and installed it all. Seems to (mostly) work except for two things:

1. When my page first loads the datepicker div is visible; and
2. The text "Next" and "Prev" are visible in large font underneath my icons. None of the examples seem to have this problem.

Now (1) I'm currently solving by:

```
#ui-datepicker-div { display: none; }
```


in another CSS file but again none of the demos seem to need this.

What am I missing?

[javascript](#) [jquery](#) [css](#) [jquery-ui](#)

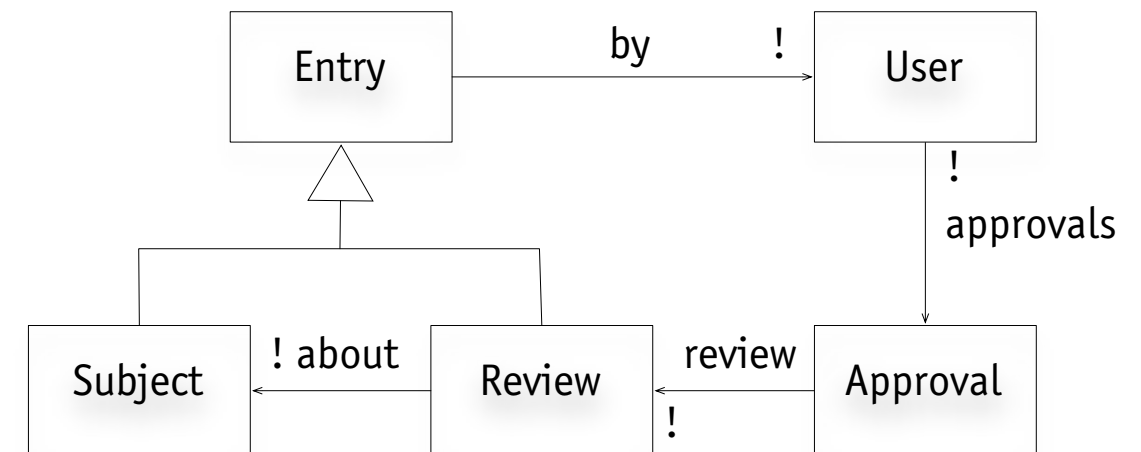
[link](#) | improve this question

asked Jan 29 '09 at 6:56

 **cletus**  
169k · 34 · 337 · 588  
83% accept rate

Boot up firebug, highlight the buttons and find out what styles are being assigned. Then post them here, that should help us debug your problem. — Mike Robinson Jan 29 '09 at 22:10

