concepts software

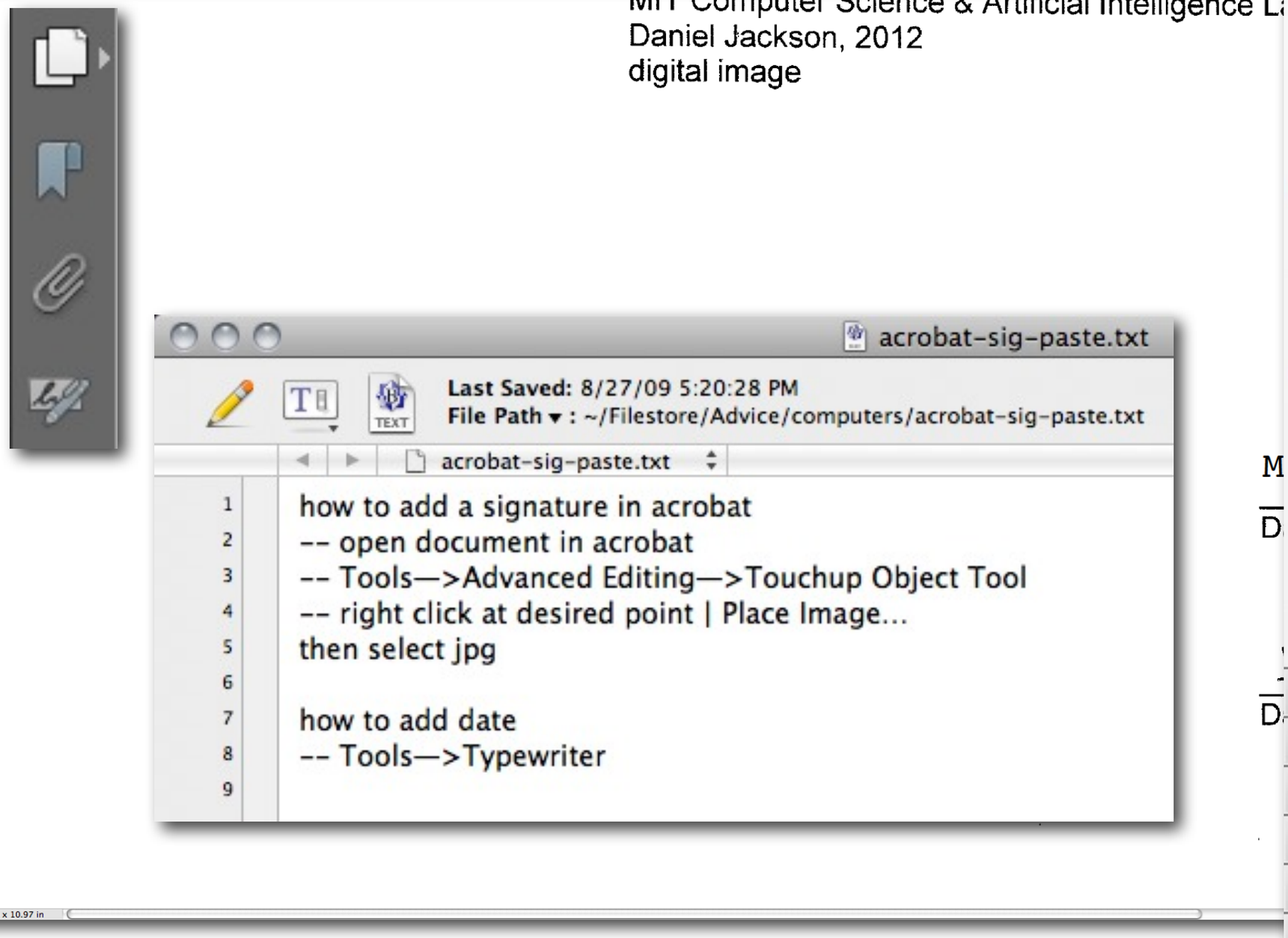
Daniel Jackson · Google, Cambridge · May 8, 2012

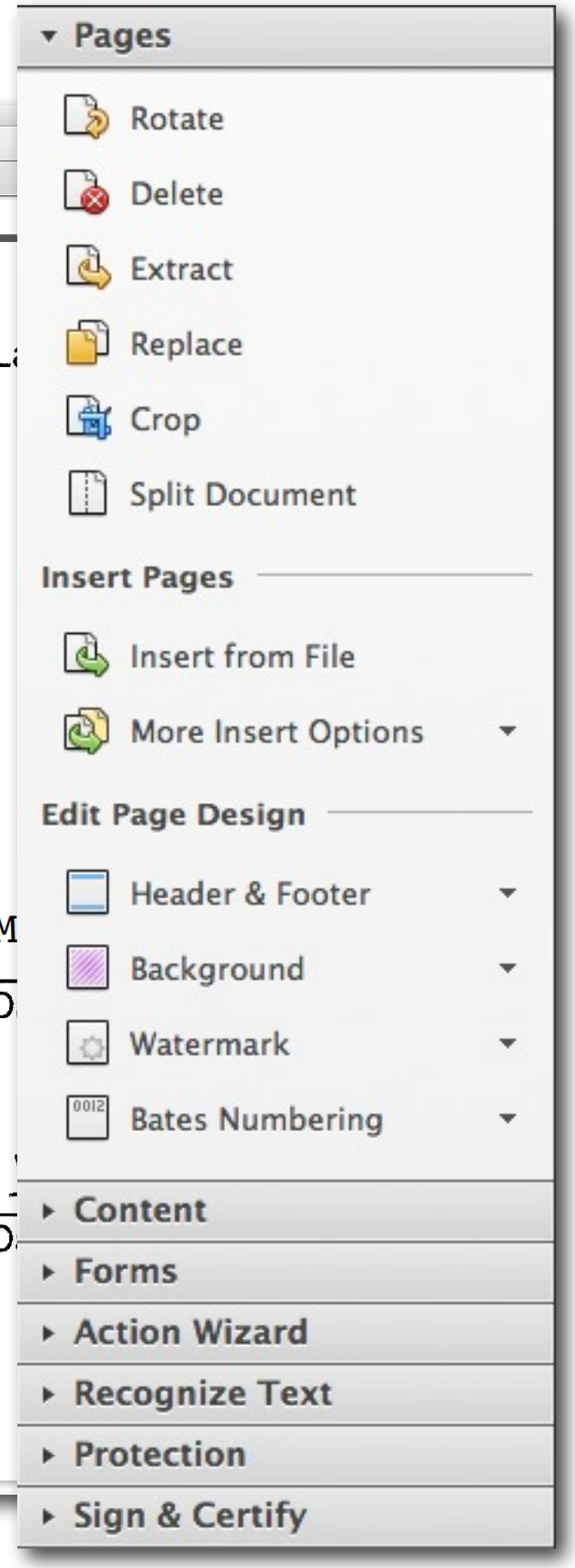
CSAIL
MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

