design by concept

Daniel Jackson · CSAIL, MIT

9th Summer School on Formal Techniques · Menlo College, CA · May 19-24, 2019
lecture one
how this project started
a simple task: sign and return
Field Trip Permission Form

Dear Parents:

Ms. Frizzle will again be taking her second grade class on an exciting field trip. Please sign and return the permission slip below.

Thank you!

Yes, I give permission for my child to go on the second grade “Touch and Feel” trip on Friday February 13th to the NastyCo Nuclear Dump. I understood that my child may encounter the normal risks of childhood play, including grazed knees, hurt feelings and exposure to toxic waste.

________________________________________  
Parents signature  Date
Dear Parents:

Ms. Frizzle will again be taking her second grade class on an exciting field trip. Please sign and return the permission slip below.

Thank you!

Yes, I give permission for my child to go on the second grade “Touch and Feel” trip on Friday February 13th to the NastyCo Nuclear Dump. I understood that my child may encounter the normal risks of childhood play, including grazed knees, hurt feelings and exposure to toxic waste.

[Signature]

Parents signature  Date
Field Trip Permission Form

Dear Parents:

Ms. Frizzle will again be taking her second grade class on an exciting field trip. Please sign and return the permission slip below.

Thank you!

Yes, I give permission for my child to go on the second grade “Touch and Feel” trip on Friday February 13th to the NastyCo Nuclear Dump. I understood that my child may encounter the normal risks of childhood play, including grazed knees, hurt feelings and exposure to toxic waste.

November 7, 2008

Parents signature Date
acrobat to the rescue?
Ms. Frizzle will again be taking her second grade class on an exciting field trip. Please sign and return the permission slip below.

Thank you!

Yes, I give permission for my child to go on the second grade “Touch and Feel” trip on Friday February 13th to the NastyCo Nuclear Dump. I understood that my child may encounter the normal risks of childhood play, including grazed knees, hurt feelings and exposure to toxic waste.

Parents signature   Date
Ms. Frizzle will again be taking her second grade class on an exciting field trip. Please sign and return the permission slip below.

Thank you!

Yes, I give permission for my child to go on the second grade “Touch and Feel” trip on Friday February 13th to the NastyCo Nuclear Dump. I understood that my child may encounter the normal risks of childhood play, including grazed knees, hurt feelings and exposure to toxic waste.

Parents signature  Date
acrobat to the rescue?

Ms. Frizzle will again be taking her second grade class on an exciting field trip. Please sign and return the permission slip below.

Thank you!

how to add a signature in acrobat
  -- open document in acrobat
  -- Tools—>Advanced Editing—>Touchup Object Tool
  -- right click at desired point | Place Image...
then select jpg

how to add date
  -- Tools—>Typewriter
Ms. Frizzle will again be taking her exciting field trip. Please sign and return the permission slip below.

Thank you!

Yes, I give permission for my child to go on the second grade “Touch and Feel” trip on Friday February 13th to the NastyCo Nuclear Dump. I understood that my child may encounter the normal risks of childhood play, including grazed knees, hurt feelings and exposure to toxic waste.

Parents signature  Date
acrobat to the rescue?

Ms. Frizzle will again be taking her class on an exciting field trip. Please sign and return the permission slip below.

Thank you!

Yes, I give permission for my child to go on the second grade “Touch and Feel” trip on Friday February 13th to the NastyCo Nuclear Dump. I understood that my child may encounter the normal risks of childhood play, including grazed knees, hurt feelings and exposure to toxic waste.

Parents signature Date
acrobat to the rescue?

Ms. Frizzle will again be taking her fourth-grade class on an exciting field trip. Please sign and return the permission slip below.

Thank you!

Yes, I give permission for my child to go on the second grade “Touch and Feel” trip on Friday February 13th to the NastyCo Nuclear Dump. I understood that my child may encounter the normal risks of childhood play, including grazed knees, hurt feelings and exposure to toxic waste.

Parents Date

Select All Place Image... Find... Edit Page...
adobe photoshop: cropping surprises
adobe photoshop: cropping surprises
adobe photoshop: cropping surprises
adobe photoshop: cropping surprises

any idea what's going on here?
adobe lightroom: easy cropping
adobe lightroom: easy cropping
i’m not alone

from http://amplicate.com
i’m not alone

from http://amplicate.com
Adobe Acrobat is a family of computer programs developed by Adobe Systems, designed to view, create, manipulate and manage files.

47% Love Acrobat

207 Positive Opinions out of 444

53% Hate Acrobat

237 Negative Opinions out of 444

Adobe Photoshop Lightroom is a photography software program developed by Adobe Systems for Mac OS X and Microsoft Windows, designed...

89% Love Lightroom

2,335 Positive Opinions out of 2,632

11% Hate Lightroom

297 Negative Opinions out of 2,632

I'm not alone
i’m not alone

from http://amplicate.com
what’s the essence of the problem?

not the user interface!
polished and organized
what's the essence of the problem?

not the user interface!
opolished and organized

task-oriented design

no unifying concepts
what’s the essence of the problem?

- not the user interface!
- polished and organized

- task-oriented design
- no unifying concepts

- complex concepts
- cropping vs. resizing
what's the essence of the problem?

- not the user interface!
- polished and organized

- task-oriented design
- no unifying concepts

- complex concepts
- cropping vs. resizing
adobe fixes acrobat

Version 9 (2008)

task-oriented design
no unifying concepts
Adobe fixes Acrobat

Version 9 (2008)
- Task-oriented design
- No unifying concepts

Version 10 (2010)
- Improved interface
- But still no concepts
Adobe fixes Acrobat

Version 9 (2008)
- Task-oriented design
- No unifying concepts

Version 10 (2010)
- Improved interface
- But still no concepts

Version 11 (2012)
- Unifying concepts
- Text/image object
a research & teaching program

designing software with concepts

- a design theory
- design case studies
- design patterns
- code platform

**concept structure & design rules**
[Onward 15]

**Gitless**
[Perez De Rosso, Onward 13, OOPSLA 16]

**about 30 so far**

**Deja Vu**
[Perez De Rosso]
how bugs led us astray
the software problem

**need**
the motivation for building the system

**implementation**
the mechanisms of the system
separating concerns

- **need**: the motivation for building the system
- **specification**: the planned behavior of the system
- **implementation**: the mechanisms of the system
separating concerns

need
the motivation for building the system

specification
the planned behavior of the system

correctness

implementation
the mechanisms of the system
separating concerns

- **need**: the motivation for building the system
- **specification**: the planned behavior of the system
- **implementation**: the mechanisms of the system

Concepts:
- **pleasantness**
- **correctness**
separating concerns

need
the motivation for building the system

specification
the planned behavior of the system

implementation
the mechanisms of the system

what we devoted ourselves to

pleasantness

correctness
separating concerns

**need**
the motivation for building the system

**specification**
the planned behavior of the system

**implementation**
the mechanisms of the system

**pleasantness**

**correctness**

---

**what we devoted ourselves to**

**what mattered more?**
| me, Alyssa (12) | hacking | meetups | javascript - Hello again B€ | 11:48 am |
correct ⇒ useful?
correct ⇒ useful?

<table>
<thead>
<tr>
<th>Label</th>
<th>Subject</th>
<th>Message</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>me, Alyssa</td>
<td>hacking</td>
<td>Hello again Ben.</td>
<td>11:48 am</td>
</tr>
<tr>
<td>me, Alyssa</td>
<td>meetups</td>
<td>Hello again Ben.</td>
<td>9:43 am</td>
</tr>
<tr>
<td>me, Alyssa</td>
<td>hacking</td>
<td>Hello again Ben.</td>
<td>9:58 am</td>
</tr>
</tbody>
</table>
correct ⇒ useful?

<table>
<thead>
<tr>
<th>Label</th>
<th>Corresponding Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>label:hacking</td>
<td>me, Alyssa (12) hacking meetups javascript - Hello again Ben 11:48 am</td>
</tr>
<tr>
<td>label:meetups</td>
<td>me, Alyssa (12) Inbox hacking javascript - Hello again Ben 9:58 am</td>
</tr>
</tbody>
</table>

No messages matched your search. Try using search options such as sender, date, size and more.
correct ⇒ useful?

any gmail users who can explain?
correct $\Rightarrow$ safe?
correct $\Rightarrow$ safe ?

airborne $\Leftrightarrow$ disabled

requirement
correct \implies safe ?

\neg \text{WheelPulse} \iff \text{disabled}

\text{specification}

\text{airborne} \iff \text{disabled}

\text{requirement}
correct $\Rightarrow$ safe?

- airborne $\Leftrightarrow$ $\neg$WheelPulse
- $\neg$WheelPulse $\Leftrightarrow$ disabled
- airborne $\Leftrightarrow$ disabled

environment $\land$ specification $\Rightarrow$? requirement
From: "TIG" <help@MIT.EDU>
Date: October 13, 2008 11:04:08 AM EDT
To: "Daniel Jackson" <dnj@csail.mit.edu>
Subject: your password

We recently ran a password checker to evaluate passwords of all CSAIL users, and your password was readily broken. Please choose a new password ASAP…
We recently ran a password checker to evaluate passwords of all CSAIL users, and your password was readily broken. Please choose a new password ASAP…

my password:

sergeantpepper1967
From: "TIG" <help@MIT.EDU>
Date: October 13, 2008 11:04:08 AM EDT
To: "Daniel Jackson" <dnj@csail.mit.edu>
Subject: your password

We recently ran a password checker to evaluate passwords of all CSAIL users, and your password was readily broken. Please choose a new password ASAP…

my password:

sergeantpepper1967

8 char limit: passwd utility silently truncated rest
We recently ran a password checker to evaluate passwords of all CSAIL users, and your password was readily broken. Please choose a new password ASAP…

my password:

sergeantpepper1967

8 char limit: passwd utility silently truncated rest

Aydal [2009]
Analyzed Tokeneer for security
Found 9 anomalous scenarios
eg, new configuration file silently ignored if one exists on disk
what’s a concept?
apps characterized by their concepts
apps characterized by their concepts

Microsoft Word
Paragraph Format Style

Twitter

Photoshop
apps characterized by their concepts

Microsoft Word
Paragraph Format Style

Twitter
Tweet Hashtag Following

Photoshop
apps characterized by their concepts

Microsoft Word
- Paragraph Format Style

Twitter
- Tweet Hashtag Following

Photoshop
- PixelMap Layer/Mask Adjustment
What's the difference between a text editor and a word processor?
app classes characterized by concepts too
app classes characterized by concepts too
app classes characterized by concepts too

text editor

word processor
app classes characterized by concepts too

desktop publishing app

word processor

text editor

waffling about concepts and fine points of design
app classes characterized by concepts too

text editor
line buffer

word processor

desktop publishing app
app classes characterized by concepts too

text editor
line buffer

word processor
paragraph format style

desktop publishing app

waffling about concepts
and fine points of design
app classes characterized by concepts too

**text editor**
- line buffer

**word processor**
- paragraph
- format
- style

**desktop publishing app**
- stylesheet
- text flow
- page template

waffling about concepts and fine points of design
app classes characterized by concepts too

- text editor
  - line buffer

- word processor
  - paragraph
  - format
  - style

- desktop publishing app
  - stylesheet
  - text flow
  - page template

jamonh
Oct 22, 2013 7:19 PM

Just upgraded to the new Pages and can't find a way to link text boxes anymore like
http://www.macobserver.com/tmo/article/pages-linking-text-boxes

Am I missing something, or is it really not possible anymore?
A mental model captures ideas in a problem domain, while a conceptual model represents 'concepts' (entities) and relationships between them.