Was this post useful to you?  Yes  No



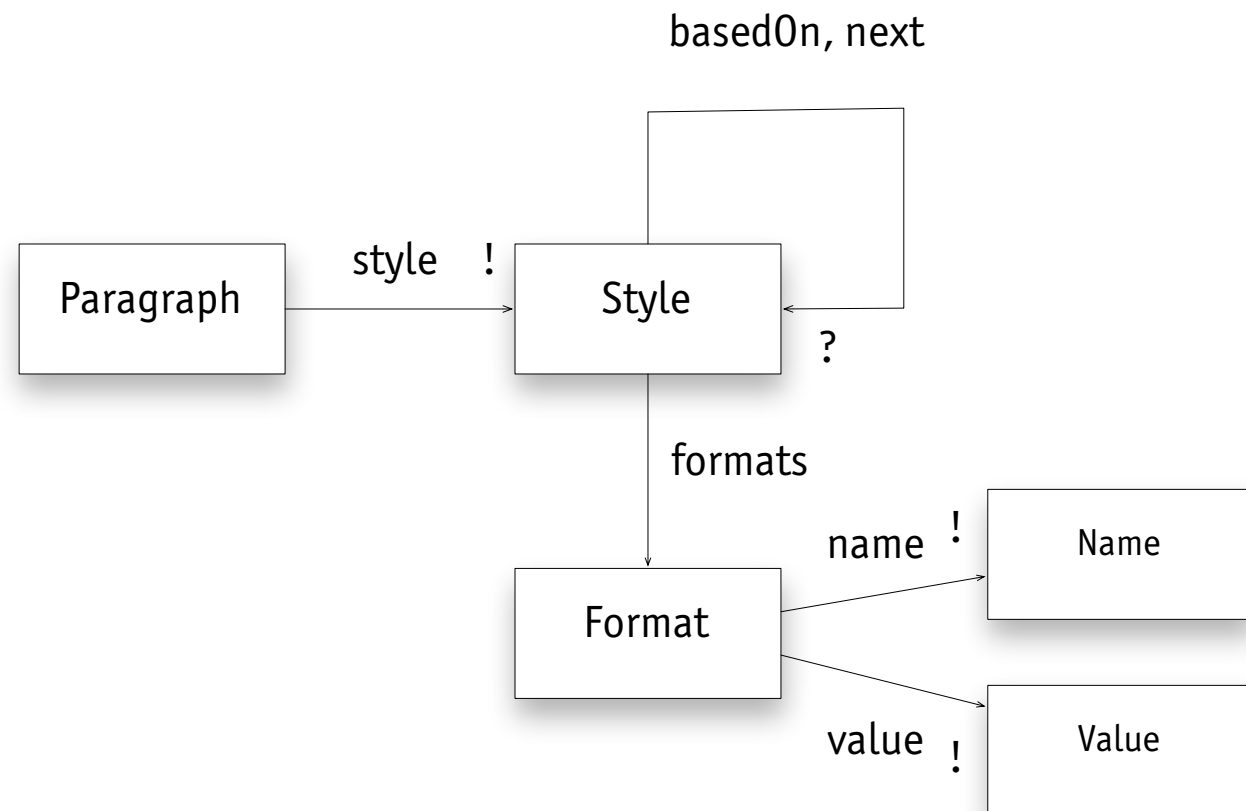
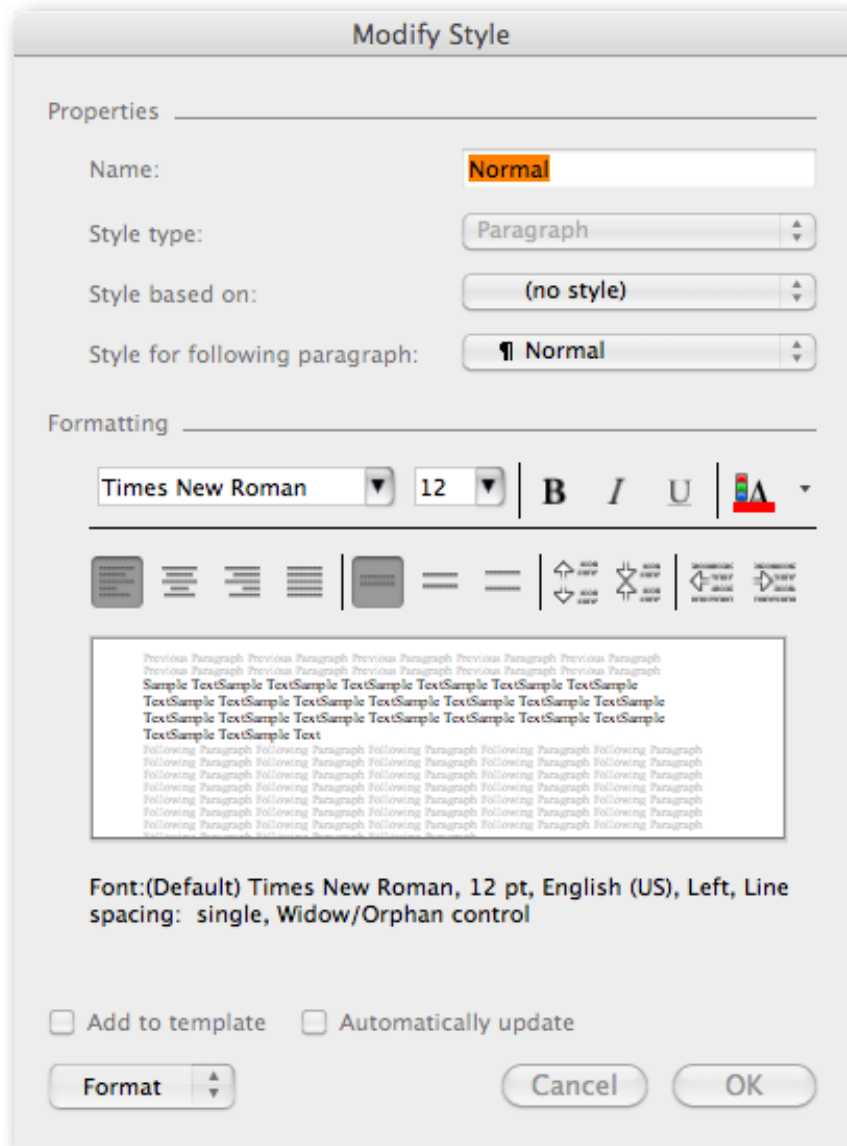
rough edges

› your suggestions?