#1

the good & the bad

adobe acrobat pro





adobe lightroom

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

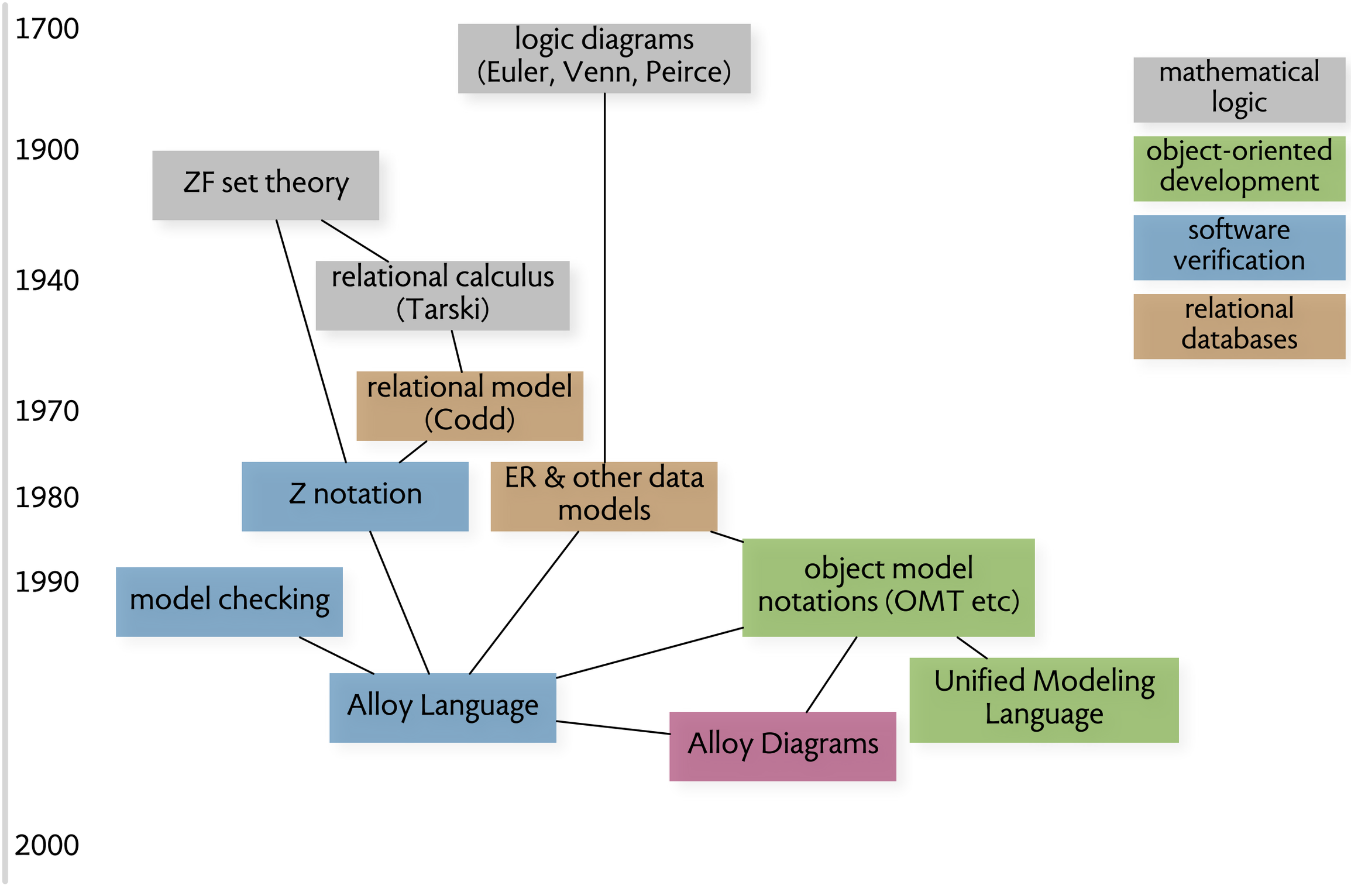
being explicit

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

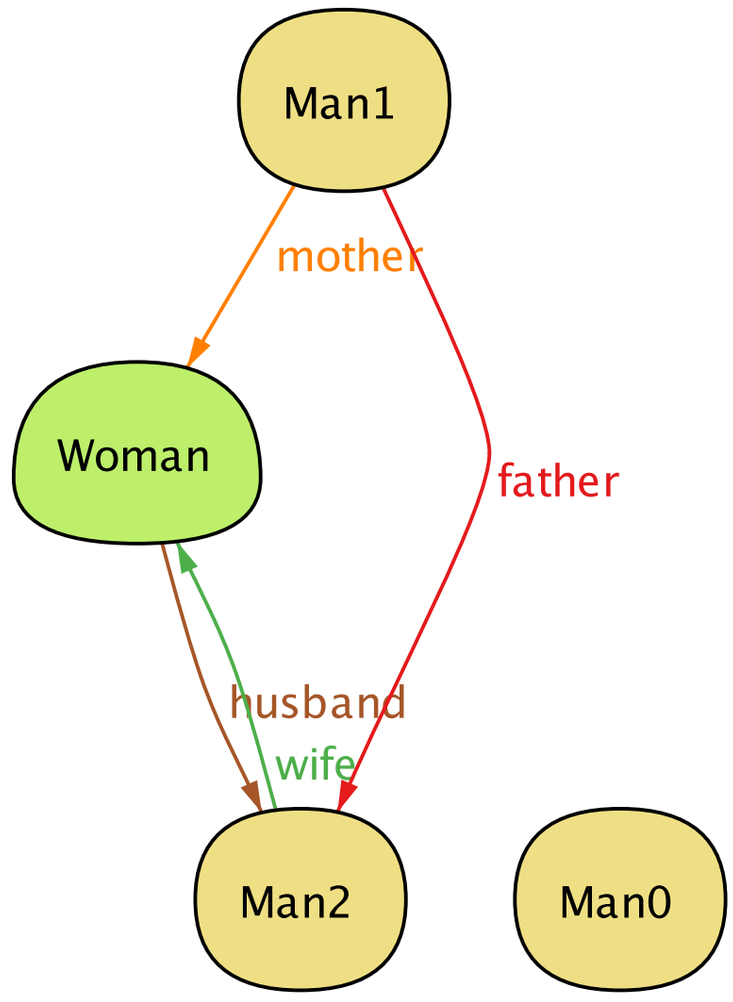


semantic concepts

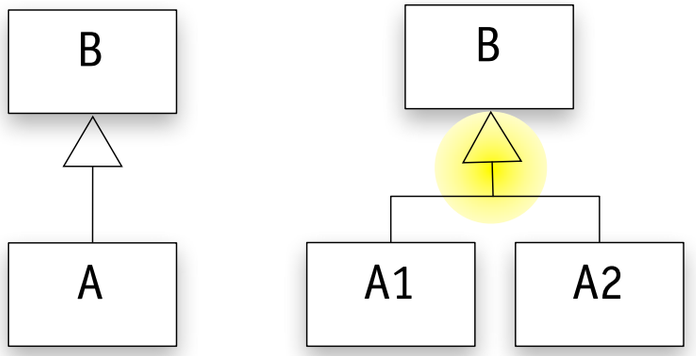
atom
indivisible
immutable
uninterpreted

relation
collection of atom tuples

set
collection of atoms
(ie, a unary relation)