A conceptual model in the field of computer science is a special case of a general conceptual model. To distinguish from other types of models, it is also known as a domain model. Conceptual modeling should not be confused with other modeling disciplines such as data modelling, logical modelling and
concepts are invented, not just out there

Tim Mott visits Ginn in 1974 brings idea of styles to PARC

Charles Simonyi’s team implements style in Bravo text editor

Simonyi brings style to Microsoft in 1983
the rewards of inventing a good concept
the rewards of inventing a good concept
the rewards of inventing a good concept

who is this and what is he doing?
concepts have purpose

purpose of **style**: enable consistent formatting

Apple Keynote adds style concept (2017?)
concept structure is designed not discovered
There is no problem in computer science that cannot be solved by introducing another level of indirection.

*David Wheeler*
concepts are reusable

Powerpoint color schemes

Indesign swatches

Keynote image styles
not an instance of style
not an instance of style

what crucial action is missing?
how to explain the style concept?

If you assign Heading to two paragraphs and then you change the style from bold to italic, both paragraphs will be changed in concert.
explaining concepts

how to explain the style concept?

If you assign Heading to two paragraphs and then you change the style from bold to italic, both paragraphs will be changed in concert.

If you create a style and assign to two elements, then when you modify the style, both elements will change...
explaining concepts

how to explain the style concept?

If you assign Heading to two paragraphs and then you change the style from bold to italic, both paragraphs will be changed in concert.

If you create a style and assign to two elements, then when you modify the style, both elements will change...

*not ontological: “a style is a mapping...”*
explaining concepts

how to explain the style concept?

If you assign Heading to two paragraphs and then you change the style from bold to italic, both paragraphs will be changed in concert.

If you create a style and assign to two elements, then when you modify the style, both elements will change.

not ontological: “a style is a mapping…”

Johnson-Laird: constructive semantics
explaining concepts

how to explain the style concept?

If you assign Heading to two paragraphs and then you change the style from bold to italic, both paragraphs will be changed in concert.

If you create a style and assign to two elements, then when you modify the style, both elements will change...

not ontological: “a style is a mapping...”

Johnson-Laird: constructive semantics

Michael Polanyi: operational principle
explaining concepts

how to explain the style concept?

If you assign Heading to two paragraphs and then you change the style from bold to italic, both paragraphs will be changed in concert.

If you create a style and assign to two elements, then when you modify the style, both elements will change...

*not ontological: “a style is a mapping...”*

Johnson-Laird: constructive semantics

Michael Polanyi: operational principle

*if you have a full spec of the behavior of a concept, is the tactic redundant?*
explaining concepts

how to explain the style concept?

If you assign Heading to two paragraphs and then you change the style from bold to italic, both paragraphs will be changed in concert.

If you create a style and assign to two elements, then when you modify the style, both elements will change.

not ontological: “a style is a mapping...”

not redundant: unlike full spec, shows how concept meets purpose

if you have a full spec of the behavior of a concept, is the tactic redundant?

Johnson-Laird: constructive semantics

Michael Polanyi: operational principle
when concepts don’t fulfill purpose
when concepts don’t fulfill purpose

can be left blank
when concepts don’t fulfill purpose

can be left blank

cannot be blank
when concepts don’t fulfill purpose

Alexander’s *misfits*: not bugs but bad specs

**can be left blank**

**cannot be blank**
concepts: modules of behavior
concepts: modules of behavior
concepts: modules of behavior

inventive
concepts: modules of behavior

- inventive
- purposeful
concepts: modules of behavior

- inventive
- purposeful
- behavioral
concepts: modules of behavior

- inventive
- purposeful
- behavioral
- self-contained
concepts: modules of behavior

- inventive
- purposeful
- behavioral
- self-contained
- reusable
a reservation concept
a reservation concept

name  reservation
<table>
<thead>
<tr>
<th>name</th>
<th>reservation</th>
</tr>
</thead>
<tbody>
<tr>
<td>purpose</td>
<td>make access to shared resource reliable</td>
</tr>
</tbody>
</table>
a reservation concept

name: reservation

purpose: make access to shared resource reliable

structure:
slots: Owner -> Slot
holds: User -> Slot
a reservation concept

name
reservation

purpose
make access to shared resource reliable

structure
slots: Owner -> Slot
holds: User -> Slot

behavior
create (o: Owner, s: Slot)
  no slots.s => slots += o -> s
reserve (u: User, o: Owner, s: Slot)
  no holds.s and o -> s in slots => holds += u -> s
cancel (u: User, s: Slot)
  u -> s in holds => holds -= u -> s
use (u: User, o: Owner, s: Slot)
  u -> s in holds and o -> s in slots =>
a reservation concept

name: reservation

purpose: make access to shared resource reliable

structure:
- slots: Owner -> Slot
- holds: User -> Slot

behavior:
- create (o: Owner, s: Slot)
  - \( \text{no slots.s} \Rightarrow \text{slots} += o \rightarrow s \)
- reserve (u: User, o: Owner, s: Slot)
  - \( \text{no holds.s and o \rightarrow s in slots} \Rightarrow \text{holds} += u \rightarrow s \)
- cancel (u: User, s: Slot)
  - \( u \rightarrow s \text{ in holds} \Rightarrow \text{holds} -= u \rightarrow s \)
- use (u: User, o: Owner, s: Slot)
  - \( u \rightarrow s \text{ in holds and o \rightarrow s in slots} \Rightarrow \)

tactic:
- if create(o,s); reserve(u,o.s); ... no cancel(u,s) ... then can use(u,o,s)
a relational diagram

Owner ! slots → Slot

User ! holds → Taken
a relational diagram

- **Owner** points to **Slot** with an arrow labeled `! slots`.
- **User** points to **Taken** with an arrow labeled `! holds`.

A box represents a set of atomic things.
a relational diagram

- Box represents set of atomic things.
- Arrow represents binary relation.

1. **Owner** ! slots ! **Slot**
2. **User** ! holds ! **Taken**
a relational diagram

- **Owner**
  - arrow represents binary relation
  - ! slots

- **Slot**
  - Taken is a subset of Slot

- **User**
  - ! holds

- **Taken**

Box represents set of atomic things.
Box represents set of atomic things.

Owner holds slots.

Slot is a subset of Slot.

User holds Taken.

Multiplicity: each Taken has one user that holds it.
alloy expressions in one slide
alloy expressions in one slide
alloy expressions in one slide

u: User
s: Slot
holds: User -> Slot
Taken: set Slot
u: User
s: Slot
holds: User -> Slot
Taken: set Slot

**a relation is a table of rows**

holds = {(u1,s1), (u1,s2)}
holds' = {(u1,s1), (u1,s2), (u2,s3)}
u: User
s: Slot
holds: User -> Slot
Taken: set Slot

**a relation is a table of rows**
holds = {(u1,s1), (u1,s2)}
holds’ = {(u1,s1), (u1,s2), (u2,s3)}

**a set is a relation with one column**
Slot = {(s1), (s2), (s3), (s4)}
Taken = {(s1), (s2), (s3)}
u: User
s: Slot
holds: User -> Slot
Taken: set Slot

**a relation is a table of rows**

holds = {(u1,s1), (u1,s2)}
holds’ = {(u1,s1), (u1,s2), (u2,s3)}

**a set is a relation with one column**

Slot = {(s1), (s2), (s3), (s4)}
Taken = {(s1), (s2), (s3)}

**a scalar is a set with one row**

u = {(u2)}
s = {(s3)}
**alloy expressions in one slide**

```
set operators
+ union, - difference, & intersection, in subset
```

```
u: User
s: Slot
holds: User -> Slot
Taken: set Slot

a relation is a table of rows
holds = {(u1,s1), (u1,s2)}
holds' = {(u1,s1), (u1,s2), (u2,s3)}

a set is a relation with one column
Slot = {(s1), (s2), (s3), (s4)}
Taken = {(s1), (s2), (s3)}

a scalar is a set with one row
u = {(u2)}
s = {(s3)}
```
alloy expressions in one slide

u: User
s: Slot
holds: User -> Slot
Taken: set Slot

**a relation is a table of rows**
holds = {(u1,s1), (u1,s2)}
holds’ = {(u1,s1), (u1,s2), (u2,s3)}

**a set is a relation with one column**
Slot = {(s1), (s2), (s3), (s4)}
Taken = {(s1), (s2), (s3)}

**a scalar is a set with one row**
u = {(u2)}
s = {(s3)}

**set operators**
+ union, - difference, & intersection, in subset
Slot - Taken = {(s4)}
holds’ - holds = {(u2,s3)}
### Alloy Expressions in One Slide