**#5**

**three conceptual models**

# microsoft word

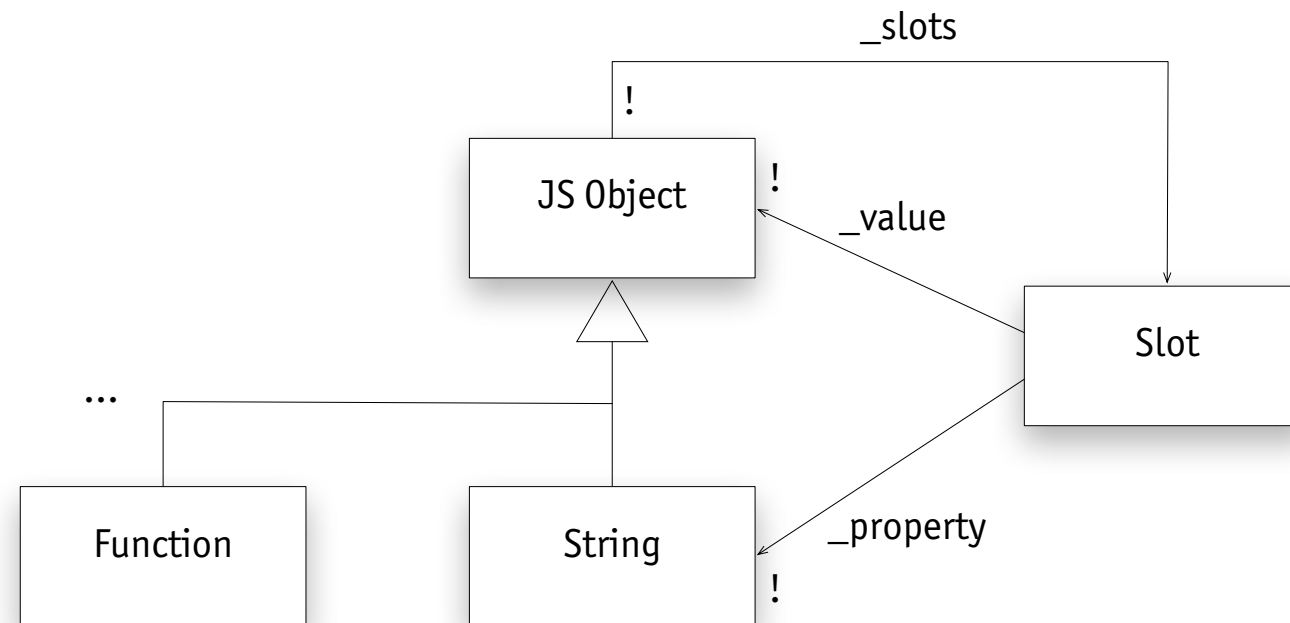


rough edges

- › special role of Normal style, etc
- › hidden memory of inherit vs replace with same

**the origins of paragraph styles**  
Bravo-X at Xerox PARC: Tim Mott, Larry Tesler, Charles Simonyi; first commercialized in Word, now ubiquitous (Pages, Indesign, Quark,...)

# javascript objects



```
gallery.play = function () {
  change(SP.mode.SLIDESHOW);
  gallery.next_photo();
  autoplayTimer = setInterval(autonext, prefs.transitionTime);
}
gallery.play.enabled = function () {
  return mode != SP.mode.SLIDESHOW && prefs.enabledModes[SP.mode.SLIDESHOW];
}
```

## rough edges

› add slots to all objects? is 23 an object?