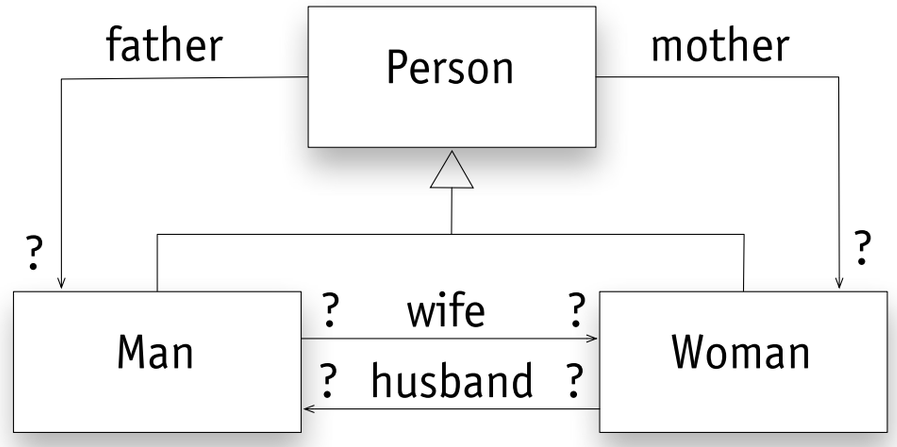
graphical syntax

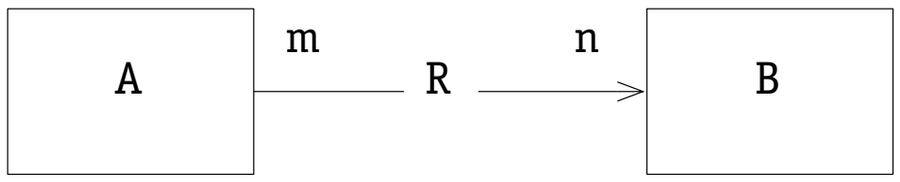


$A \subseteq B$

$A1 \subseteq B,\ A2 \subseteq B$

$A1 \cap A2 = \varnothing$

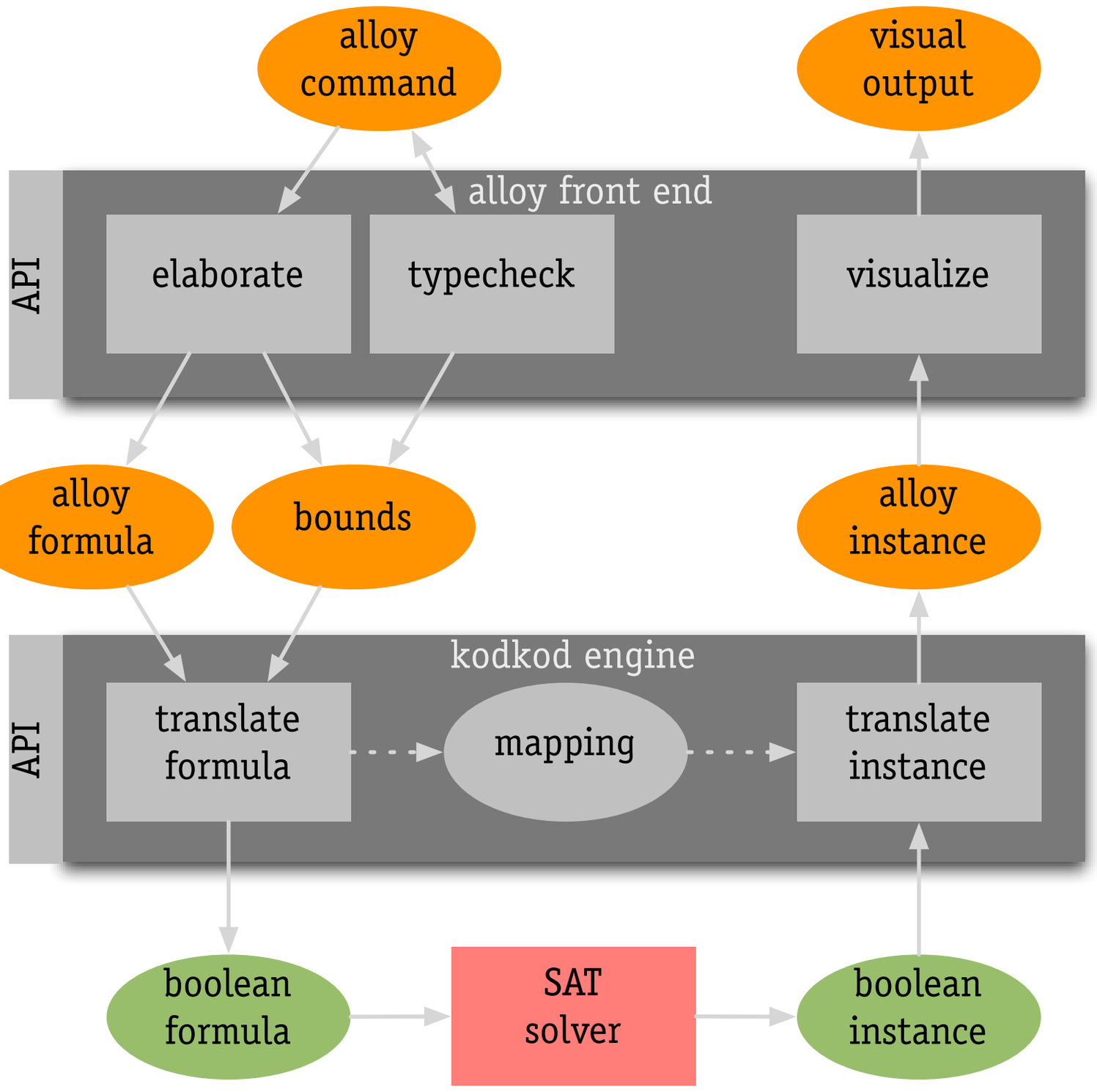




- R maps **m** A's to each B
- R maps each A to **n** B's

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

the alloy analyzer: a model finder



i'm my own grandpa

/Users/dnj/Filestore/Talks/google-12/models/grandpa.als

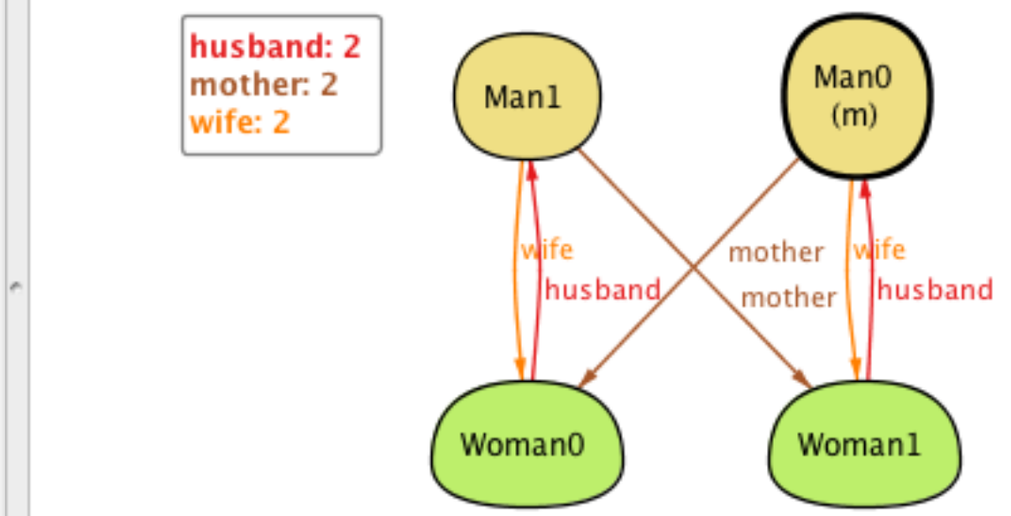
New Open Reload Save Execute Show

```
/*
 * 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 isomorophic 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
```

```
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.
```

(grandpa) Run ownGrandpa for 4 Person

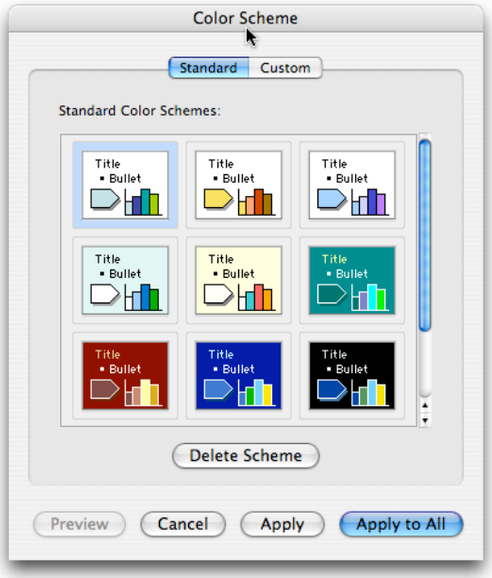
Viz Dot XML Tree Theme Magic Layout Evaluator Next Projection: none



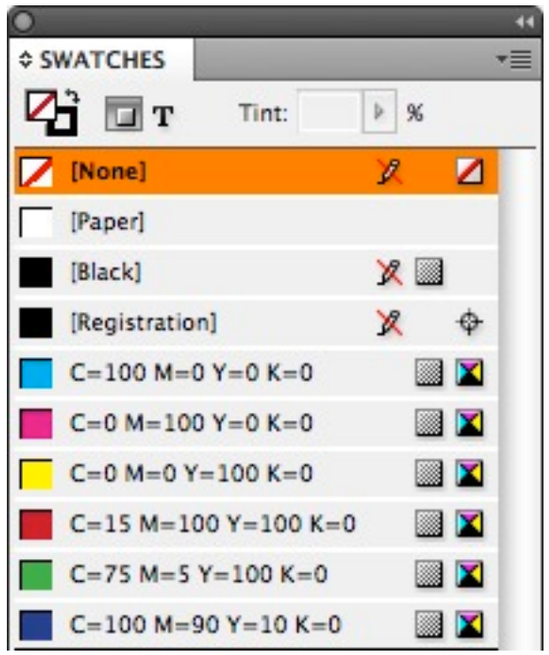
#4

some generic concepts

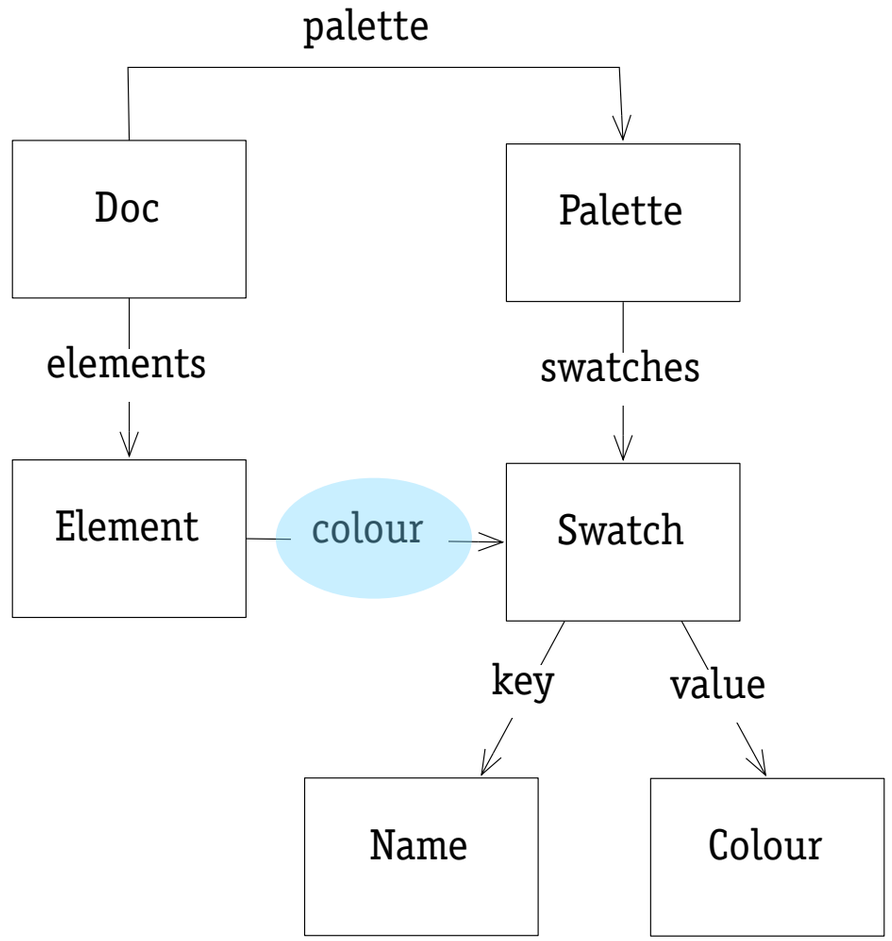
"style"