**Set Operators**
- `+` union
- `-` difference
- `&` intersection
- `in` subset

**Relation Operators**
- `->` product
- `.` join

---

**Example**

- **u**: User
- **s**: Slot
- **holds**: User -> Slot
- **Taken**: set Slot

**A relation is a table of rows**
- `holds = {((u1,s1), (u1,s2))}`
- `holds' = {((u1,s1), (u1,s2), (u2,s3))}`

**A set is a relation with one column**
- **Slot = {s1, s2, s3, s4}**
- **Taken = {s1, s2, s3}**

**A scalar is a set with one row**
- **u = {(u2)}**
- **s = {(s3)}**
u: User
s: Slot
holds: User -> Slot
Taken: set Slot

**a relation is a table of rows**
holds = {((u1,s1), (u1,s2))}
holds' = {((u1,s1), (u1,s2), (u2,s3))}

**a set is a relation with one column**
Slot = {((s1), (s2), (s3), (s4))}
Taken = {((s1), (s2), (s3))}

**a scalar is a set with one row**
u = {((u2))}
s = {((s3))}

**set operators**
+ union, - difference, & intersection, in subset
Slot - Taken = {((s4))}
holds' - holds = {((u2,s3))}

**relation operators**
-> product
. join

a -> b = { (a0,..., an, b0,..., bm) | (a0,..., an) ∈ a ∧ (b0,..., bm) ∈ b }
set operators
+ union, - difference, & intersection, in subset

Slot - Taken = \{(s4)\}
holds' - holds = \{(u2,s3)\}

relation operators
-> product
  . join

product examples
u -> s = \{(u2,s3)\}
u -> Taken = \{(u2,s1), (u2,s2), (u2,s3)\}

a relation is a table of rows
holds = \{((u1,s1), (u1,s2))\}
holds' = \{((u1,s1), (u1,s2), (u2,s3))\}

a set is a relation with one column
Slot = \{((s1), (s2), (s3), (s4))\}
Taken = \{((s1), (s2), (s3))\}

a scalar is a set with one row
u = \{((u2))\}
s = \{((s3))\}

a -> b = \{(a_0,\ldots,a_n, b_0,\ldots,b_m) \mid (a_0,\ldots,a_n) \in a \land (b_0,\ldots,b_m) \in b\}
set operators
+ union, - difference, & intersection, in subset
Slot - Taken = \{(s4)\}
holds' - holds = \{(u2,s3)\}

relation operators
-> product
join

product examples
u -> s = \{(u2,s3)\}
u -> Taken = \{(u2,s1), (u2,s2), (u2,s3)\}
a \rightarrow b = \{(a_0,\ldots, a_n, b_0,\ldots, b_m) \mid (a_0,\ldots, a_n) \in a \land (b_0,\ldots, b_m) \in b\}
a.b = \{(a_0,\ldots, a_{n-1}, b_1,\ldots, b_m) \mid (a_0,\ldots, a_n) \in a \land (a_n, b_1,\ldots, b_m) \in b\}
**set operators**
+ union, - difference, & intersection, in subset

**relation operators**
-> product
* join

**product examples**

a set is a relation with one column

<table>
<thead>
<tr>
<th>Slot = {(s1), (s2), (s3), (s4)}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taken = {(s1), (s2), (s3)}</td>
</tr>
</tbody>
</table>

a scalar is a set with one row

u = {(u2)}
s = {(s3)}

a relation is a table of rows

| holds = {{(u1,s1), (u1,s2)}} |
| holds' = {{(u1,s1), (u1,s2), (u2,s3)}} |

**a relation is a table of rows**

| holds = {{(u1,s1), (u1,s2)}}, holds' = {{(u1,s1), (u1,s2), (u2,s3)}} |
| Slot - Taken = {{(s4)}} |
| holds' - holds = {{(u2,s3)}} |

**relation operators**

- -> product
  - join

**product examples**

| u -> s = {{(u2,s3)}} |
| u -> Taken = {{(u2,s1), (u2,s2), (u2,s3)}} |

**join examples**

| u.holds' = {(s3)} |
| holds'.s = {{(u2)}} |
| holds.Slot = {{(u1)}} |

a -> b = \{(a_0, \ldots, a_n, b_0, \ldots, b_m) \mid (a_0, \ldots, a_n) \in a \land (b_0, \ldots, b_m) \in b\} 

a.b = \{(a_0, \ldots, a_{n-1}, b_1, \ldots, b_m) \mid (a_0, \ldots, a_n) \in a \land (a_n, b_1, \ldots, b_m) \in b\}
**alloy expressions in one slide**

**set operators**
+ union, - difference, & intersection, in subset

Slot - Taken = \{(s4)\}
holds' - holds = \{(u2,s3)\}

**relation operators**
-> product
• join

**product examples**
u -> s = \{(u2,s3)\}
u -> Taken = \{(u2,s1), (u2,s2), (u2,s3)\}

**join examples**
u.holds' = \{(s3)\}
holds'.s = \{(u2)\}
holds.Slot = \{(u1)\}

**formula examples**
holds' = holds + u -> s
(also written holds += u -> s
User.holds = Taken
holds in User -> Slot

\[ a \rightarrow b = \{ (a_0,\ldots, a_n, b_0,\ldots, b_m) \mid (a_0,\ldots, a_n) \in a \land (b_0,\ldots, b_m) \in b \} \]
\[ a.b = \{ (a_0,\ldots, a_{n-1}, b_1,\ldots, b_m) \mid (a_0,\ldots, a_n) \in a \land (a_n, b_1,\ldots, b_m) \in b \} \]

**u: User**

**s: Slot**

**holds: User -> Slot**

**Taken: set Slot**

**a relation is a table of rows**
holds = \{\{(u1,s1), (u1,s2)\}\}
holds' = \{\{(u1,s1), (u1,s2), (u2,s3)\}\}

**a set is a relation with one column**
Slot = \{(s1), (s2), (s3), (s4)\}
Taken = \{(s1), (s2), (s3)\}

**a scalar is a set with one row**
u = \{\{(u2)\}\}
s = \{\{(s3)\}\}

**a relation is a table of rows**
holds = \{\{(u1,s1), (u1,s2)\}\}
holds' = \{\{(u1,s1), (u1,s2), (u2,s3)\}\}

**a set is a relation with one column**
Slot = \{(s1), (s2), (s3), (s4)\}
Taken = \{(s1), (s2), (s3)\}

**a scalar is a set with one row**
u = \{\{(u2)\}\}
s = \{\{(s3)\}\}

**relation operators**
-> product
• join

**product examples**
u -> s = \{\{(u2,s3)\}\}
u -> Taken = \{\{(u2,s1), (u2,s2), (u2,s3)\}\}

**join examples**
u.holds' = \{\{(s3)\}\}
holds'.s = \{\{(u2)\}\}
holds.Slot = \{\{(u1)\}\}

**formula examples**
holds' = holds + u -> s
(also written holds += u -> s
User.holds = Taken
holds in User -> Slot

\[ a \rightarrow b = \{ (a_0,\ldots, a_n, b_0,\ldots, b_m) \mid (a_0,\ldots, a_n) \in a \land (b_0,\ldots, b_m) \in b \} \]
\[ a.b = \{ (a_0,\ldots, a_{n-1}, b_1,\ldots, b_m) \mid (a_0,\ldots, a_n) \in a \land (a_n, b_1,\ldots, b_m) \in b \} \]
a reservation concept

name: reservation

purpose: make access to shared resource reliable

structure:
- slots: Owner -> Slot
- holds: User -> Slot

behavior:
- create (o: Owner, s: Slot):
  - no slots.s => slots += o -> s
- reserve (u: User, o: Owner, s: Slot):
  - no holds.s and o -> s in slots => holds += u -> s
- cancel (u: User, s: Slot):
  - u -> s in holds => holds -= u -> s
- use (u: User, o: Owner, s: Slot):
  - u -> s in holds and o -> s in slots =>

tactic:
if create(o,s); reserve(u,o,s); ... no cancel(u,s) ... then can use(u,o,s)
checking a tactic with electrum

```plaintext
sig Slot {}
sig Owner {var slots: set Slot}
sig User {var holds: set Slot}

pred create [o: Owner, s: Slot] {
  no slots.s
  slots' = slots + o -> s
  holds' = holds
}

pred reserve [u: User, o: Owner, s: Slot] {
  no holds.s
  o -> s in slots
  holds' = holds + u -> s
  slots' = slots
}

pred cancel [u: User, s: Slot] {
  u -> s in holds
  holds' = holds - u -> s
  slots' = slots
}

pred can_use [u: User, o: Owner, s: Slot] {
  u -> s in holds and o -> s in slots
}

pred skip {slots' = slots and holds' = holds}
```

see: https://github.com/haslab/Electrum
checking a tactic with electrum

fact {
    no holds and no slots -- initially
    always (skip or some u: User, s: Slot, o: Owner |
        create[o,s] or reserve [u,o,s] or cancel[u,s])
}

check {
    -- can always use after reserve: not true
    all u: User, s: Slot, o: Owner |
    always (create[o,s] and after reserve[u,o,s] |
            implies after after always can_use[u,o,s])
}

see: https://github.com/haslab/Electrum
checking a tactic with electrum

fact {
  no holds and no slots -- initially
  always (skip or some u: User, s: Slot, o: Owner |
    create[o,s] or reserve [u,o,s] or cancel[u,s])
}

check {
  -- can always use after reserve: not true
  all u: User, s: Slot, o: Owner |
    always (create[o,s] and after reserve[u,o,s]
      implies after after always can_use[u,o,s])
}

see: https://github.com/haslab/Electrum
Design by concept is a new approach to creating software. A software product—whether an app, a service or a system—is viewed as a collection of interacting concepts, each with its own purpose, structure and behavior. Concepts can be invented afresh, but they can also be reused, exploiting the knowledge embodied in previous successful designs.

This book explains what concepts are and why they are central to software design; shows examples of concepts (from the most effective and ingenious to the most flawed and frustrating) taken from well-known applications; and presents design principles that can identify and eliminate flaws in existing and new designs.