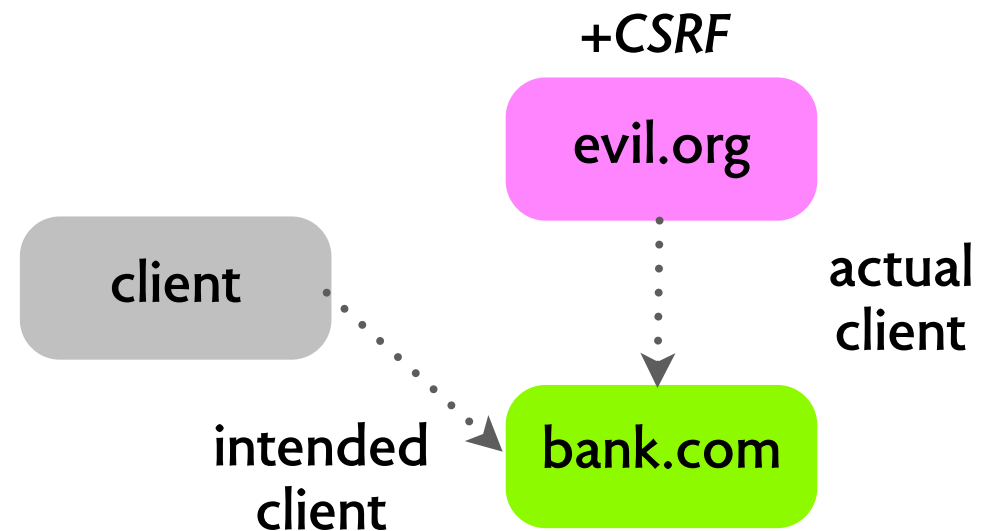
**origins & referrers**



# referrers & origins

```

```



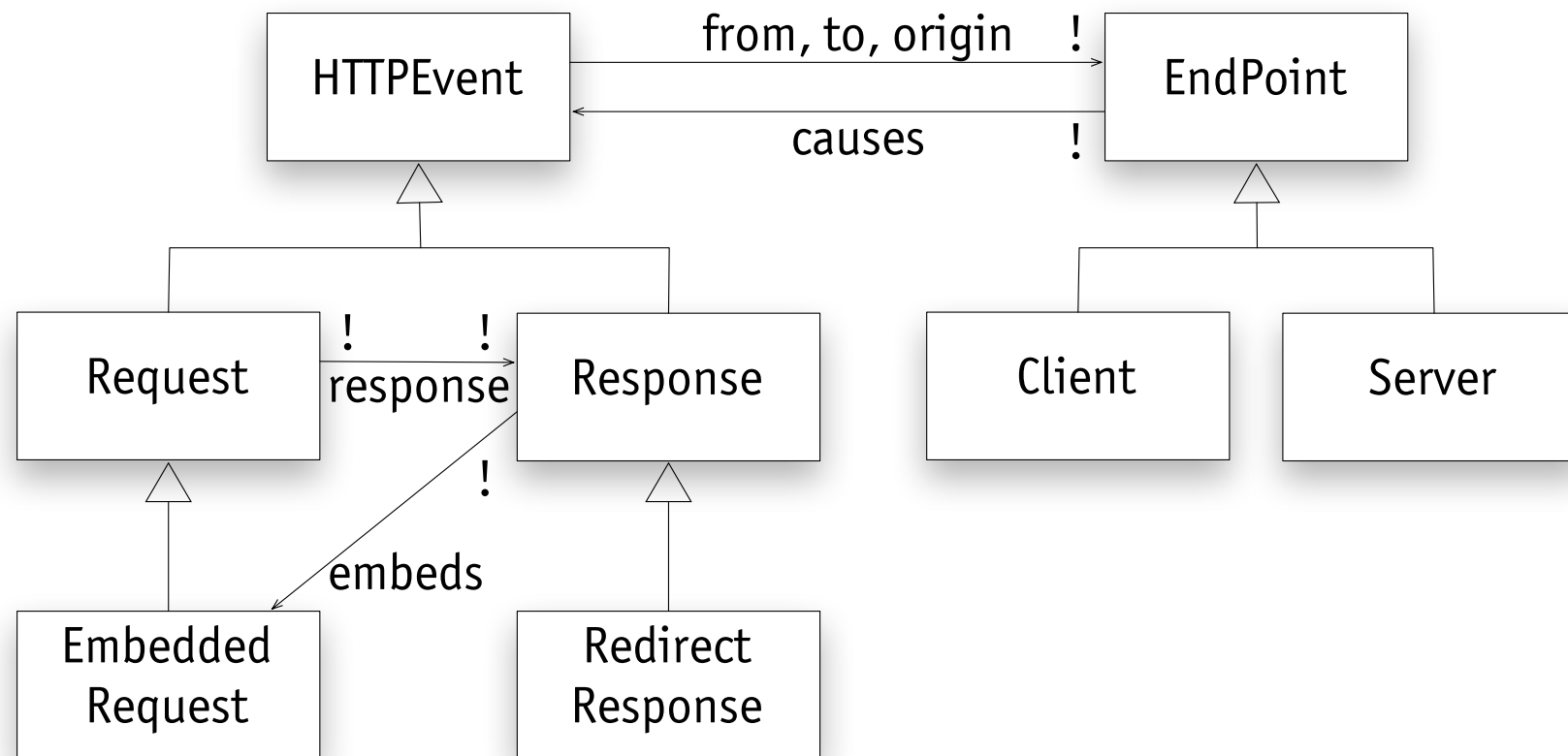
**a strategy for XSS and CSRF**

browser tracks "origin" of each request  
with HTTP request, includes origin as "referrer"  
if referrer is not self, server rejects it

**critical property**

$s = \text{origin}(r)$  iff  $s$  really is cause of  $r$

# modeling origins



# define basic concepts

```
abstract sig HTTPEvent {from, to, origin: EndPoint}
abstract sig EndPoint { causes: set HTTPEvent }
  { causes = {e: HTTPEvent - Embedded | e.from = this} + causes.embeds }

sig Client, Server extends EndPoint {}

sig Request extends HTTPEvent { response: Response }
  { from in Client and to in Server }

sig Response extends HTTPEvent { embeds: set Embedded }
  { from in Server and to in Client }

sig Embedded extends Request {}
fact {Embedded = Response.embeds}

sig Redirect extends Response {}

fact RequestResponse {
  response in Request one -> one Response
  all r: Request | r.from = r.response.to and r.to = r.response.from
}
```

# define origin tracking

```
fact Origin {  
  // for a redirect, origin is same as request, else server  
  all r: Request | r.response.origin =  
    (r.response in Redirect implies r.origin else r.response.from)  
  
  // embedded requests have the same origin as the response  
  all r: Response, e: r.embeds | e.origin = r.origin  
  
  // requests that are not embedded come from the client  
  all r: Request - Embedded | r.origin = r.from  
}
```