Powerpoint



Indesign

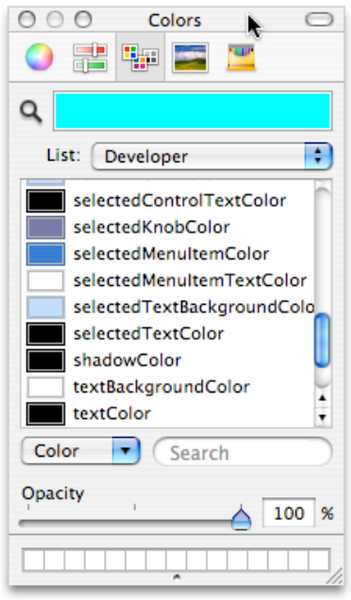


There is no problem in computer science that cannot be solved by introducing another level of indirection. --David Wheeler

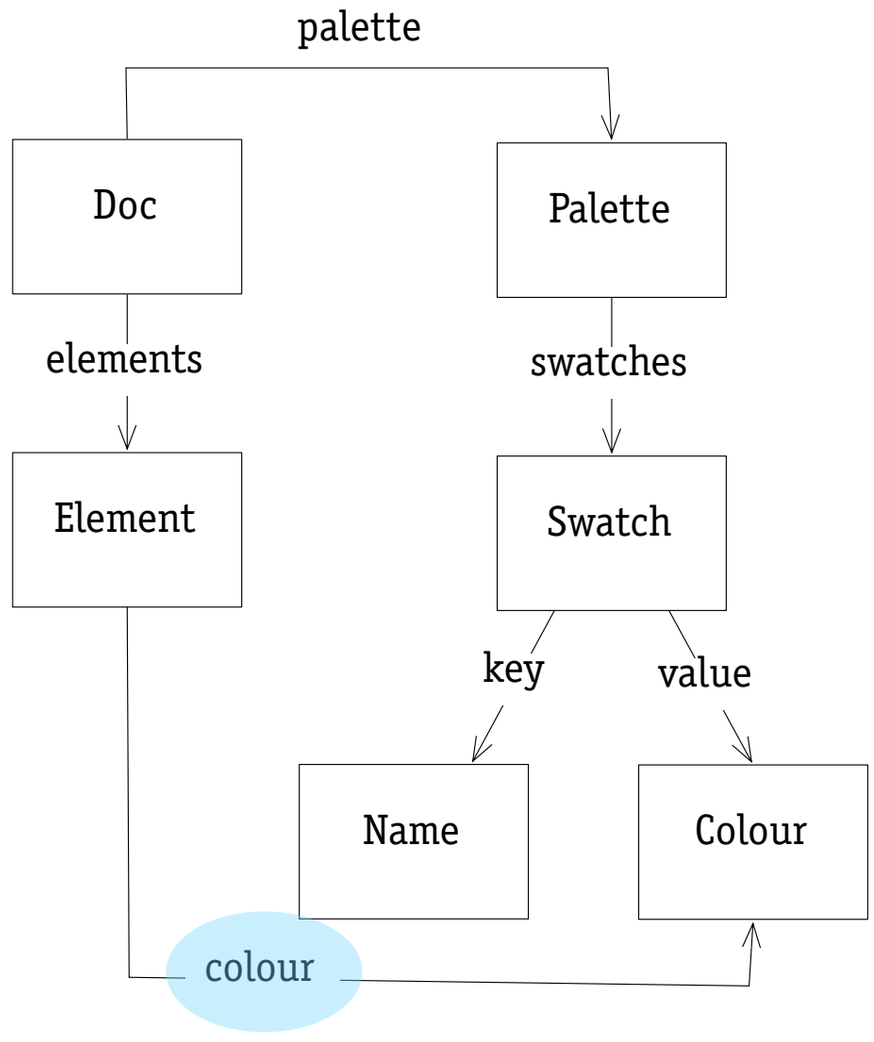
rough edges

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

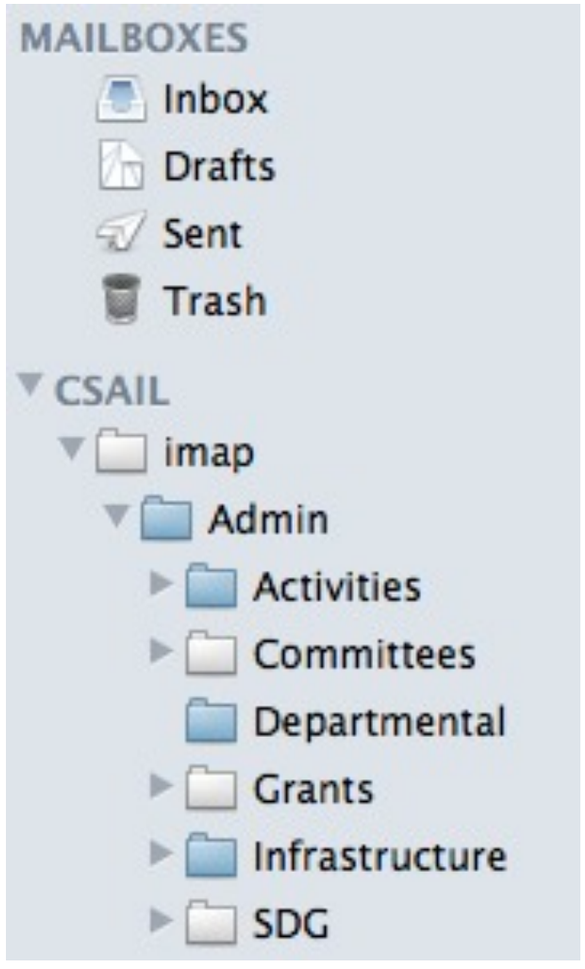
"pseudo style"

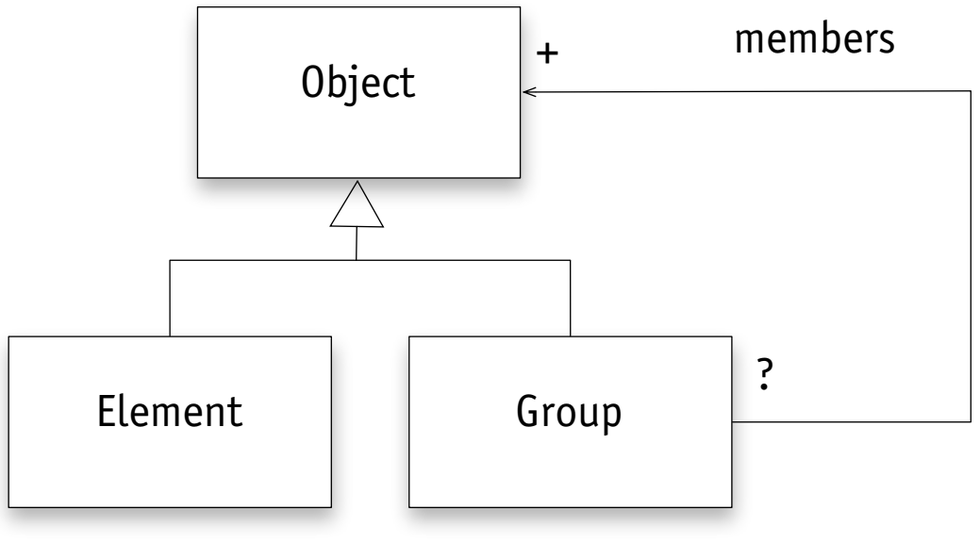


Keynote



"composite"