Daniel Jackson is Professor of Computer Science, a MacVicar fellow, and Associate Director of the Computer Science and Artificial Intelligence Lab at MIT. His past research focused on software modeling and analysis; he is the creator of the Alloy language, and author of Software Abstractions: Logic, Language, and Analysis (MIT Press; second ed. 2012). His current interests include software design for improved usability, security and safety, and new programming paradigms. He was a recipient of the 2016 ACM SIGSOFT Impact Award, the 2017 ACM SIGSOFT Outstanding Research Award, and is an ACM Fellow.
Portraits of Resilience

Daniel Jackson

foreword by David A. Karp

http://portraitsofresilience.com
studio 1
identifying concepts: resy

here's a typical reservation app. what concepts can you identify?
identifying concepts: stack exchange

another example: a typical Q&A app
gmail
surprises
organizing messages
organizing messages

0 GB (0%) of 15 GB used
Manage

Terms - Privacy

Last account activity: 14 hours ago
Details

Alyssa P. Hacker - javascript - Reminds you of the old days, eh? 9:14 pm
organizing messages
automating filtering
automating filtering
automating filtering

Google

Gmail

Inbox (2)
Starred
Sent Mail
Drafts
Trash

Categories
Social
Promotions
Updates (1)
Forums

hacking (1)
meetups
todo

Primary
Social
Promotions

Alyssa P. Hacker

buy this! - My new JS book is out!
May 6

0 GB (0%) of 15 GB used
Manage

Terms - Privacy

Last account activity: 14 hours ago
Details
slightly surprising behavior #1
slightly surprising behavior #1
slightly surprising behavior #1

javascript

Alyssa P. Hacker
Reminds you of the old days, eh?

Ben Bitdiddle <benito.bitdiddle@gmail.com>
Yes, it does.

Click here to Reply or Forward
slightly surprising behavior #2

me, Alyssa (12) - hacking meetups javascript - Hello again Bé - 11:48 am
<table>
<thead>
<tr>
<th>Starred</th>
<th>Name</th>
<th>Label</th>
<th>Message Content</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>me, Alyssa (12)</td>
<td>hacking</td>
<td>javascript - Hello again Ben</td>
<td>11:48 am</td>
</tr>
<tr>
<td></td>
<td>me, Alyssa (12)</td>
<td>Inbox</td>
<td>javascript - Hello again Ben</td>
<td>9:43 am</td>
</tr>
</tbody>
</table>
slightly surprising behavior #2

<table>
<thead>
<tr>
<th>Email ID</th>
<th>Subject</th>
<th>Sender</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Image -104x-268 to 924x836]</td>
<td>hacking, meetups, javascript - Hello again Ben.</td>
<td>me, Alyssa (12)</td>
<td>9:43 am</td>
</tr>
<tr>
<td>[Image -104x-268 to 924x836]</td>
<td>hacking, meetups, javascript - Hello again Ben.</td>
<td>me, Alyssa (12)</td>
<td>9:58 am</td>
</tr>
</tbody>
</table>
slightly surprising behavior #2

No messages matched your search. Try using search options such as sender, date, size and more.
slightly surprising behavior #3
slightly surprising behavior #3

<table>
<thead>
<tr>
<th>Primary</th>
<th>Social</th>
<th>Promotions</th>
</tr>
</thead>
<tbody>
<tr>
<td>me, Alyssa (10)</td>
<td>hacking, meetups</td>
<td>javascript - Hello again Be</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>has:nouserlabels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alyssa P. Hacker</td>
</tr>
<tr>
<td>me, Alyssa (10)</td>
</tr>
</tbody>
</table>
slightly surprising behavior #4
slightly surprising behavior #4

<table>
<thead>
<tr>
<th>me, Alyssa (13)</th>
<th>hacking</th>
<th>meetups</th>
<th>todo</th>
<th>javascript - Hello</th>
<th>11:48 am</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andy from Google</td>
<td>Updates</td>
<td>Ben, welcome to your new Google</td>
<td>9:01 am</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
slightly surprising behavior #4

Empty Trash now (messages that have been in Trash more than 30 days will be automatically deleted)

- me, Alyssa (13)  hacking  meetups  todo  javascript - Hello  11:48 am
- Andy from Google  Updates  Ben, welcome to your new Google  9:01 am

label:todo

There are no conversations with this label.
slightly surprising behavior #4

Empty Trash now (messages that have been in Trash more than 30 days will be automatically deleted)

- me, Alyssa (13) 11:48 am
- Andy from Google 9:01 am

There are no conversations with this label.

- label:todo label:trash

- me, Alyssa 10:11 am
slightly surprising behavior #4

- Empty Trash now: messages that have been in Trash more than 30 days will be automatically deleted.
  - me, Alyssa (13) 11:48 am
  - Andy from Google 9:01 am

- label:todo
  - There are no conversations with this label.

- label:todo label:trash
  - me, Alyssa 10:11 am

- label:todo OR label:meetup
  - Some messages in Trash or Spam match your search. View messages.
slightly surprising behavior #5

<table>
<thead>
<tr>
<th>Primary</th>
<th>Social</th>
<th>Promotions</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️ ⭐️</td>
<td>Alyssa P. Hacker</td>
<td>Promotions</td>
</tr>
<tr>
<td>Category</td>
<td>User</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Promotions</td>
<td>Alyssa P. Hacker</td>
<td>buy this! - My new JS book is out</td>
</tr>
<tr>
<td>Social</td>
<td>Alyssa P. Hacker</td>
<td>buy this! - My new JS book is out!</td>
</tr>
</tbody>
</table>
slightly surprising behavior #5

<table>
<thead>
<tr>
<th>Label</th>
<th>From</th>
<th>Subject</th>
<th>10:33 am</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alyssa P. Hacker</td>
<td>buy this! - My new JS book is out</td>
<td>10:33 am</td>
</tr>
</tbody>
</table>

(label: social)
slightly surprising behavior #5

<table>
<thead>
<tr>
<th>Label</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Promotions</td>
<td>Alyssa P. Hacker - My new JS book is out</td>
</tr>
<tr>
<td>Social</td>
<td>Alyssa P. Hacker - My new JS book is out</td>
</tr>
<tr>
<td>label:social</td>
<td>Alyssa P. Hacker - My new JS book is out</td>
</tr>
<tr>
<td>label:promotions</td>
<td>Alyssa P. Hacker - My new JS book is out</td>
</tr>
</tbody>
</table>
slightly surprising behavior #5

<table>
<thead>
<tr>
<th>Label</th>
<th>Message</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Promotions</td>
<td>buy this! - My new JS book is out</td>
<td>10:33 am</td>
</tr>
<tr>
<td>Social</td>
<td>buy this! - My new JS book is out</td>
<td>10:33 am</td>
</tr>
<tr>
<td>Inbox, Promotions, Social</td>
<td>buy this! - My new JS book is out</td>
<td>10:33 am</td>
</tr>
</tbody>
</table>

No messages matched your search. Try using search options such as sender, date, size and more.
find a partner so you can work in a pair

pick one of the Gmail surprises
all slides at https://tinyurl.com/ssft9a

analyze it in term of concepts
what are the key concepts involved?
which concept(s) is responsible for the surprise?
is the surprise a bug, a conceptual flaw or a user misunderstanding?
can you explain precisely what's going wrong?
can you generalize your observation?

design a fix
propose a modification that eliminates the surprise
lecture
two
three design problems
gmail
categories
### Gmail's Categories

<table>
<thead>
<tr>
<th>Primary</th>
<th>Social</th>
<th>Promotions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>New sign-in from Chrome on Mac - New sign-in from Cr</td>
<td>23 new</td>
</tr>
<tr>
<td>Keith Muhammad at DeMont.</td>
<td>DeMontrond Auto Group - 14101 North Freeway Houston</td>
<td>10:37 am</td>
</tr>
<tr>
<td>AT&amp;T High Speed Internet.</td>
<td>AT&amp;T High Speed Internet Service Activation - Your A1</td>
<td>12:19 pm</td>
</tr>
<tr>
<td>Keith Muhammad at DeMont.</td>
<td>DeMontrond Auto Group - 14101 North Freeway Houston</td>
<td>Aug 26</td>
</tr>
<tr>
<td>betterbatonrougejobs.com</td>
<td>Job Update -- 2015-08-26 - Looking For An Advantage W</td>
<td>Aug 26</td>
</tr>
</tbody>
</table>
Choose which message categories to show as inbox tabs. Other messages will appear in the Primary tab.
some reactions
Everything You Need to Know About Gmail's New, Super-Confusing Layout

Melanie Pinola
Filed to: GMAIL  6/07/13 9:00am

301,482  
12  

some reactions
Like many Gmail users, I greeted the news of the introduction of tabs to the interface with a degree of anticipation -- now it was just a matter of waiting for the feature to roll out so I could try it for myself. Earlier today I was randomly signed out of my Gmail account, and after signing back in and checking the settings menus, I could see that tabs were now available to me. Excitement was short-lived, however; it quickly became apparent that this new feature is a disaster.
Are you a Gmail user? Did you wake up a week or two ago to find that your new messages were now being automatically organized by Gmail into tabs of different, pre-determined categories? And, did you think, like me, that they were really ugly, stupid, and unnecessary? Here's a quick tip on how to rid yourself of them!
Using labels

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

associate tabs with labels
feature available only for categories
what you can’t do

associate tabs with labels
feature available only for categories

create new categories
only new labels
what you can’t do

associate tabs with labels
feature available only for categories

create new categories
only new labels

use tabs outside inbox
tabs disappear when you filter on a label
fuji aspect ratio
my camera fuji x100s
image quality setting
image quality setting
image quality setting

- Fine
- Normal
- Fine+RAW
- Normal+RAW
- RAW
aspect ratio
aspect ratio
image size setting
image size setting
image size setting
image size setting
non-standard ratio + raw?
what you can’t do

non-standard aspect ratio + raw
even though raw images get nice nondestructive crop!
what you can’t do

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

define a style that italicizes
  Arno Regular to Arno Italic
  Futura Book to Futura Book Oblique
  Magma Light to Magma Light Italic
introducing a concept
introducing a concept

Keynote ’09: has subfamilies
introducing a concept

Keynote '09: has subfamilies

Keynote 6: gone again!
What's going on?
what's going on?