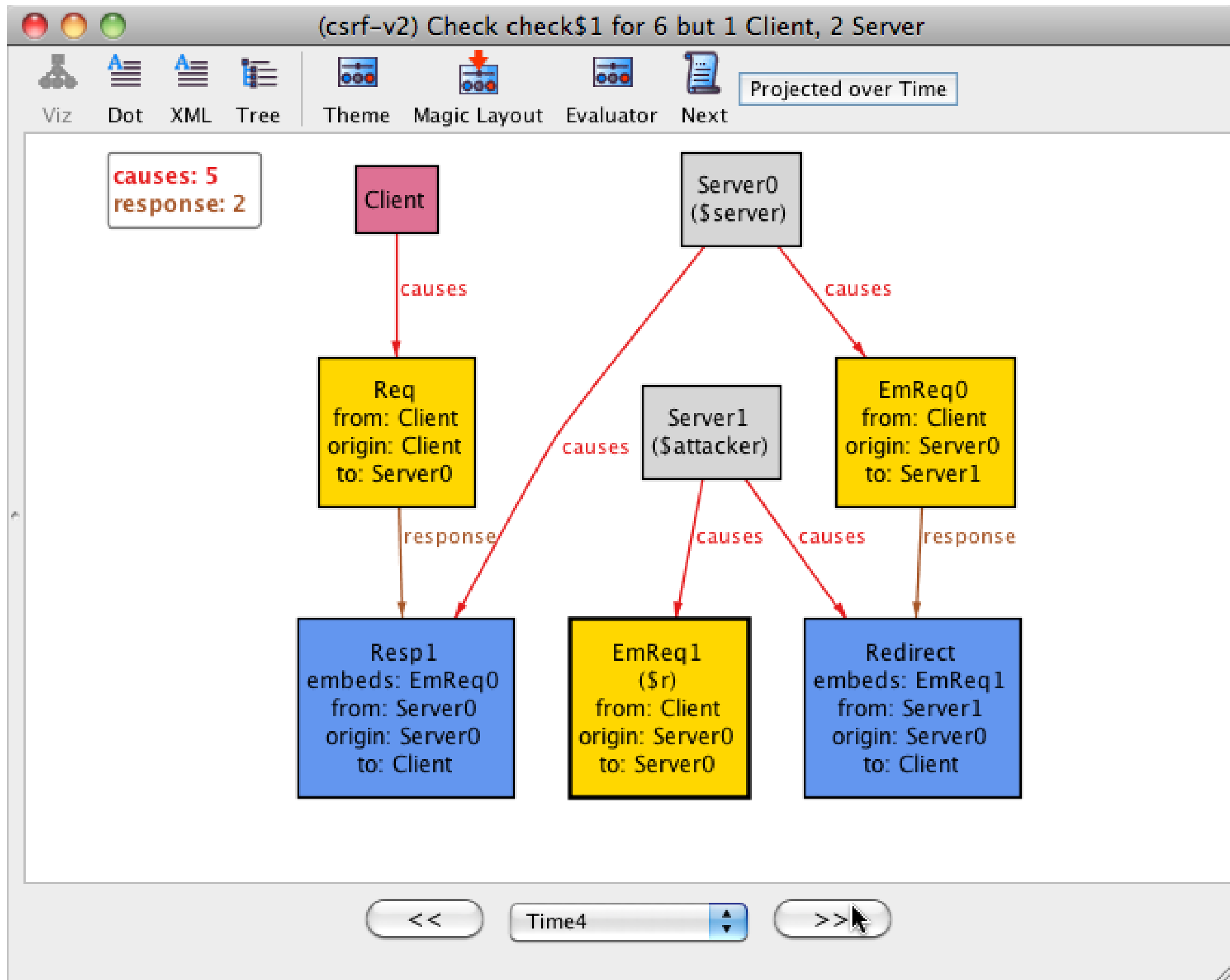
```
pred appliesSOP (s: Server) {  
  // request is only accepted if origin is server itself or sender  
  all r: Request | r.to = s implies r.origin = r.to or r.origin = r.from  
}
```

# does the policy work?

```
check {  
  no server: Server, attacker: Server - server {  
    // no direct request to attacker  
    no r: Request | r.to = attacker and r.origin in Client  
    // trusted server obeys origin policy  
    server.appliesSOP  
    // and attacker still gets request through  
    some r: attacker.causes | r.to = server  
  }  
} for 6 but 1 Client, 2 Server
```



# counterexample!



# Towards a Formal Foundation of Web Security [2010]

## Akhawe, Barth, Lam, Mitchell & Song

generic model of web security  
HTTP, certificates, cookies, script contexts  
about 2,000 lines of Alloy

Case Study	Lines of new code	No. of clauses	CNF gen. time (sec)	CNF solve time (sec)
Origin Header	25	977,829	26.45	19.47
CORS	80	584,158	24.07	82.76
Referer Validation	35	974,924	30.75	9.06
HTML5 Forms	20	976,174	27.67	73.54
WebAuth	214	355,093	602.4	35.44

applied to 5 case studies  
in each, found vulnerabilities  
2 known, 3 unknown

# more examples: alloy.mit.edu

[about](#)[community](#)[download](#)[documentation](#)[book](#)[applications](#)[people](#)[thanks](#)

## alloy: a language & tool for relational models

### about alloy

Alloy is a language for describing structures and a tool for exploring them. It has been used in a wide range of [applications](#) from finding holes in security mechanisms to designing telephone switching networks.

An Alloy model is a collection of constraints that describes (implicitly) a set of structures, for example: all the possible security configurations of a web application, or all the possible topologies of a switching network. Alloy's tool, the [Alloy Analyzer](#), is a solver that takes the constraints of a model and finds structures that satisfy them. It can be used both to explore the model by generating sample structures, and to check properties of the model by generating counterexamples. Structures are displayed graphically, and their appearance can be customized for the domain at hand.

At its core, the Alloy language is a simple but expressive logic based on the notion of relations, and was inspired by the Z specification language and Tarski's relational calculus. Alloy's syntax is designed to make it easy to build models incrementally, and was influenced by modeling languages (such as the object models of OMT and UML). Novel features of Alloy include a rich subtype facility for factoring out common features and a uniform and powerful syntax for navigation expressions.