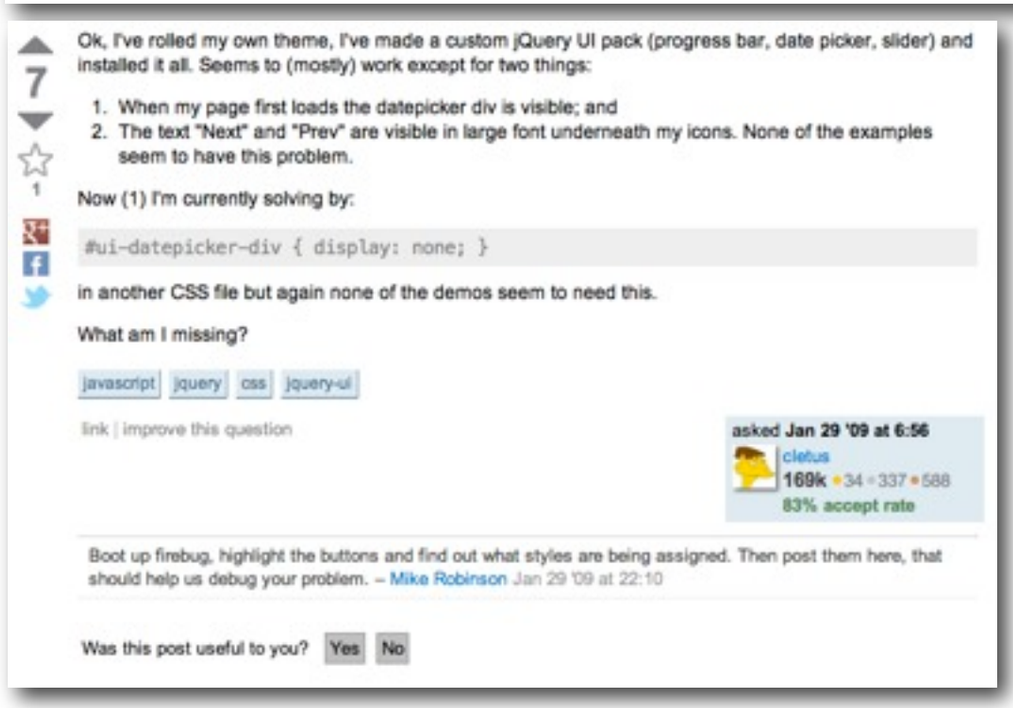


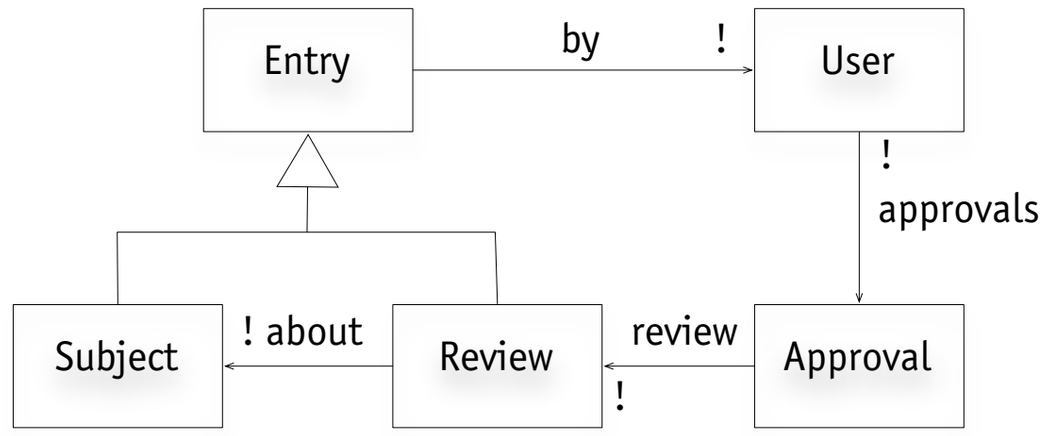


rough edges

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

“approval”





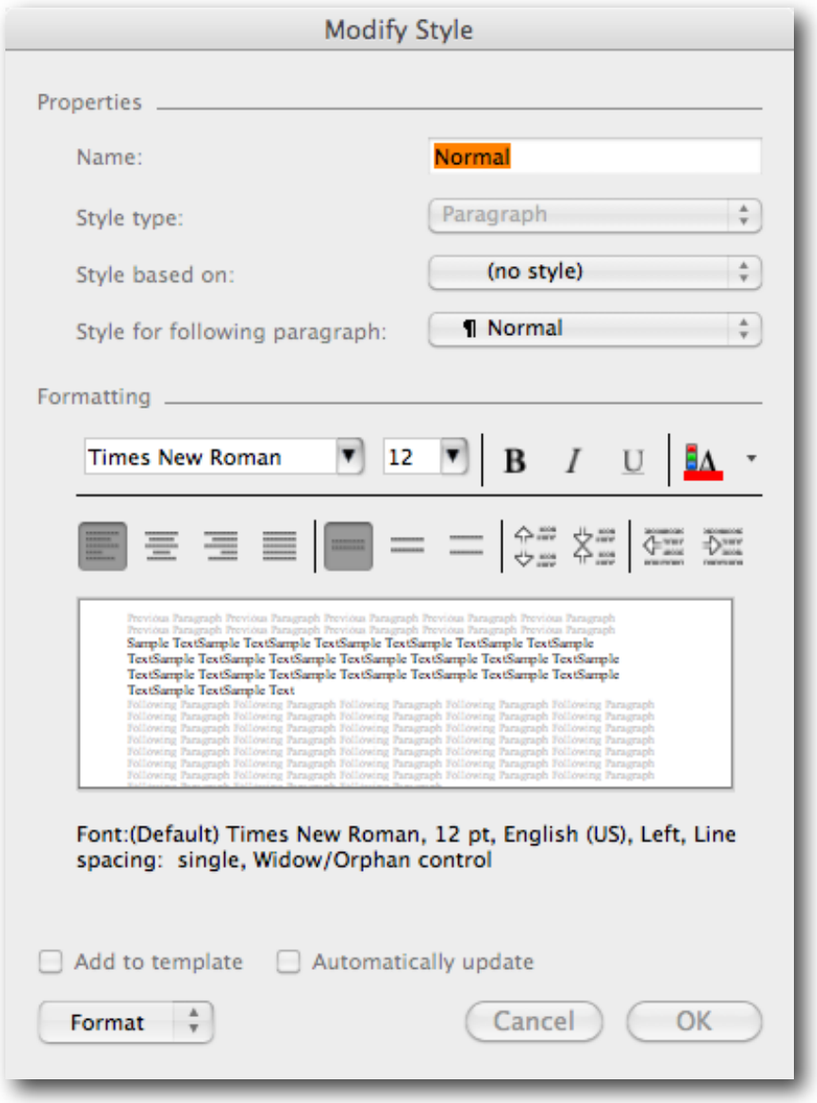
rough edges

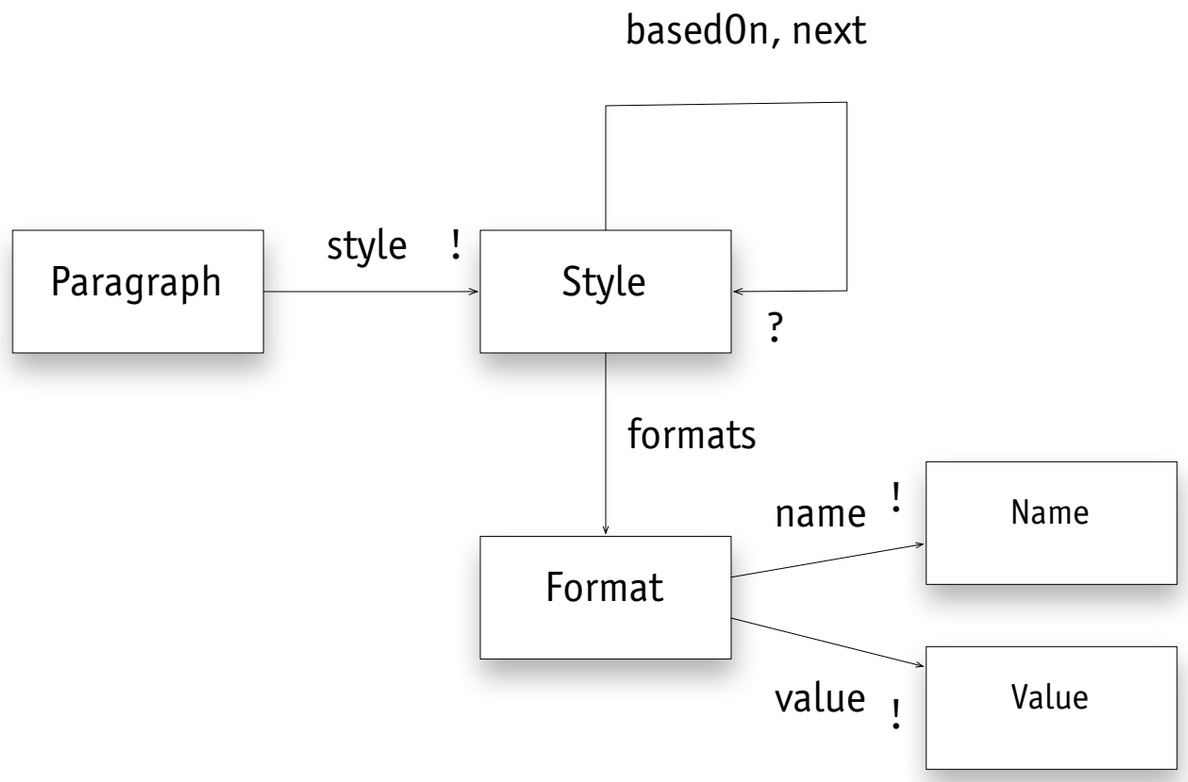
› your suggestions?