gmail
one purpose :: two concepts
organizing messages :: label + category
what’s going on?

gmail
one purpose :: two concepts
organizing messages :: label + category

camera
two purposes :: one concept
aspect ratio + image resolution :: image size
what’s going on?

gmail
one purpose :: two concepts
organizing messages :: label + category

camera
two purposes :: one concept
aspect ratio + image resolution :: image size

style
one purpose :: no concept
specify a font-independent styling :: ?
the singularity rule
Mitchell and Webb: “Unity of Purpose”
Mitchell and Webb: “Unity of Purpose”
one-to-one mapping

purposes

P1 —— C1

P2 —— C2

concepts
one-to-one mapping

purposes

P1

C1

P2

C2

concepts

Nam Suh: Axiomatic Design
four ways to fail

- unfulfilled purpose
  - P1
  - C1

- unmotivated concept
  - P1
  - C1
  - C2

- overloaded concept
  - P1
  - C1

- redundant concepts
  - P1
  - C1
  - C2
kinds of overloading
overloaded concepts

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

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

4 forms of overloading:
- piggybacking: new purpose hacked onto old concept
- false convergence: two purposes looked the same
- emergent purpose: second purpose emerged with use
- denial: designer believes second purpose unnecessary
piggybacking fuji camera

new purpose hacked onto old concept
piggybacking fuji camera

new purpose hacked onto old concept

image size
aspect ratio piggybacked on JPEG dimensions
piggybacking epson driver
piggybacking epson driver
piggybacking epson driver
piggybacking epson driver
result: can’t create custom size for front loading
also, page size presets in Lightroom hold feed setting
false convergence facebook friend

two purposes looked the same

filter incoming posts
control access to my posts
distinct purposes

2011: Facebook added subscribe/follow
emergent purpose email subject

users find second purpose for concept
users find second purpose for concept

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

initial purpose: summarize content
emergent purpose email subject

users find second purpose for concept

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

initial purpose: summarize content

To: csail-related@lists.csail.mit.edu
Re: [csail-related] turn off the lights?

emergent purpose: show sender
if you bcc a list, subject reveals to-address

thanks to Shriram Krishnamurthi
users find second purpose for concept

initial purpose: summarize content

emergent purpose: show sender
if you bcc a list, subject reveals to-address

thanks to Shriram Krishnamurthi

emergent purpose: group by conversation

emergent purpose: show sender
if you bcc a list, subject reveals to-address

thanks to Shriram Krishnamurthi

emergent purpose: group by conversation

emergent purpose: group by conversation

can’t label reservations from Expedia by trip

thanks to Eunsuk Kang
designer believes second purpose unnecessary
the uniformity rule
what makes a usable concept?

operational principle is uniform
always the same actions, irrespective of context
what makes a usable concept?

operational principle is uniform
always the same actions, irrespective of context
what makes a usable concept?

operational principle is uniform
always the same actions, irrespective of context
what makes a usable concept?

operational principle is uniform
always the same actions, irrespective of context

concept: **Group** (Keynote)
*purpose*: treat set as one
*OP*: ... select(objs); group(); mutate()...
what makes a usable concept?

operational principle is uniform
always the same actions, irrespective of context

concept: **Group** (Keynote)
purpose: treat set as one
op: … select(objs); group(); mutate()…

quantified over state & args
what makes a usable concept?

operational principle is uniform
always the same actions, irrespective of context

concept: **Group** (Keynote)

*purpose*: treat set as one

*OP*: ... select(objs); group(); mutate()...

*unless* objs contains a text body object

 quantified over state & args
non-uniformity range
concept: **Range** (Numbers)
purpose: define formula over adjustable group of cells
*OP*: ... define formula over range... select(c) in range... add(direction)... formula updated
**concept:** Range (Numbers)

**purpose:** define formula over adjustable group of cells

**OP:** ... define formula over range... select(c) in range... add(direction)...formula updated

unless range cell c is at top of range and dir is above or...
non-uniformity conversation

action applied to every message in conversation
non-uniformity conversation

action applied to every message in conversation **unless** message in other folder or action is reply ...
kinds of non-uniformity

Keynote grouping unless objs contains a text body object

Fuji aspect ratio setting unless set to raw only mode

Dropbox share folder unless folder is ancestor or descendant of shared folder

Git branch unless working directory contains uncommitted file or...

Twitter mention unless mention includes first character of tweet
kinds of non-uniformity

- Keynote grouping: *unless* obj contains a text body object
- Fuji aspect ratio setting: *unless* set to raw only mode
- Dropbox share folder: *unless* folder is ancestor or descendant of shared folder
- Git branch: *unless* working directory contains uncommitted file or...
- Twitter mention: *unless* mention includes first character of tweet
kinds of non-uniformity

- **varies over type**: Keynote grouping **unless** `objs` contains a text body object
- **varies over mode**: Fuji aspect ratio setting **unless** set to raw only mode
- Dropbox share folder **unless** folder is ancestor or descendant of shared folder
- Git branch **unless** working directory contains uncommitted file or...
- Twitter mention **unless** mention includes first character of tweet
kinds of non-uniformity

- **varies over type**: Keynote grouping **unless** `objs` contains a text body object
- **varies over mode**: Fuji aspect ratio setting **unless** set to raw only mode
- **varies over state**: Dropbox share folder **unless** folder is ancestor or descendant of shared folder
- **varies over state**: Git branch **unless** working directory contains uncommitted file or...
- **varies over state**: Twitter mention **unless** mention includes first character of tweet
kinds of non-uniformity

- **varies over type**: Keynote grouping unless `objs` contains a text body object
- **varies over mode**: Fuji aspect ratio setting unless set to raw only mode
- **varies over state**: Dropbox share folder unless folder is ancestor or descendant of shared folder
- **varies over state**: Git branch unless working directory contains uncommitted file or...
- **varies over arg**: Twitter mention unless mention includes first character of tweet
the genericity rule
how concepts get applied

related

upvote

notification

NY Times  StackExchange  Amazon
how concepts get applied

upvote

notification

related

NY Times

StackExchange

Amazon

comments
how concepts get applied

NY Times  StackExchange  Amazon

upvote  comments  answers

notification

related
how concepts get applied
how concepts get applied

- upvote
- comments
- answers
- reviews
- notification
- breaking news
- related

NY Times
StackExchange
Amazon
how concepts get applied

upvote
notification
related

NY Times
StackExchange
Amazon

comments
answers
reviews
breaking news
replies
how concepts get applied

**upvote**

**notification**

**related**

<table>
<thead>
<tr>
<th>NY Times</th>
<th>StackExchange</th>
<th>Amazon</th>
</tr>
</thead>
<tbody>
<tr>
<td>comments</td>
<td>answers</td>
<td>reviews</td>
</tr>
<tr>
<td>breaking news</td>
<td>replies</td>
<td>when shipped</td>
</tr>
</tbody>
</table>
how concepts get applied

- upvote
- notification
- related
- comments
- breaking news
- articles
- NY Times
- StackExchange
- answers
- replies
- Amazon
- reviews
- when shipped
how concepts get applied
how concepts get applied

NY Times
- upvote
- comments

StackExchange
- answers
- replies

Amazon
- reviews
- when shipped

related
- articles
- questions
- items
why reuse a concept?

familiarity
users will get it

save work
design options known

no surprises
misfits anticipated
why reuse a concept?

familiarity
users will get it

save work
design options known

no surprises
misfits anticipated

options for upvote?
why reuse a concept?

- familiarity
- users will get it
- save work
- design options known
- no surprises
- misfits anticipated

options for upvote?

misfit of notification?
the genericity rule

reusing a well-known generic concept is usually preferable to inventing one
the genericity rule

reusing a well-known generic concept is usually preferable to inventing one
the genericity rule

reusing a well-known generic concept is usually preferable to inventing one
the genericity rule

reusing a well-known generic concept is usually preferable to inventing one

in Powerpoint

overloading (7)
uniformity (5)
generics (4)

27
generic concepts

28
which concepts get applied

29
why reuse a concept?

30
meaning and not

in Keynote

kinds of overloading

the uniformity rule

generic concepts

what would you call this concept?

meaning and not
the genericity rule

reusing a well-known generic concept is usually preferable to inventing one

what would you call this concept?
the genericity rule

reusing a well-known generic concept is usually preferable to inventing one.

in Powerpoint

in Keynote

Powerpoint commands

what role does slide selection play in add?

what would you call this concept?
concept
composition
example: reservation

**name**  reservation

**purpose**  make access to shared resource reliable

**structure**  reserved: bool = false

**behavior**

```plaintext
reserve()
  reserved := true
use()
  reserved => reserved := false
cancel()
  reserved => reserved := false
```

**tactic**  if reserve() and no cancel then can use()
example: reservation

<table>
<thead>
<tr>
<th>name</th>
<th>reservation</th>
</tr>
</thead>
<tbody>
<tr>
<td>purpose</td>
<td>make access to shared resource reliable</td>
</tr>
<tr>
<td>structure</td>
<td>reserved: bool = false</td>
</tr>
</tbody>
</table>
| behavior | reserve()  
reserved := true  
use ()  
reserved => reserved := false  
cancel ()  
reserved => reserved := false |
| tactic | if reserve() and no cancel then can use() |
Example: Reservation

**Name**: reservation

**Purpose**: make access to shared resource reliable

**Behavior**: define state space

```plaintext
reserved: bool = false

reserve()
reserved := true

use()
reserved => reserved := false

cancel()
reserved => reserved := false
```

