Knowing What You See
Let’s Talk About Front-end Consistency

Yifan Wu, UC Berkeley
@yifanwu

advised by Prof Joe Hellerstein and Prof Eugene Wu, worked with Larry Xu
Front-end Consistency?
Front-end Consistency?

conformity for the sake of logic, accuracy
Usually it’s fine…
Usually it’s fine...
Usually it’s fine…
You Don't See What You Expect
You Don't See What You Expect
You Don't See What You Expect
Time Diagram

- User
- Frontend
- Backend
Time Diagram
Time Diagram
Time Diagram

User

open

Frontend

Backend

Restaurants Nearby
Time Diagram

Restaurants Nearby

User

Frontend

Backend

open

get map

time
Time Diagram
Time Diagram

Restaurants Nearby

User: open

Frontend: get map

Backend

time
Time Diagram

User

Restaurants Nearby

open

get map

get pins

Frontend

Backend

time
Time Diagram
Time Diagram

Restaurants Nearby

User

open

Frontend

Backend

get map
get pins

time
Time Diagram

Restaurants Nearby

User

Frontend

Backend

open

get map

get pins

render pins
Time Diagram

User:
- open
- read

Frontend:
- get map
- get pins
- render pins

Backend:
Time Diagram
Time Diagram

Restaurants Nearby

User

Frontend

Backend

open

read

get map

get pins

render pins

render map
Time Diagram

- User
  - open
  - read
- Frontend
- Backend
- get map
- get pins
- render pins
- render map
Time Diagram

- **User**
  - open
  - read

- **Frontend**
  - get map
  - get pins
  - render pins
  - render map

- **Backend**

```
server.py
```
Time Diagram

Anomaly: Reordering
Current Poster: 11x14 Poster Print, $10.99
should i wait or refresh?
should i wait or refresh?

Anomaly: Varying Delays
Anomalies So Far

Restaurants Nearby

reordering
Anomalies So Far

reordering

varying delays
Anomalies So Far

Restaurants Nearby

reordering

varying delays
Anomalies So Far

reordering

varying delays
Anomalies So Far

reordering

varying delays
Customer Distribution by Age Group and Income

Share by Region

Sales

<table>
<thead>
<tr>
<th>Region</th>
<th>8%</th>
<th>10%</th>
<th>11%</th>
<th>29%</th>
<th>35%</th>
</tr>
</thead>
<tbody>
<tr>
<td>North</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>South</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>East</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>West</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0 10 20 30 40

http://4vector.com/i/free-vector-world-map-clip-art_104870_World_Map_clip_art_hight.png
what region does the data correspond to?
what region does the data correspond to?
what region does the data correspond to?
what region does the data correspond to?
Time Diagram

Sales

Share by Region

Customer Distribution by Age Group and Income

User

Frontend

Backend

0
25
50
75
100
April
May
June
July

7%
7%
8%
8%
10%
10%
11%
11%
29%
29%
35%
35%

0
4500
9000
10
20
30
40

0
500
1000
3000
5000
7000
9000

0
10
20
30
40

Backend

Frontend

User
Time Diagram

Sales

Share by Region

Customer Distribution by Age Group and Income

get shares
get sales
get distribution
Time Diagram

Sales

0 25 50 75 100

April May June July

Share by Region

15% 24% 16% 45%

Customer Distribution by Age Group and Income

0 10 20 30 40

Frontend

Backend

User

read

get sales

get shares

get distribution

get distribution

read
Time Diagram

- Sales
- Share by Region
- Customer Distribution by Age Group and Income
- Customer Distribution by Age Group and Income

Graphs:
- Sales (April, May, June, July)
- Share by Region
- Customer Distribution by Age Group and Income

Arrows:
- Get shares
- Get sales
- Get distribution

User, Frontend, Backend
Anomaly: Mistaken Correspondence
Wait, but it’s simple!
Why can’t we pre-fetch everything?
Why can’t we pre-fetch everything?
Why can’t we pre-fetch everything?
Why can’t we pre-fetch everything?
Why can’t we pre-fetch everything?
Why can’t we pre-fetch everything?
Why can’t we pre-fetch everything?
Why can’t we pre-fetch everything?
Why can’t we pre-fetch everything?
Functionality vs Performance
Why not just ask the backend engineers to make things fast?
Why not just ask the backend engineers to make things fast?

server compute time
Why not just ask the backend engineers to make things fast?