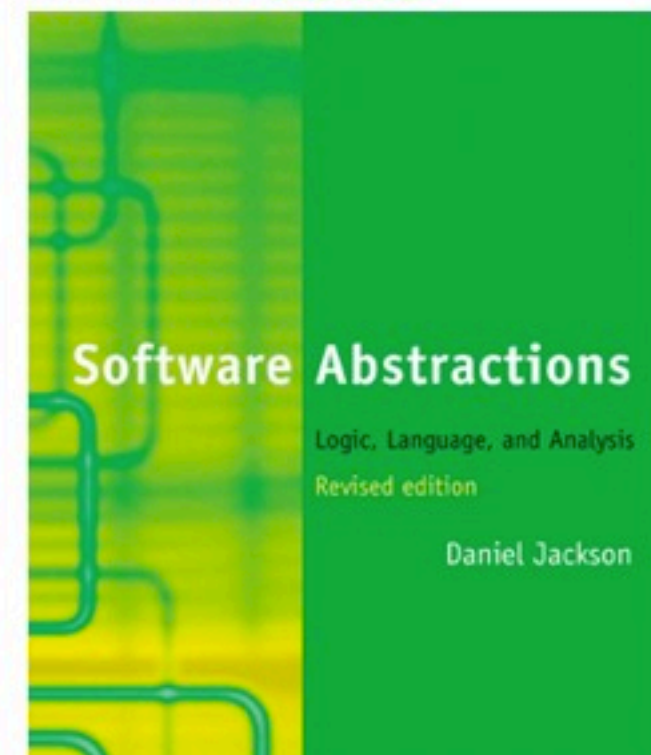
The Alloy Analyzer works by reduction to SAT. Version 4 was a complete rewrite that included [Kodkod](#), a new model finding engine that optimizes the reduction, and a new front end.

### news

[ASM, Alloy, B and Z Conference](#): papers now due January 22!

Research programmer position [available](#) on Alloy project!

Revised edition of book now out!  
Available from [MIT Press](#).



**#6**

**anti-patterns**

## **non-uniformity**

**members of set have different properties or behaviors**

eg: in Photoshop, base layer is different

## **coupling**

**concepts are not independent**

eg: in OS X, folder view vs. network access

eg: in CSS, element position vs. wrap around

## **over-generalization**

**distinct concepts merged**

eg: in mail clients, trashed messages have no deletion date



# unity of purpose?

Conceptual integrity is the most important consideration in system design. It is better to have a system **omit certain anomalous features** and improvements, but to reflect one set of design ideas, than to have one that contains many good but independent and uncoordinated ideas.

-- Fred Brooks, 1975

**thank you!**

# squander

```
public class Sudoku {
    private int [][] grid = new int [9][9];

    @Ensures ({
        "all row in {0..8} | this.grid[row][int] = {1..9}",
        "all col in {0..8} | this.grid[int][col] = {1..9}",
        "all r , c in {0, 1, 2} |
            this.grid[{r*3..r*3+2}][{c*3..c*3+2}] = {1..9}"
    })
    @Modifies("this.grid[int].elems | _<2> = 0")
    public void solve() { Squander.exe(this); }

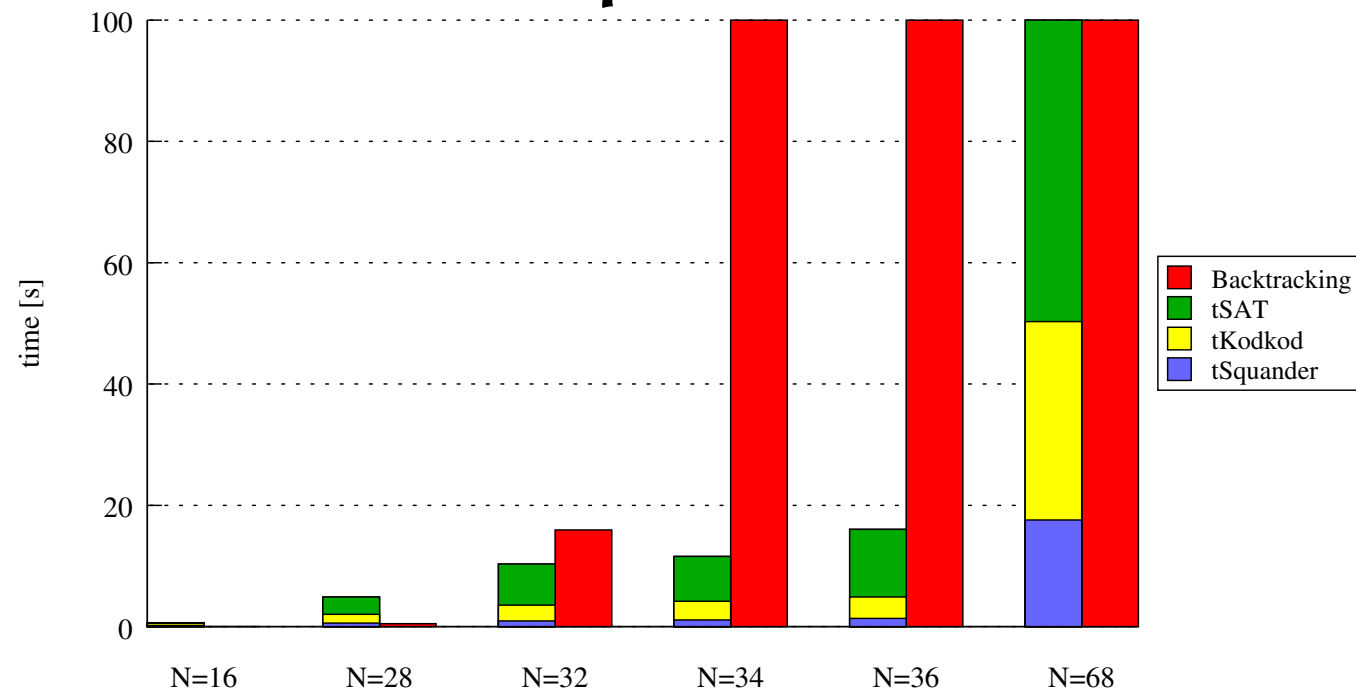
    public static void main(String[] args) {
        Sudoku s = new Sudoku();
        s.grid[0][3] = 1; ...; s.grid[8][8] = 5;
        s.solve( );
        System.out.println(s);
    }
}
```