**Tactic**: if reserve() and no cancel then can use()

**Structure**: defines state space

**Actions**: give a labeled transition relation defining a trace set
example: reservation

<table>
<thead>
<tr>
<th>name</th>
<th>reservation</th>
</tr>
</thead>
<tbody>
<tr>
<td>purpose</td>
<td>make access to shared resource reliable</td>
</tr>
<tr>
<td>structure</td>
<td>reserved: bool = false</td>
</tr>
</tbody>
</table>
| behavior | reserve()  
reserved := true  
use ()  
reserved => reserved := false  
cancel ()  
reserved => reserved := false |
| tactic  | if reserve() and no cancel then can use() |

structure defines state space

actions give a labeled transition relation defining a trace set

```plaintext
{<>,
<reserve>,
<reserve, cancel>,
<reserve, use>,
<reserve, use, cancel>,
...}
```
example: reservation

name: reservation

purpose: make access to shared resource reliable

structure:
reserved: bool = false

behavior:
reserve()
   reserved := true
use ()
   reserved => reserved := false
cancel ()
   reserved => reserved := false

tactic:
if reserve() and no cancel then can use()
example: authentication

name | authentication

purpose | identify participant in interaction

structure | ok: bool = false

behavior | login()
         ok := true
logout()
         ok => ok := false
auth()
         ok =>

tactic | if login() and no logout() then can auth()
example: authentication

name authentication

purpose identify participant in interaction

structure ok: bool = false

behavior

login()
  ok := true
logout()
  ok => ok := false
auth()
  ok =>

tactic if login() and no logout() then can auth()

what are the traces?
### example: authentication

<table>
<thead>
<tr>
<th>name</th>
<th>authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>purpose</td>
<td>identify participant in interaction</td>
</tr>
<tr>
<td>structure</td>
<td>ok: bool = false</td>
</tr>
<tr>
<td>behavior</td>
<td>login()</td>
</tr>
<tr>
<td></td>
<td>ok := true</td>
</tr>
<tr>
<td></td>
<td>logout()</td>
</tr>
<tr>
<td></td>
<td>ok =&gt; ok := false</td>
</tr>
<tr>
<td></td>
<td>auth()</td>
</tr>
<tr>
<td></td>
<td>ok =&gt;</td>
</tr>
<tr>
<td>tactic</td>
<td>if login() and no logout() then can auth()</td>
</tr>
</tbody>
</table>

```c
{<>,
 <login>,
 <login, auth>,
 <login, auth, auth>,
 <login, logout>,
 <login, auth, logout>,
 ...}
```
composing concepts

application  MyReservationApp

includes  reservation, authentication

behavior

login
  authentication.login
logout
  authentication.logout
reserve:
  reservation.reserve
  authentication.auth
cancel:
  reservation.cancel
  authentication.auth
use:
  reservation.use
composing concepts

application: MyReservationApp

includes: reservation, authentication

behavior:
  login: authentication.login
  logout: authentication.logout
  reserve:
    reservation.reserve
    authentication.auth
  cancel:
    reservation.cancel
    authentication.auth
  use:
    reservation.use
composing concepts

**application**
- MyReservationApp

**includes**
- reservation, authentication

**behavior**
- login
  - authentication.login
- logout
  - authentication.logout
- reserve:
  - reservation.reserve
  - authentication.auth
- cancel:
  - reservation.cancel
  - authentication.auth
- use:
  - reservation.use

**concepts used**

**action of app is binding of concept actions**

**binding:** Action -> Concept -> Action

{(login, authentication, login),
 (logout, authentication, logout),
 (reserve, reservation, reserve),
 (reserve, authentication, auth),
 (cancel, reservation, cancel),
 (cancel, authentication, auth),
 (use, reservation, use)}
traces of reservation

\{\langle\rangle, \\
  \langle\text{reserve}\rangle, \\
  \langle\text{reserve, cancel}\rangle, \\
  \langle\text{reserve, use}\rangle, \\
  \langle\text{reserve, use, cancel}\rangle, \\
  \ldots \\
\}
semantics of composition

traces of **reservation**

\{<>,
    <reserve>,
    <reserve, cancel>,
    <reserve, use>,
    <reserve, use, cancel>,
    ...
\}

traces of **authentication**

\{<>,
    <login>,
    <login, auth>,
    <login, auth, auth>,
    <login, logout>,
    <login, auth, logout>,
    ...
\}
## Semantics of Composition

### Traces of Reservation

\{<>,
<reserve>,
<reserve, cancel>,
<reserve, use>,
<reserve, use, cancel>,
...
\}

### Traces of Authentication

\{<>,
<login>,
<login, auth>,
<login, auth, auth>,
<login, logout>,
<login, auth, logout>,
...
\}

### Binding: Action \rightarrow Concept \rightarrow Action

\{(login, authentication, login),
(logout, authentication, logout),
(reserve, reservation, reserve),
(reserve, authentication, auth),
(cancel, reservation, cancel),
(cancel, authentication, auth),
(use, reservation, use)\}
semantics of composition

traces of **reservation**

\{
  \langle\rangle,
  \langle\text{reserve}\rangle,
  \langle\text{reserve, cancel}\rangle,
  \langle\text{reserve, use}\rangle,
  \langle\text{reserve, use, cancel}\rangle,
  \ldots
\}

traces of **authentication**

\{
  \langle\rangle,
  \langle\text{login}\rangle,
  \langle\text{login, auth}\rangle,
  \langle\text{login, auth, auth}\rangle,
  \langle\text{login, logout}\rangle,
  \langle\text{login, auth, logout}\rangle,
  \langle\text{cancel}\rangle,
  \langle\text{login, auth, cancel}\rangle,
  \langle\text{cancel}\rangle,
  \langle\text{login, auth, cancel}\rangle,
  \langle\text{use}\rangle,
  \langle\text{login, reservation, use}\rangle,
  \langle\text{login, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,
  \langle\text{login, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation, reservation}\rangle,

map trace \( t \) onto concept \( C \) with binding \( B \)

let map\( (t, C, B) = \)
map \( (\langle\rangle, C, B) = \langle\rangle \)
map \( (\text{append}(t, a), C, B) = \)
  if no \( C.(a.B) \) then map\( (t, C, B) \) 
  else append(map\( (t, C, B), C.(a.B) \))
semantics of composition

traces of reservation

\{<>,
<reserve>,
<reserve, cancel>,
<reserve, use>,
<reserve, use, cancel>,
...
\}

traces of authentication

\{<>,
<login>,
<login, auth>,
<login, auth, auth>,
<login, logout>,
<login, auth, logout>,
...
\}

binding: Action -> Concept -> Action

\{(login, authentication, login),
(logout, authentication, logout),
(reserve, reservation, reserve),
(reserve, authentication, auth),
(cancel, reservation, cancel),
(cancel, authentication, auth),
(use, reservation, use)\}

map trace t onto concept C with binding B

let map(t, C, B) =
map (<>, C, B) = <>
map (append(t, a), C, B) =
  if no C.(a.B) then map(t, C, B)
  else append(map(t,C, B), C.(a.B))

traces are all those consistent with concept traces

traces = \{t in action* \mid all C: includes I map(t, C, B) in traces(C)\}
semantics of composition

traces of reservation

\{<>,
<reserve>,
<reserve, cancel>,
<reserve, use>,
<reserve, use, cancel>,
...
\}

traces of authentication

\{<>,
<login>,
<login, auth>,
<login, auth, auth>,
<login, logout>,
<login, auth, logout>,
<login, auth, logout>,
...
\}

binding: Action -> Concept -> Action

\{(login, authentication, login),
(logout, authentication, logout),
(reserve, reservation, reserve),
(reserve, authentication, auth),
(cancel, reservation, cancel),
(cancel, authentication, auth),
(use, reservation, use)\}

map trace t onto concept C with binding B

let map(t, C, B) =
map (<>, C, B) = <>
map (append(t, a), C, B) =
  if no C.(a.B) then map(t, C, B)
  else append(map(t,C, B), C.(a.B))

traces are all those consistent with concept traces

traces = \{t in action* | all C: includes | map(t, C, B) in traces(C)\}
reservation

name
reservation

purpose
make access to shared resource reliable

structure
slots: Owner -> Slot
holds: User -> Slot

behavior
create (o: Owner, s: Slot)
  no slots.s => slots += o -> s
reserve (u: User, o: Owner, s: Slot)
  no holds.s and o -> s in slots => holds += u -> s
cancel (u: User, s: Slot)
  u -> s in holds => holds -= u -> s
use (u: User, o: Owner, s: Slot)
  u -> s in holds and o -> s in slots =>

tactic
if create(o,s); reserve(u,o,s); ... no cancel(u,s) ... then can use(u,o,s)
authentication (again)

- **name**: authentication
- **purpose**: identify participant in interaction
- **structure**: password: User -> Password
  sessions: set User
- **behavior**: register (u: User, p: Password)
  - no u.password => password += u -> p
  login (u: User, p: Password)
  - u.password = p => sessions += u
  logout (u: User)
  - u in sessions => sessions -= u
  auth (u: User)
  - u in sessions =>

**tactic**: if register(u,p), login(u,p), no logout(u) then can auth(u)
**name**
rating

**purpose**
identify participant in interaction

**structure**
used: User -> Item
rated: User -> Item -> Int
rating: Item -> Int = \{i: Item, r: Int | avg (User, rated)\}

**behavior**
use (u: User, i: Item)
  used += u -> i
rate (u: User, i: Item, r: Int)
  u -> i in used => u.rated += i -> r
show (i: Item): Int
  result = i.rating

**tactic**
if user(u,i), rate(u,i,r)... for multiple u... and show(i):r then r is avg of user’s ratings
reservation app (again)

application

MyReservationApp

includes

reservation, authentication, rating

behavior

register(u,p)
authentication.register(u,p)
login(u,p)
authentication.login(u,p)
logout(u)
authentication.logout(u)
reserve(u,o,s)
reservation.reserve(u,o,s)
use(u,o,s)
reservation.use(u,o,s)
cancel(u,s)
reservation.cancel(u,s)

rate(u,o,r)
authentication.auth(u)
rating.rate(u,o,r)
showRating(o)
rating.show(o)
reservation app (again)

- **application**: MyReservationApp
- **includes**: reservation, authentication, rating
- **behavior**:
  - register(u,p)
  - authentication.register(u,p)
  - login(u,p)
  - authentication.login(u,p)
  - logout(u)
  - authentication.logout(u)
  - reserve(u,o,s)
  - reservation.reserve(u,o,s)
  - use(u,o,s)
  - reservation.use(u,o,s)
  - cancel(u,s)
  - reservation.cancel(u,s)
  - rate(u,o,r)
  - authentication.auth(u)
  - rating.rate(u,o,r)
  - showRating(o)
  - rating.show(o)

**ratings are authenticated**
reservation app (again)

application | MyReservationApp

includes | reservation, authentication, rating

behavior

- register(u,p)
  - authentication.register(u,p)
- login(u,p)
  - authentication.login(u,p)
- logout(u)
  - authentication.logout(u)
- reserve(u,o,s)
  - reservation.reserve(u,o,s)
  - authentication.auth(u)
- use(u,o,s)
  - reservation.use(u,o,s)
  - rating.use(u,o)
- cancel(u,s)
  - reservation.cancel(u,s)
  - authentication.auth(u)

rate(u,o,r)
  - authentication.auth(u)
  - rating.rate(u,o,r)
  - rating.show(o)

ratings are authenticated

can’t rate until you’ve used reservation
the integrity rule
looking at sent messages in gmail
looking at sent messages in gmail
looking at sent messages in gmail

Alyssa P. Hacker <alyssa.pure.hacker@gmail.com>

to me

Reminds you of the old days, eh?