- server compute time
- cache miss

https://thenounproject.com/term/light-bulb/508634/
Why not just ask the backend engineers to make things fast?

- server compute time
- cache miss
- slow connection
Why not just ask the backend engineers to make things fast?

- server compute time
- cache miss
- slow connection
There are more anomalies : ( 
Real Time Interaction
Real Time Interaction
Real Time Interaction
Real Time Interaction

average dividend: 2.37%
Real Time Interaction

average dividend: 2.37%
Real Time Interaction

average dividend: 2.37%
Real Time Interaction

average dividend: 2.37%
Real Time Interaction

average dividend: 2.37%

is yellow selected?
Real Time Interaction

Anomaly: Unclear Selection

average dividend: 2.37%

is yellow selected?
Real Time Interaction
Updates **Before** Interaction
Real Time Interaction

Updates **Before** Interaction

![Graph showing data trends](image-url)
Real Time Interaction

Updates Before Interaction
Real Time Interaction
Updates Before Interaction
Real Time Interaction

Updates Before Interaction

intended to select between 25 and 55
Real Time Interaction

Updates Before Interaction

average dividend: 2.37%

intended to select between 25 and 55
Real Time Interaction

Updates **Before** Interaction

average dividend: 2.37%

intended to select between 25 and 55
Real Time Interaction
Updates Before Interaction

Anomaly: Wrong Selection

average dividend: 2.37%

intended to select between 25 and 55

Anomaly: Wrong Selection
Incremental Visualization
Incremental Visualization
Incremental Visualization

view 1

view 2
Incremental Visualization
Incremental Visualization
Incremental Visualization

view 1

view 2
Incremental Visualization

view 1

view 2
Incremental Visualization
Incremental Visualization

view 1

view 2
Incremental Visualization

Anomaly: Overwritten Marks
Incremental Visualization

Isn’t this a layout issue?

Anomaly: Overwritten Marks
Find all the bugs!

http://hyperboleandahalf.blogspot.com/
• Reordering

Find all the bugs!

http://hyperboleandahalf.blogspot.com/
• Reordering

• Varying Delays

• Mistaken Correspondence

http://hyperboleandahalf.blogspot.com/
Find all the bugs!

- Reordering
- Varying Delays
  - Mistaken Correspondence
- Concurrent Updates
  - Unclear Selection
  - Wrong Selection
  - Overwritten Marks

http://hyperboleandahalf.blogspot.com/
Consistency Definition?
Consistency Definition?

Consistency is in the eyes of the beholder.
Consistency Definition?

Consistency is in the eyes of the beholder.

Application Semantics
Consistency Definition?

Consistency is in the eyes of the beholder.

Application Semantics
Consistency Definition?

Consistency is in the **eyes of the beholder**.

Application Semantics
Consistency Definition?

Consistency is in the **eyes of the beholder**.

Application Semantics

checking
Consistency Definition?

Consistency is in the eyes of the beholder.

Application Semantics

checking
balance > 0
Consistency Definition?

Consistency is in the eyes of the beholder.

Application Semantics

checking balance > 0 credit card
Consistency Definition?

Consistency is in the eyes of the beholder.

Application Semantics

checking balance > 0

credit card balance > - $8000
What Gives?
What Gives?

- Data analysts frustration
What Gives?

- Data analysts frustration
- Engineering frustration
What Gives?

- Data analysts frustration
- Engineering frustration
- **Wrong conclusions**
What Gives?