#5

three conceptual models

microsoft word





rough edges

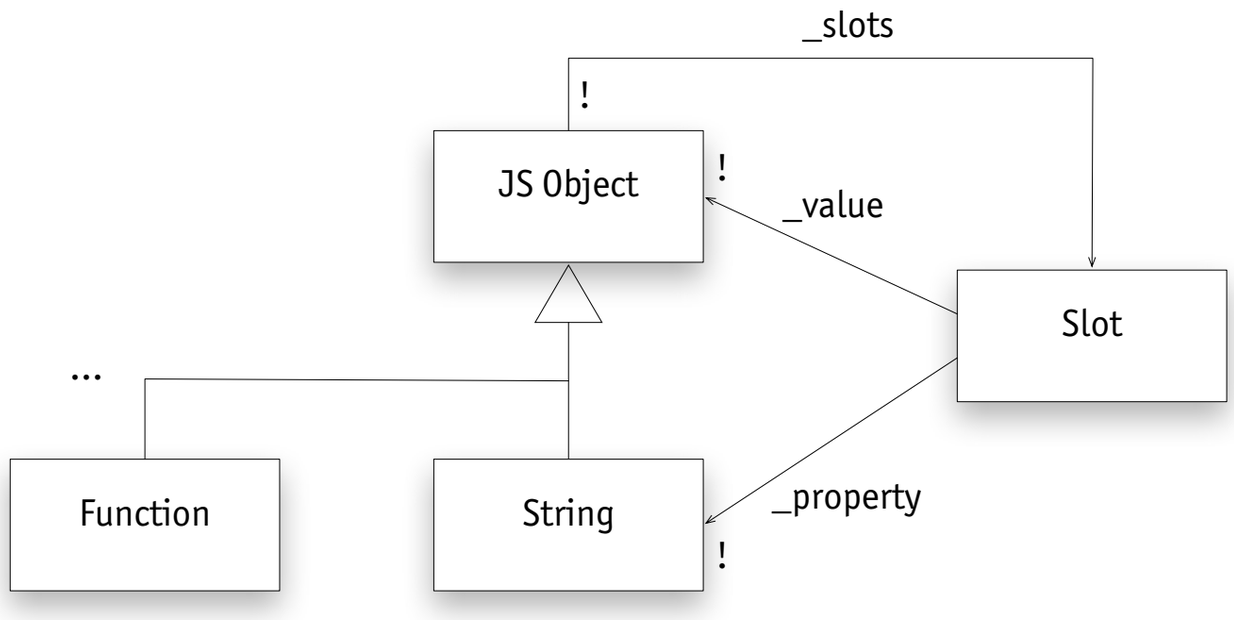
› special role of Normal style, etc

› 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];
}
```

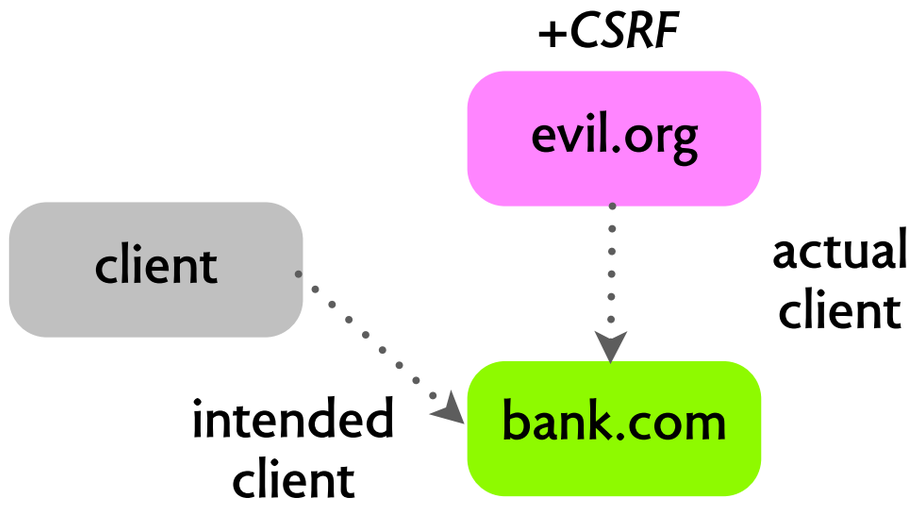
rough edges

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

origins & referrers

referrers & origins

```
<img src="http://mybank.com/transferFunds?
amount=1000&destination=attackersAcct"
width="0" height="0"/>
```



a strategy for XSS and CSRF

browser tracks "origin" of each request

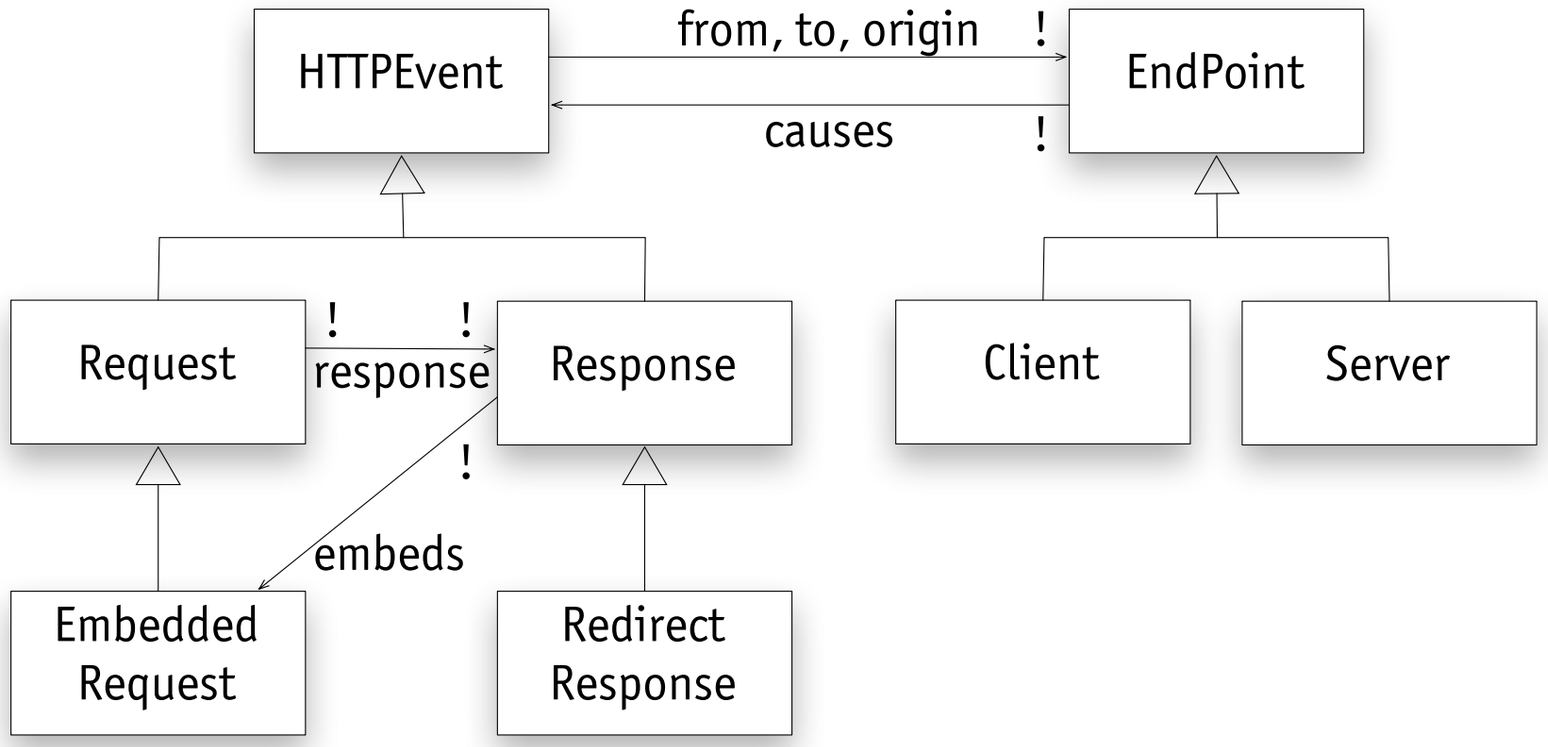
with HTTP request, includes origin as "referrer"

if referrer is not self, server rejects it

critical property

s=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

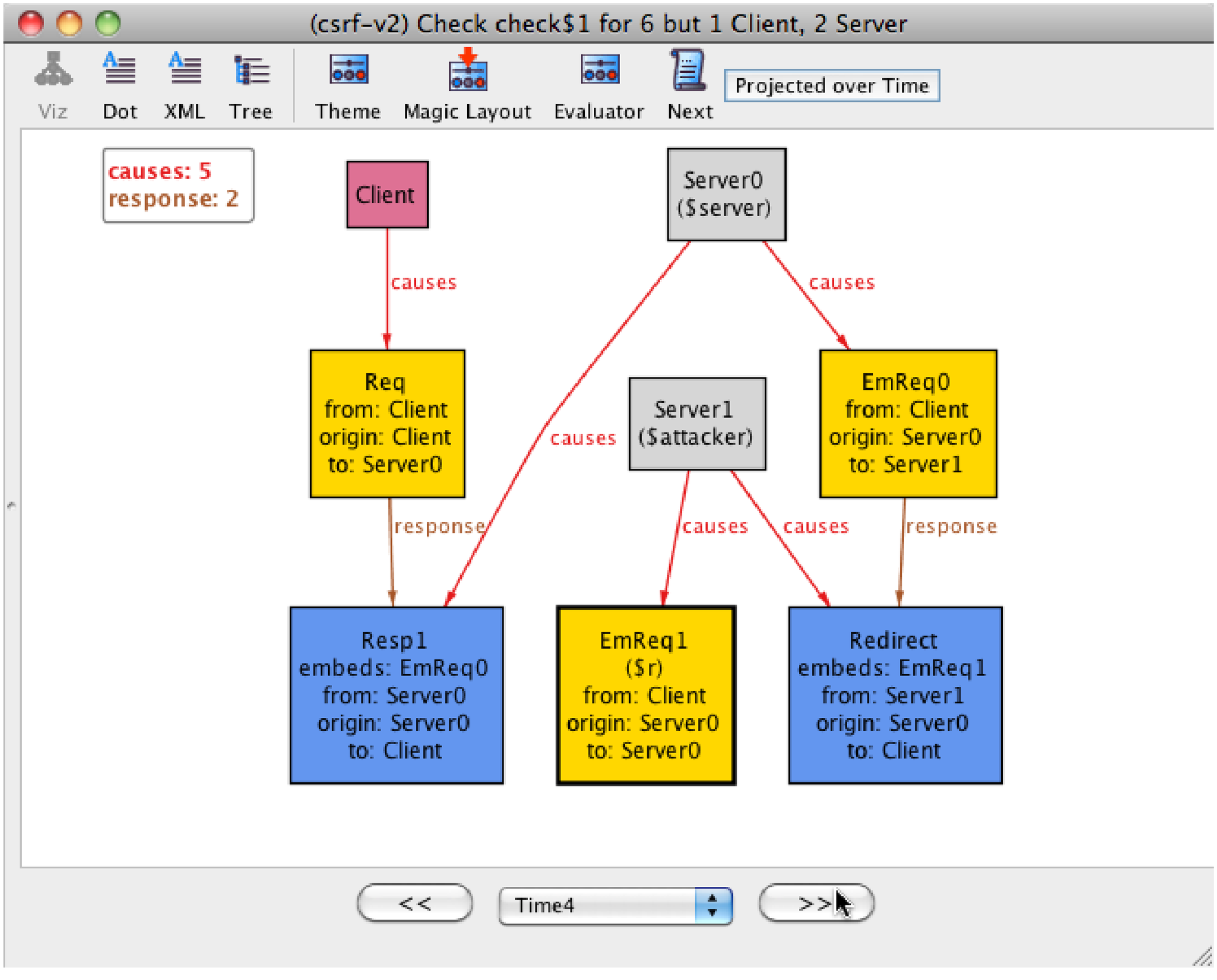
```
fact Origin {
    // for a redirect, origin is same as request, else server
    all r: Request | r.response.origin =
        (r.response in Redirect implies r.origin else r.response.from)