Ben Bitdiddle <benito.bitdiddle@gmail.com>

to Alyssa

Yes, it does.

Alyssa P. Hacker

JavaScript makes me feel nostalgic for Scheme.

Ben Bitdiddle <benito.bitdiddle@gmail.com>

to Alyssa

Is JavaScript just Scheme with prototypes and some hacky coercions?
looking at sent messages in gmail

can’t see which messages were sent
interpreting composite behavior

each action in composite system interpreted as zero or more actions in each concept
interpreting composite behavior

each action in composite system interpreted as zero or more actions in each concept
the integrity rule

when concepts are combined, each concept's behavior and OP should still apply

a simple criterion
projected behavior must satisfy concept spec:
∀ c: concept | ∀ t: traces(sys) | Rc(t) ∈ traces(c)
the integrity rule

when concepts are combined, each concept’s behavior and OP should still apply

a simple criterion

projected behavior must satisfy concept spec:

$\forall c: \text{concept} \mid \forall t: \text{traces(sys)} \mid R_c(t) \in \text{traces}(c)$
a simple criterion
projected behavior must satisfy concept spec:
\[ \forall c: \text{concept} \mid \forall t: \text{traces(sys)} \mid R_c(t) \in \text{traces}(c) \]
the integrity rule

when concepts are combined, each concept’s behavior and OP should still apply

a simple criterion
projected behavior must satisfy concept spec:
∀ c: concept | ∀ t: traces(sys) | \( R_c(t) \in \text{traces}(c) \)
the integrity rule

when concepts are combined, each concept’s behavior and OP should still apply

a simple criterion

projected behavior must satisfy concept spec:

∀ c: concept | ∀ t: traces(sys) | R_c(t) ∈ traces(c)
the integrity rule

when concepts are combined, each concept’s behavior and OP should still apply

a simple criterion

projected behavior must satisfy concept spec:

\[ \forall c: \text{concept} \mid \forall t: \text{traces(sys)} \mid R_c(t) \in \text{traces}(c) \]
the integrity rule
when concepts are combined, each concept’s behavior and OP should still apply

a simple criterion
projected behavior must satisfy concept spec:
\[ \forall c: \text{concept} \mid \forall t: \text{traces(sys)} \mid R_c(t) \in \text{traces}(c) \]
the integrity rule

when concepts are combined, each concept’s behavior and OP should still apply

a simple criterion
projected behavior must satisfy concept spec:

\[ \forall c: \text{concept} \mid \forall t: \text{traces(sys)} \mid R_c(t) \in \text{traces}(c) \]
the integrity rule

when concepts are combined, each concept’s behavior and OP should still apply

a simple criterion

projected behavior must satisfy concept spec:

$$\forall \ c: \text{concept} \mid \forall \ t: \text{traces(sys)} \mid R_c(t) \in \text{traces}(c)$$
the label concept
the label concept

name  label
the label concept

name: label

purpose: organize items for easy retrieval
the label concept

- **name**: label
- **purpose**: organize items for easy retrieval
- **structure**: labels: $X \rightarrow \text{Label}$
the label concept

name: label

purpose: organize items for easy retrieval

structure: labels: X -> Label

behavior:

mark (x: X, p: Label)
    labels += x -> p

unmark (x: X, p: Label)
    p in x.labels => labels -= x -> p

find (ps: set Label): set X
    result = {x | ps in x.labels}
the label concept

name | label

purpose | organize items for easy retrieval

structure | labels: X -> Label

behavior | mark (x: X, p: Label)
        | labels += x -> p
unmark (x: X, p: Label)
        | p in x.labels => labels -= x -> p
find (ps: set Label): set X
        | result = {x | ps in x.labels}

if mark(x, p); find(p):xs then x in xs
if no mark(x, P); find(p):xs then x !in xs
conversation breaks label
Alyssa P. Hacker <alyssa.pure.hacker@gmail.com>  
Reminds you of the old days, eh?

Ben Bitdiddle <benito.bitdiddle@gmail.com>  
Yes, it does.

Alyssa P. Hacker  
JavaScript makes me feel nostalgic for Scheme.

Ben Bitdiddle <benito.bitdiddle@gmail.com>  
Is JavaScript just Scheme with prototypes and some hacky coercions?
conversation breaks label

when message $m$ is sent

Label.mark($m$, 'sent')

occurs implicitly
conversation breaks label

when message \( m \) is sent

\[
\text{Label.mark}(m, \text{`sent'})
\]

occurs implicitly

when Sent link is clicked

\[
\text{Label.find('sent'):ms}
\]

occurs
conversation breaks label

when message \( m \) is sent
\[
\text{Label.mark}(m, \text{`sent'})
\]
occurs implicitly

when Sent link is clicked
\[
\text{Label.find(`sent')}:ms
\]
occurs

but \( ms \) includes
messages never marked
integrity violations trash
what happens when you unmount a drive?
interaction of Trash and Volume (Apple Finder)
unmount of Volume removes files from Trash
not expressible in terms of Trash actions
a solution: one trash/volume?
deja vu
reversing the process
reversing the process

the same concepts, again & again
post, comment, upvote, notification, ...

hard work to build
libraries often just client- or server-side
easy in a CMS, but structure hard-wired
reversing the process

the same concepts, again & again
post, comment, upvote, notification, ...

hard work to build
libraries often just client- or server-side
easy in a CMS, but structure hard-wired

idea: concept cliches
full stack implementation
app-specific assembly
in HTML, no JS or backend code

action synchronization
build app action by joining cliche actions
architecture of deja vu

client

client-side library

post

comment

upvote

gateway

post

comment

upvote

server
architecture of deja vu

client

client-side library

post

comment

upvote

app

action

gateway

post

comment

upvote

server
a sample app
A sample app
<table>
<thead>
<tr>
<th>DV Hacker News</th>
<th>Submit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SPLASH 2018</strong> (2018.splashcon.org)</td>
<td>3 points by eva</td>
</tr>
<tr>
<td><strong>Software Design Group</strong> (sdg.csail.mit.edu)</td>
<td>2 points by eva</td>
</tr>
<tr>
<td><strong>Concept design</strong> (sdg.csail.mit.edu)</td>
<td>1 points by alyssa</td>
</tr>
<tr>
<td><strong>Déjà Vu</strong> (sdg.csail.mit.edu)</td>
<td>1 points by ben</td>
</tr>
<tr>
<td>DV Hacker News</td>
<td>Submit</td>
</tr>
<tr>
<td>----------------</td>
<td>--------</td>
</tr>
<tr>
<td><strong>SPLASH 2018</strong> (2018.splashcon.org)</td>
<td>3 points by eva</td>
</tr>
<tr>
<td><strong>Software Design Group</strong> (sdg.csail.mit.edu)</td>
<td>2 points by eva</td>
</tr>
<tr>
<td><strong>Concept design</strong> (sdg.csail.mit.edu)</td>
<td>1 points by alyssa</td>
</tr>
<tr>
<td><strong>Déjà Vu</strong> (sdg.csail.mit.edu)</td>
<td>1 points by ben</td>
</tr>
<tr>
<td>DV Hacker News Submit</td>
<td>ben</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----</td>
</tr>
<tr>
<td><strong>SPLASH 2018</strong> (2018.splashcon.org)</td>
<td>auth</td>
</tr>
<tr>
<td>3 points by eva</td>
<td>comments</td>
</tr>
<tr>
<td><strong>Software Design Group</strong> (sdg.csail.mit.edu)</td>
<td></td>
</tr>
<tr>
<td>2 points by eva</td>
<td>comments</td>
</tr>
<tr>
<td><strong>Concept design</strong> (sdg.csail.mit.edu)</td>
<td></td>
</tr>
<tr>
<td>1 points by alyssa</td>
<td>comments</td>
</tr>
<tr>
<td><strong>Déjà Vu</strong> (sdg.csail.mit.edu)</td>
<td></td>
</tr>
<tr>
<td>1 points by ben</td>
<td>comments</td>
</tr>
</tbody>
</table>
a sample app

<table>
<thead>
<tr>
<th>DV Hacker News</th>
<th>Submit</th>
<th>ben</th>
<th>Sign Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLASH 2018</td>
<td>3 points by eva</td>
<td>comments</td>
<td></td>
</tr>
<tr>
<td>Software Design Group</td>
<td>2 points by eva</td>
<td>comments</td>
<td></td>
</tr>
<tr>
<td>Concept design</td>
<td>1 points by alyssa</td>
<td>comments</td>
<td></td>
</tr>
<tr>
<td>Déjà Vu</td>
<td>1 points by ben</td>
<td>comments</td>
<td></td>
</tr>
</tbody>
</table>
```json
{
  "name": "hackernews",
  "usedCliches": {
    "authentication": {},
    "comment": {},
    "property": {…},
    "scoringposts": {
      "name": "scoring"
    },
    "scoringcomments": {
      "name": "scoring"
    }
  }
}
```
a sample app