- Data analysts frustration
- Engineering frustration
- **Wrong conclusions**
Mission Critical
Mission Critical
Mission Critical
Mission Critical
Existing Approaches
Blocking

Sales by Month

Share by Region

Customer Distribution by Age Group and Income
Blocking

- Sales by Month
- Share by Region
- Customer Distribution by Age Group and Income
Blocking

consistency

Sales by Month

Share by Region

Customer Distribution by Age Group and Income

April  May  June  July

0  20  40  60  80

0  10  20  30  40

16%  05%  24%
Blocking

- **consistency**: ✫
- **interactivity**: ✗
- **ease of understanding**: 

Sales by Month: April, May, June, July

Share by Region:
- Region A: 16%
- Region B: 24%
- Region C: 15%

Customer Distribution by Age Group and Income:

- Age Group 1: 0%
- Age Group 2: 10%
- Age Group 3: 20%
- Age Group 4: 30%
- Age Group 5: 40%
Blocking

- Consistency: ★
- Interactivity: ✗
- Ease of understanding: ★
- Ease of programming:

Sales by Month

Customer Distribution by Age Group and Income
Blocking

- consistency: ⭐
- interactivity: ✗
- ease of understanding: ⭐
- ease of programming: 🟢

Sales by Month

April: 20
May: 40
June: 60
July: 80

Customer Distribution by Age Group and Income
Spinners

Sales

April  May  June  July

Share by Region

8% 10% 29% 35%

Customer Distribution by Age Group and Income

0 20 30 40
0 4500 9000
Spinners

Sales

April | May | June | July

Share by Region

15% 24% 16% 45%

Customer Distribution by Age Group and Income

0 10 20 30 40

9000 4500

http://4vector.com/i/free-vector-world-map-clip-art_104870_World_Map_clip_art_hight.png
Spinners

Sales

Share by Region

Customer Distribution by Age Group and Income

http://4vector.com/i/free-vector-world-map-clip-art_104870_World_Map_clip_art_hight.png
Spinners, Multiple Zoom Levels
Spinners, Multiple Zoom Levels

Sales

February 75 50 25 0

April 25 50 75 100

May 50 75 100 125

June 25 50 75 100

July 0 25 50 75

Share by Region

15% 15% 16% 16% 45%

Customer Distribution by Age Group and Income

http://4vector.com/i/free-vector-world-map-clip-art_104870_World_Map_clip_art_hight.png
Spinners, Multiple Zoom Levels

Sales

April  May  June  July

Share by Region

15%  15%  16%  16%

45%  24%

Customer Distribution by Age Group and Income

http://4vector.com/i/free-vector-world-map-clip-art_104870_World_Map_clip_art_hight.png
Spinners, Multiple Zoom Levels

Sales

Share by Region

Customer Distribution by Age Group and Income

Spinners, Multiple Zoom Levels

- Consistency
- Interactivity

Sales

Share by Region

Customer Distribution by Age Group and Income
Spinners, Multiple Zoom Levels

- **Consistency**
- **Interactivity**
- **Ease of Understanding**

Sales

- April: 25
- May: 50
- June: 75
- July: 100

Share by Region

- 15%
- 16%
- 45%
- 24%

Customer Distribution by Age Group and Income

- 0
- 20
- 40
- 60
- 80
- 100

Spinners, Multiple Zoom Levels

Spinners

consistency

interactivity

ease of understanding

ease of programming

Sales

Share by Region

Customer Distribution by Age Group and Income

April | May | June | July

Spinners, Multiple Zoom Levels

Consistency

Interactivity

Ease of understanding

Ease of programming

Spinners
Labeling

Sales by Month
start lat: -85 long: -180
end lat: +85, long: +180

Share by Region
start lat: -85, long: -180
end lat: +85, long: +180

Customer Distribution by Age Group and Income
start lat: -85, long: -180
end lat: +85, long: +180
Labeling

Sales by Month
start lat: -85 long: -180
end lat: +85, long: +180

Share by Region
start lat: -85, long: -180
end lat: +85, long: +180

Customer Distribution by Age Group and Income
start lat: -85, long: -180
end lat: +85, long: +180
Labeling

Sales by Month
start lat: -85 long: -180
day lat: +85, long: +180

Share by Region
start lat: -85, long: -180
day lat: +85, long: +180

Customer Distribution
by Age Group and
Income
start lat: -85, long: -180
day lat: +85, long: +180
Labeling