    // 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.

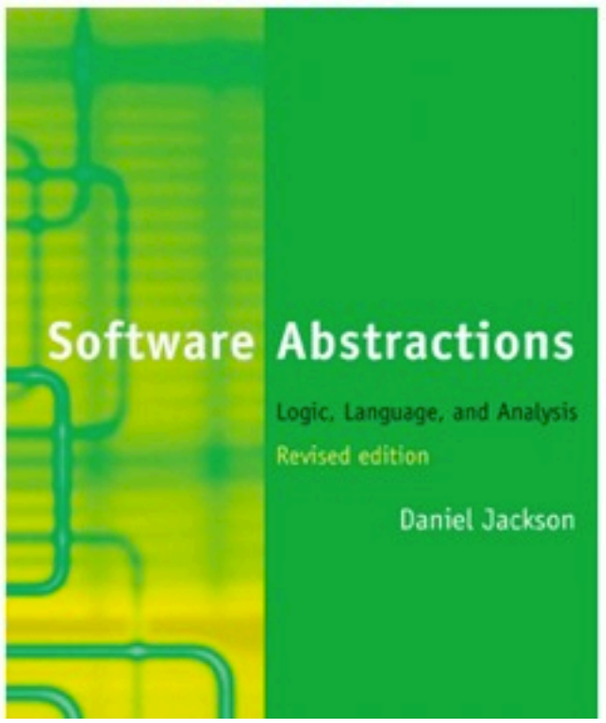
The Alloy Analyzer works by reduction to SAT. Version 4 was a complete rewrite that included Kodkod, a new model finding engine that optimizes the reduction, and a new front end.

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

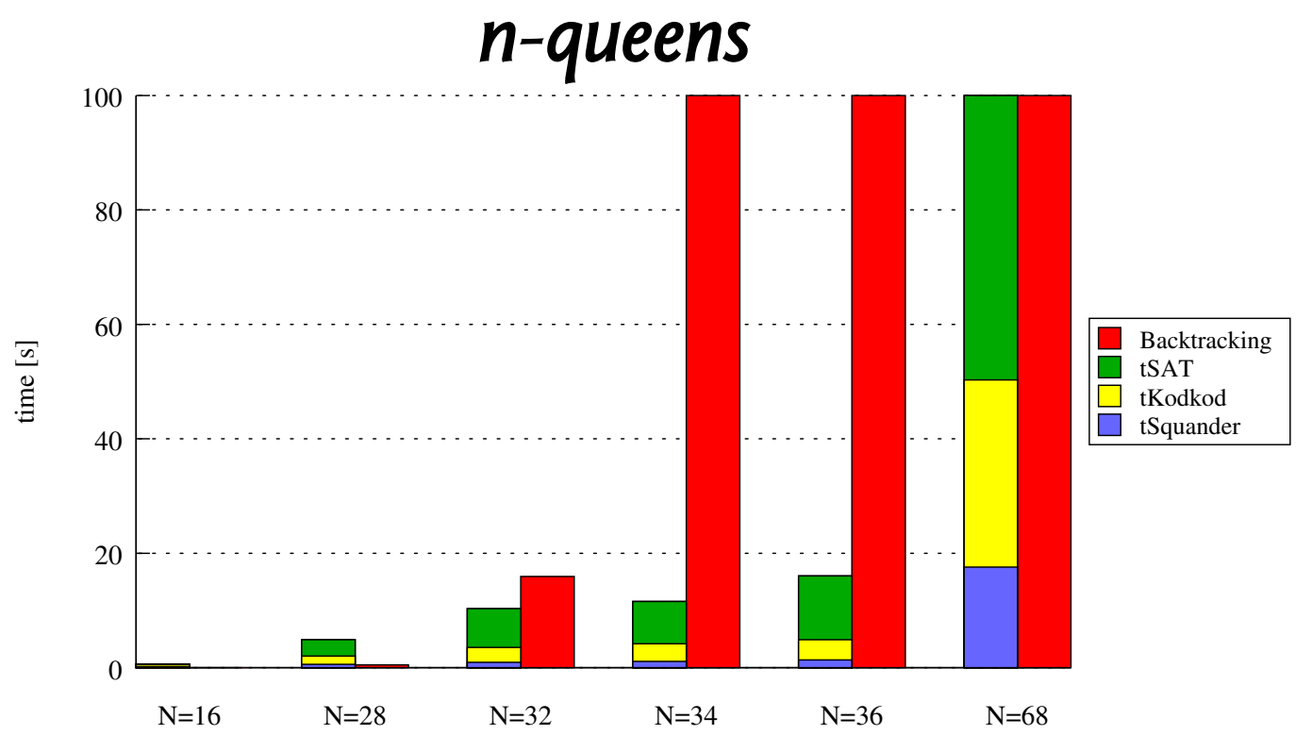
thank you!