<table>
<thead>
<tr>
<th>DV Hacker News</th>
<th>Submit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **SPLASH 2018** (2018.splashcon.org)
  - 3 points by eva | comments
- **Software Design Group** (sdg.csail.mit.edu)
  - 2 points by eva | comments
- **Concept design** (sdg.csail.mit.edu)
  - 1 points by alyssa | comments
- **Déjà Vu** (sdg.csail.mit.edu)
  - 1 points by ben | comments

```json
{  "name": "hackernews",  "usedCliches": {    "authentication": {},    "comment": {},    "property": {...},    "scoringposts": {"name": "scoring"},    "scoringcomments": {"name": "scoring"}  },  "routes": [    { "route": ", "action": "home" },    { "route": "news", "action": "home" },    { "route": "post", "action": "post-detail" },    { "route": "login", "action": "login" },    { "route": "submit", "action": "submit-post" }  ]}
```
<table>
<thead>
<tr>
<th>Title</th>
<th>URL</th>
<th>Points</th>
<th>Author</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLASH 2018</td>
<td>2018.splashcon.org</td>
<td>3</td>
<td>eva</td>
<td>comments</td>
</tr>
<tr>
<td>Software Design Group</td>
<td>sdg.csail.mit.edu</td>
<td>2</td>
<td>eva</td>
<td>comments</td>
</tr>
<tr>
<td>Concept design</td>
<td>sdg.csail.mit.edu</td>
<td>1</td>
<td>alyssa</td>
<td>comments</td>
</tr>
<tr>
<td>Déjà Vu</td>
<td>sdg.csail.mit.edu</td>
<td>1</td>
<td>ben</td>
<td>comments</td>
</tr>
</tbody>
</table>
<dv.action name="home">
<hackernews.navbar />
<div class="main">
<scoringposts.show-targets-by-score
  noTargetsText="No posts yet"
  showAscending=false
  showScores=false
  showTarget=<hackernews.show-post post=$target id=$id />
</scoringposts.show-targets-by-score>
</div>
</dv.action>
Submit post action

DV Hacker News  Submit

Title *

Url *

Submit
<dv.action name="submit-post">
<hackernews.navbar />
<div class="main"> <dv.tx>
  <dv.gen-id />
  <property.create-object
    id=dv.gen-id.id
    initialValue={ author: hackernews.navbar.user.username }
    showExclude=["author"]
    buttonLabel="submit"
    newObjectSavedText="Post submitted" />
  <scoringposts.create-score
    targetId=dv.gen-id.id
    value=0
    hidden=true />
  <authentication.authenticate id=hackernews.navbar.user hidden=true />
  <dv.link href="/item" params={ id: dv.gen-id.id } />
</dv.tx> </div> </dv.action>
<dv.action name="submit-post">
<hackernews.navbar />
<div class="main"> <dv.tx>
<dv.gen-id />

<property.create-object
id=dv.gen-id.id
initialValue={ author: hackernews.navbar.user.username }
showExclude=["author"]
buttonLabel="submit"
newObjectSavedText="Post submitted" />
<scoringposts.create-score
targetId=dv.gen-id.id
value=0
hidden=true />
<authentication.authenticate id=hackernews.navbar.user hidden=true />
<dv.link href="/item" params={ id: dv.gen-id.id } />
</dv.tx> </div> </dv.action>
<h1>submit post action</h1>

```
<dv.action name="submit-post">
  <hackernews.navbar />
  <div class="main"> <dv.tx>
    <dv.gen-id />
    <property.create-object
      id=dv.gen-id.id
      initialValue={ author: hackernews.navbar.user.username }
      showExclude=["author"]
      buttonLabel="submit"
      newObjectSavedText="Post submitted" />
    <scoringposts.create-score
      targetId=dv.gen-id.id
      value=0
      hidden=true />
    <authentication.authenticate id=hackernews.navbar.user hidden=true />
    <dv.link href="/item" params={ id: dv.gen-id.id } />
  </dv.tx> </div> </dv.action>
```
submit post action

```html
<dv.action name="submit-post">
<hackernews.navbar />
<div class="main"> <dv.tx>
<dv.gen-id />
<property.create-object
  id=dv.gen-id.id
  initialValue={ author: hackernews.navbar.user.username }
  showExclude=["author"]
  buttonLabel="submit"
  newObjectSavedText="Post submitted" />
<scoringposts.create-score
  targetId=dv.gen-id.id
  value=0
  hidden=true />
<authentication.authenticate id=hackernews.navbar.user hidden=true />
<dv.link href="/item" params={ id: dv.gen-id.id } />
</dv.tx> </div> </dv.action>
```
<dv.action name="submit-post">
  <hackernews.navbar />
  <div class="main">
    <dv.tx>
      <dv.gen-id />
      <property.create-object
        id=dv.gen-id.id
        initialValue={ author: hackernews.navbar.user.username }
        showExclude=["author"]
        buttonLabel="submit"
        newObjectSavedText="Post submitted"
      />
      <scoringposts.create-score
        targetId=dv.gen-id.id
        value=0
        hidden=true />
      <authentication.authenticate id=hackernews.navbar.user hidden=true />
      <dv.link href="/item" params={ id: dv.gen-id.id } />
    </dv.tx>
  </div>
</dv.action>
submit post action

```xml
<dv.action name="submit-post">
  <hackernews.navbar />
  <div class="main">
    <dv.gen-id />
    <property.create-object
      id=dv.gen-id.id
      initialValue={ author: hackernews.navbar.user.username }
      showExclude=["author"]
      buttonLabel="submit"
      newObjectSavedText="Post submitted" />
    <scoringposts.create-score
      targetId=dv.gen-id.id
      value=0
      hidden=true />
    <authentication.authenticate id=hackernews.navbar.user hidden=true />
    <dv.link href="/item" params={ id: dv.gen-id.id } />
  </div>
</dv.action>
```
rebuilding class projects

<table>
<thead>
<tr>
<th></th>
<th>Authentication</th>
<th>Authorization</th>
<th>Passkey</th>
<th>Transfer</th>
<th>Rating</th>
<th>Follow</th>
<th>Geolocation</th>
<th>Comment</th>
<th>Event</th>
<th>Label</th>
<th>Group</th>
<th>Task</th>
<th>Scoring</th>
<th>Property</th>
<th>Chat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accord</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>ChoreStar</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MapMIT</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Rendezvous</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SweetSpots</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Potluck</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>GroceryShip</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Lingua</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>LiveScorecard</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
other aspects of déjà vu

WYSIWYG designer (Barry McNamara)

cliche library for social apps (Maryam Archie)

security: stop request forgeries

cliche support (Czarina Lao)
closing thoughts
user-centered design: conceptual model should be designed
formal methods: software defined by its behavior
both originating around 1974
https://tinyurl.com/dbctouch
to keep in touch and be notified about publication of book

https://tinyurl.com/postcard-get
to sign up for monthly resilience postcards
studio 2
construct concept models in this order
post, friend, comment, upvote, tag

specify application binding

for each concept, give
purpose: informally stated
structure: text or diagram
behavior: actions specified formally
tactic: informal scenario

hints: make each concept
minimal: only essential functionality
free-standing: makes sense alone
orthogonal: avoid overlap

what issues came up?
reminder: a reservation concept

**name**

**reservation**

**purpose**

make access to shared resource reliable

**structure**

slots: Owner -> Slot
holds: User -> Slot

**behavior**

create (o: Owner, s: Slot)
  \[\text{no} \; \text{slots} \cdot s \Rightarrow \text{slots} += o \rightarrow s\]

reserve (u: User, o: Owner, s: Slot)
  \[\text{no} \; \text{holds} \cdot s \; \text{and} \; o \rightarrow s \; \text{in} \; \text{slots} \Rightarrow \text{holds} += u \rightarrow s\]

cancel (u: User, s: Slot)
  \[u \rightarrow s \; \text{in} \; \text{holds} \Rightarrow \text{holds} -= u \rightarrow s\]

use (u: User, o: Owner, s: Slot)
  \[u \rightarrow s \; \text{in} \; \text{holds} \; \text{and} \; o \rightarrow s \; \text{in} \; \text{slots} \Rightarrow\]

**tactic**

\[\text{if} \; \text{create}(o,s); \; \text{reserve}(u,o,s); \; \ldots \; \text{no} \; \text{cancel}(u,s) \; \ldots \; \text{then can} \; \text{use}(u,o,s)\]
**Reminder: Alloy expressions in one slide**

**Set operators**
- `+ union`, `- difference`, `& intersection`, `in subset`

Slot - Taken = `{(s4)}`
holds’ - holds = `{(u2,s3)}`

**Relation operators**
- `-> product`
  - `.` `join`

**Product examples**
- `u -> s = {(u2,s3)}`
- `u -> Taken = {(u2,s1), (u2,s2), (u2,s3)}`

**Join examples**
- `u.holds’ = {(s3)}`
- `holds’.s = {(u2)}`
- `holds.Slot = {(u1)}`

**Formula examples**
- `holds’ = holds + u -> s`
  - (also written `holds += u -> s`)
- `User.holds = Taken`
- `holds in User -> Slot`

\[ a -> b = \{ (a_0,\ldots, a_n, b_0,\ldots, b_m) | (a_0,\ldots, a_n) \in a \land (b_0,\ldots, b_m) \in b \} \]
\[ a . b = \{ (a_0,\ldots, a_{n-1}, b_1,\ldots, b_m) | (a_0,\ldots, a_n) \in a \land (a_n, b_1,\ldots, b_m) \in b \} \]

**User**

\[ u: User \]

**Slot**

\[ s: Slot \]

**holds**

\[ holds: User -> Slot \]

**Taken**

\[ Taken: set Slot \]

**A relation is a table of rows**
- `holds = {(u1,s1), (u1,s2)}`
- `holds’ = {(u1,s1), (u1,s2), (u2,s3)}`

**A set is a relation with one column**
- `Slot = {(s1), (s2), (s3), (s4)}`
- `Taken = {(s1), (s2), (s3)}`

**A scalar is a set with one row**
- `u = {(u2)}`
- `s = {(s3)}`

**A table**

\[ a . b = \{ (a_0,\ldots, a_n, b_0,\ldots, b_m) | (a_0,\ldots, a_n) \in a \land (b_0,\ldots, b_m) \in b \} \]

\[ a . b = \{ (a_0,\ldots, a_{n-1}, b_1,\ldots, b_m) | (a_0,\ldots, a_n) \in a \land (a_n, b_1,\ldots, b_m) \in b \} \]