6			1		8	2		3
	2			4			9	
8		3			5	4		
5		4	6		7			9
	3						5	
7			8		3	1		2
		1	7			9		6
	8			3			2	
3		2	9		4			5

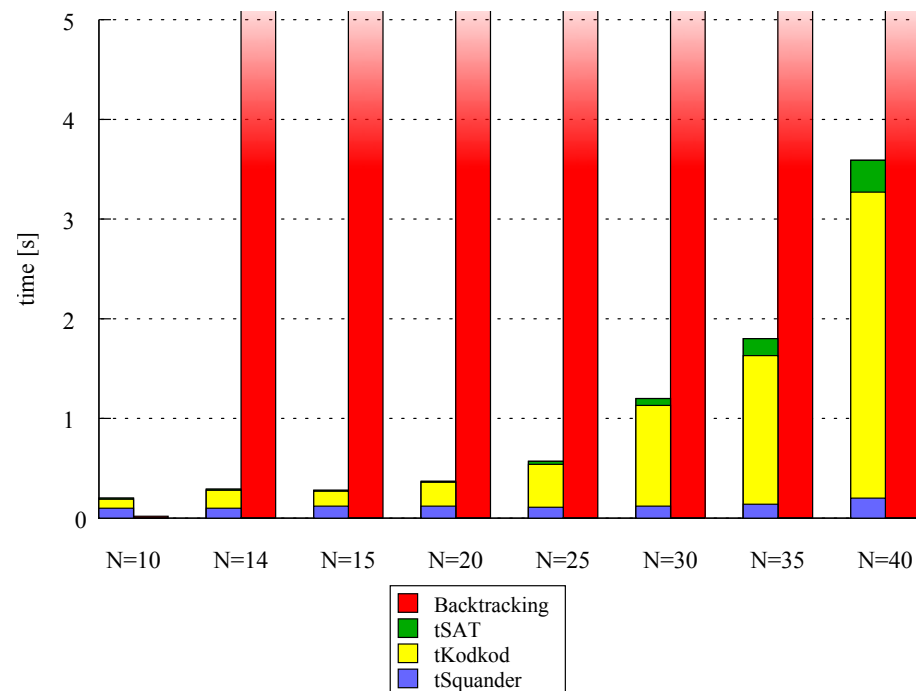


# performance

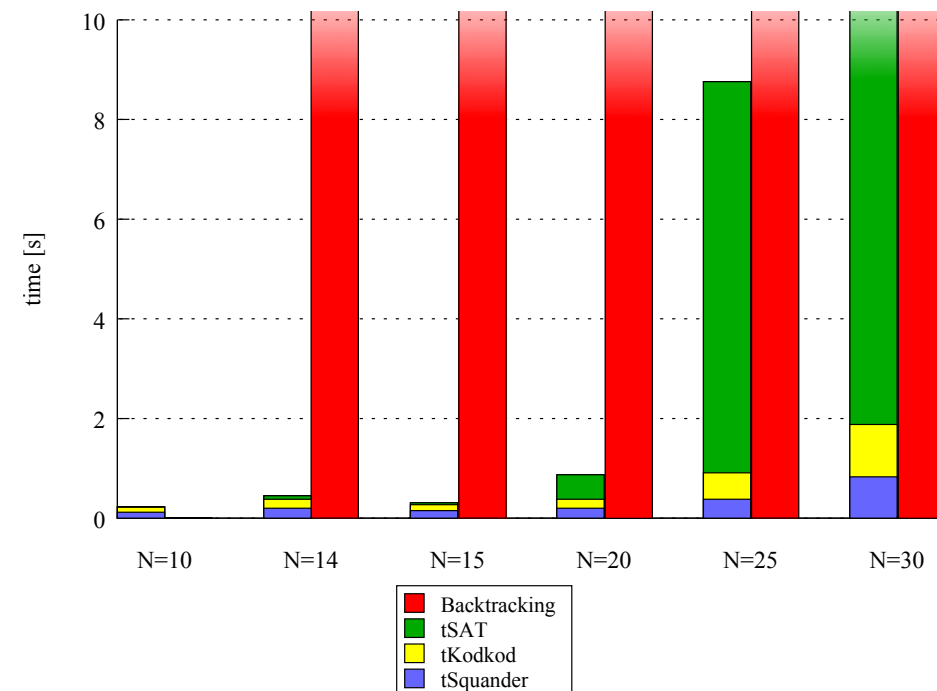
## *n*-queens



## *hamiltonian path, none*



## *hamiltonian path, some*



# Rubicon specs

```
it "user included in list of users" do  
  user = Factory(:user)  
  get :index  
  assigns[:users].should include user  
end
```