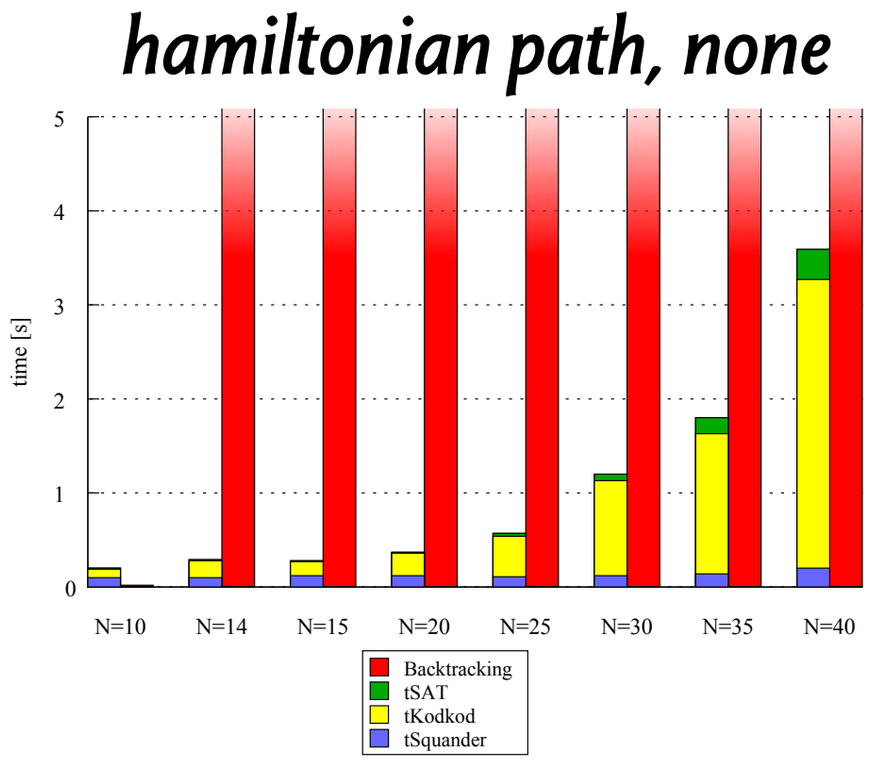
squander

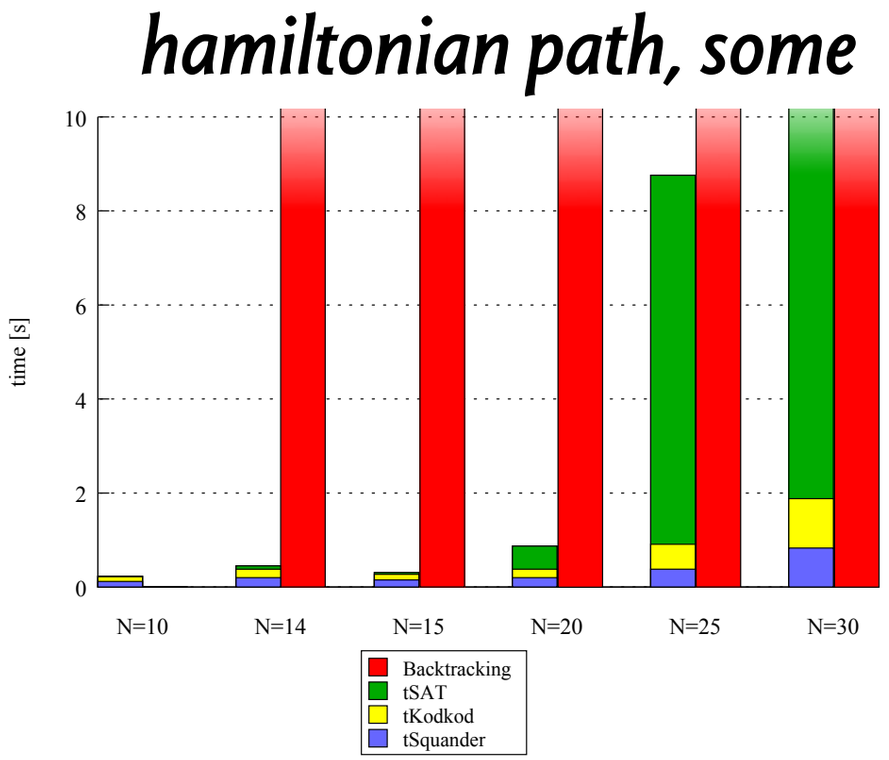
```
public class Sudoku {
  private int [][] grid = new int [9][9];

@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







Rubicon specs

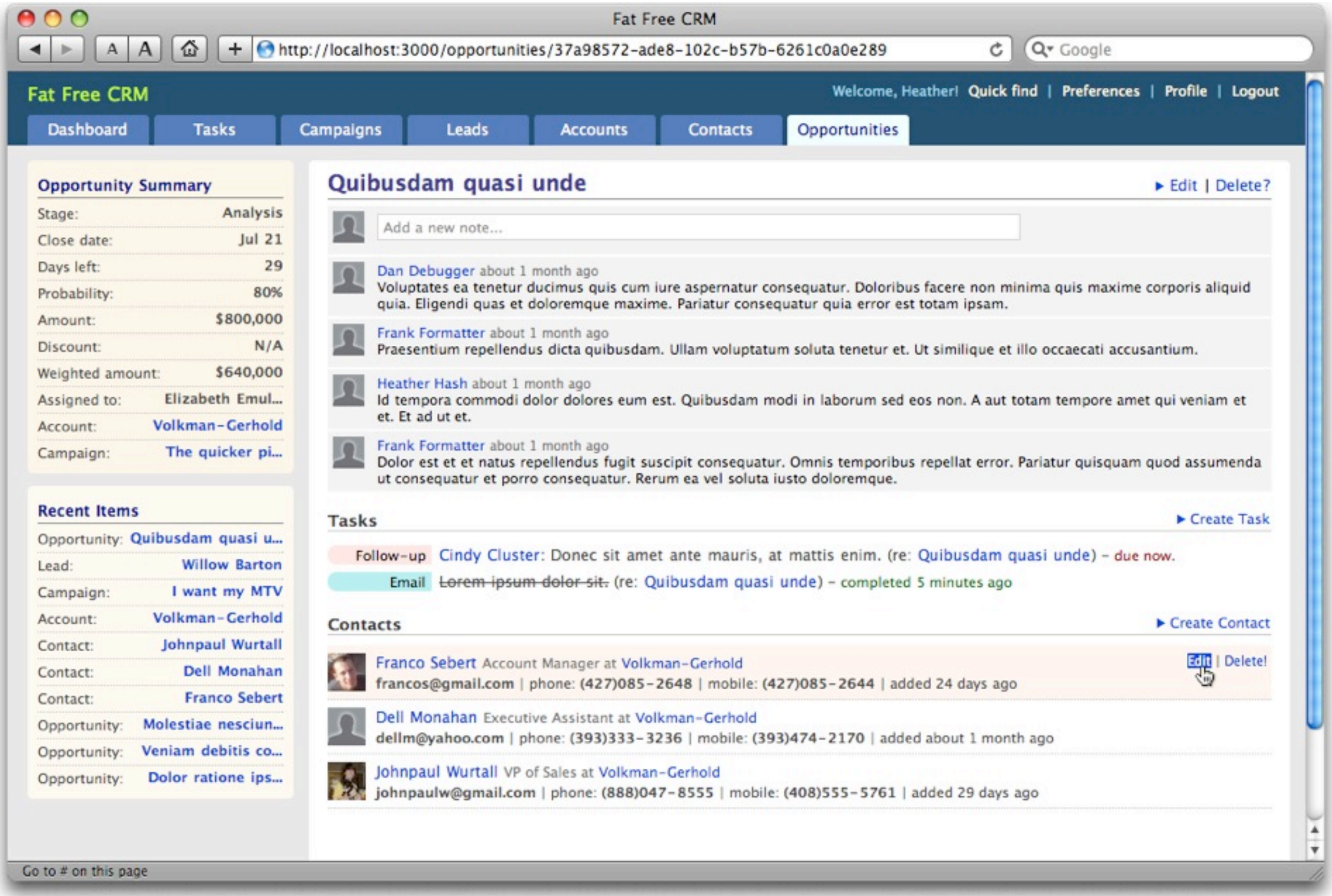
```
it "user included in list of users" do
  user = Factory(:user)
  get :index
  assigns[:users].should include user
end
```

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


Fat Free CRM
http://localhost:3000/opportunities/37a98572-ade8-102c-b57b-6261c0a0e289
Google
Fat Free CRM
Welcome, Heather! Quick find | Preferences | Profile | Logout
Dashboard
Tasks
Campaigns
Leads
Accounts
Contacts
Opportunities
Opportunity Summary
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...
Recent Items
Opportunity: Quibusdam quasi u...
Lead: Willow Barton
Campaign: I want my MTV
Account: Volkman-Gerhold
Contact: Johnpaul Wurtall
Contact: Dell Monahan
Contact: Franco Sebert
Opportunity: Molestiae nesciun...
Opportunity: Veniam debitis co...
Opportunity: Dolor ratione ips...
Quibusdam quasi unde
▸ Edit | Delete?
Add a new note...
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.
Tasks
▸ Create Task
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
Contacts
▸ Create Contact
Franco Sebert Account Manager at Volkman-Gerhold
francos@gmail.com | phone: (427)085-2648 | mobile: (427)085-2644 | added 24 days ago
Edit | Delete!
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
Go to # on this page

prototype Apache analyzer