**RSpec test**

```
it "all users included in list of users" do  
  User.forall do |user|  
    get :index  
    assigns[:users].should include(user)  
  end  
end
```

**Rubicon spec**

# Fat Free CRM

The screenshot displays the Fat Free CRM web application. The browser address bar shows the URL `http://localhost:3000/opportunities/37a98572-ade8-102c-b57b-6261c0a0e289`. The application header includes the title "Fat Free CRM", a user greeting "Welcome, Heather!", and navigation links for "Quick find", "Preferences", "Profile", and "Logout". A main navigation bar contains tabs for "Dashboard", "Tasks", "Campaigns", "Leads", "Accounts", "Contacts", and "Opportunities".

The "Opportunity Summary" section on the left lists the following details:

- Stage: Analysis
- Close date: Jul 21
- Days left: 29
- Probability: 80%
- Amount: \$800,000
- Discount: N/A
- Weighted amount: \$640,000
- Assigned to: Elizabeth Emul...
- Account: Volkman-Gerhold
- Campaign: The quicker pi...

The main content area displays the opportunity title "Quibusdam quasi unde" with "Edit" and "Delete" links. Below the title is a "Add a new note..." input field. A list of notes follows, each with a user profile picture, name, and timestamp:

- Dan Debugger** about 1 month ago: Voluptates ea tenetur ducimus quis cum iure aspernatur consequatur. Doloribus facere non minima quis maxime corporis aliquid quia. Eligendi quas et doloremque maxime. Pariatur consequatur quia error est totam ipsam.
- Frank Formatter** about 1 month ago: Praesentium repellendus dicta quibusdam. Ullam voluptatum soluta tenetur et. Ut similique et illo occaecati accusantium.
- Heather Hash** about 1 month ago: Id tempora commodi dolor dolores eum est. Quibusdam modi in laborum sed eos non. A aut totam tempore amet qui veniam et et. Et ad ut et.
- Frank Formatter** about 1 month ago: Dolor est et et natus repellendus fugit suscipit consequatur. Omnis temporibus repellat error. Pariatur quisquam quod assumenda ut consequatur et porro consequatur. Rerum ea vel soluta iusto doloremque.

The "Tasks" section includes a "Create Task" link and two task entries:

- Follow-up** Cindy Cluster: Donec sit amet ante mauris, at mattis enim. (re: Quibusdam quasi unde) - due now.
- Email** Lorem ipsum dolor sit. (re: Quibusdam quasi unde) - completed 5 minutes ago

The "Contacts" section features a "Create Contact" link and a list of contacts:

- Franco Sebert** Account Manager at Volkman-Gerhold. `francos@gmail.com` | phone: (427)085-2648 | mobile: (427)085-2644 | added 24 days ago. Includes "Edit" and "Delete" links.
- Dell Monahan** Executive Assistant at Volkman-Gerhold. `dellm@yahoo.com` | phone: (393)333-3236 | mobile: (393)474-2170 | added about 1 month ago.
- Johnpaul Wurtall** VP of Sales at Volkman-Gerhold. `johnpaulw@gmail.com` | phone: (888)047-8555 | mobile: (408)555-5761 | added 29 days ago.

At the bottom left, there is a "Go to # on this page" link.

# prototype Apache analyzer

Apache Configuration Analyzer

The Apache Configuration Analyzer started successfully.  
An Apache config file successfully loaded.  
The document root successfully specified.  
Analyzing the input configuration...

Analysis Report

**A potential security vulnerability detected in the input configuration!**

**Security Failure:**  
The web server exposes the contents of directory \$DOCROOT.

**Threat:**  
A potentially malicious client from "102.169.118.40" issues a request for the listing of directory \$DOCROOT.

**Vulnerability:**  
The global configuration is missing a directive to control the listing of directory contents.

**Recommended Mitigation:**  
Add an "Indexes" option to the global configuration file to disable the listing of directory contents.

```
graph TD; ApacheWebServer[ApacheWebServer] -- received --> Request((Request (Malicious) op: GET)); Request -- target --> $DOCROOT[$DOCROOT]; Request -- source --> IP[102.169.118.40]; Root["/"]; Root --> www[/var/www/]; Root --> cgi[/usr/lib/cgi-bin/]; Root --> share[/usr/share/doc/]; Root --> $DOCROOT; $DOCROOT --> dir1[$DOCROOT/dir1/]; $DOCROOT --> calendar[$DOCROOT/calendar/]; $DOCROOT --> financial[$DOCROOT/financial/]; $DOCROOT --> meetings[$DOCROOT/meetings/]; $DOCROOT --> models[$DOCROOT/models/]; dir1 --> nyreports[nyreports]; dir1 --> myfiles1[$DOCROOT/dir1/myfiles1];
```

Global configuration file: /etc/apache2/apache2.conf  
Document root path: /home/eskang/public\_html