Sales by Month
start lat: -85 long: -180
end lat: +85, long: +180

Share by Region
start lat: +25, long: -124
end lat: +50, long: -66

Customer Distribution by Age Group and Income
start lat: +25, long: -124
end lat: +50, long: -66
Labeling

Sales by Month
start lat: +25, long: -124
day lat: +50, long: -66

April | May | June | July

Share by Region
start lat: +25, long: -124
day lat: +50, long: -66

Customer Distribution
by Age Group and
Income
start lat: +25, long: -124
day lat: +50, long: -66
Labeling

Sales by Month
- April
- May
- June
- July

Share by Region
- 15%
- 24%
- 16%
- 45%

Customer Distribution by Age Group and Income
- Start lat: +25, long: -124
- End lat: +50, long: -66
Labeling

Customer Distribution by Age Group and Income
start lat: +25, long: -124
end lat: +50, long: -66

Share by Region
start lat: +25, long: -124
end lat: +50, long: -66

Sales by Month
start lat: +25, long: -124
end lat: +50, long: -66
Labeling

consistency

Sales by Month
- April: 20
- May: 40
- June: 60
- July: 80

Share by Region
- Start lat: +25, long: -124
- End lat: +50, long: -66

Customer Distribution by Age Group and Income
- Start lat: +25, long: -124
- End lat: +50, long: -66
Labeling

- consistency
- interactivity

Sales by Month
- April: 0
- May: 40
- June: 60
- July: 80

Share by Region
- Start lat: +25, long: -124
- End lat: +50, long: -66
- 15%
- 15%
- 16%
- 24%
- 24%
- 45%
- 0

Customer Distribution by Age Group and Income
- Start lat: +25, long: -124
- End lat: +50, long: -66
- 0
- 20
- 40
- 60
- 80
- 10000
- 20000
- 30000
- 40000

Labeling
Labeling

- consistency
- interactivity

Sales by Month
- April
- May
- June
- July

Share by Region

Customer Distribution by Age Group and Income
Labeling

<table>
<thead>
<tr>
<th></th>
<th>Labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>consistency</td>
<td>★</td>
</tr>
<tr>
<td>interactivity</td>
<td>★</td>
</tr>
<tr>
<td>ease of understanding</td>
<td>❌</td>
</tr>
<tr>
<td>ease of programming</td>
<td>❌</td>
</tr>
</tbody>
</table>

Sales by Month
- April
- May
- June
- July

Customer Distribution by Age Group and Income
- Start lat: +25, long: -124
- End lat: +50, long: -66
Unaccounted for Anomalies
Unaccounted for Anomalies

Unclear Selection

Overwritten Marks
<table>
<thead>
<tr>
<th></th>
<th>consistency</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>interactivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ease of</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>understanding</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ease of</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Design Space

<table>
<thead>
<tr>
<th>Anything Goes</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>consistency</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interactivity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ease of understanding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ease of programming</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Design Space

<table>
<thead>
<tr>
<th></th>
<th>Anything Goes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>consistency</strong></td>
<td>X</td>
</tr>
<tr>
<td><strong>interactivity</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ease of understanding</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ease of programming</strong></td>
<td></td>
</tr>
</tbody>
</table>
# Design Space

<table>
<thead>
<tr>
<th></th>
<th>Anything Goes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>consistency</strong></td>
<td>✖️</td>
</tr>
<tr>
<td><strong>interactivity</strong></td>
<td>🌟</td>
</tr>
<tr>
<td><strong>ease of understanding</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ease of programming</strong></td>
<td></td>
</tr>
</tbody>
</table>
## Design Space

<table>
<thead>
<tr>
<th></th>
<th>Anything Goes</th>
</tr>
</thead>
<tbody>
<tr>
<td>consistency</td>
<td>✗</td>
</tr>
<tr>
<td>interactivity</td>
<td>✭</td>
</tr>
<tr>
<td>ease of understanding</td>
<td>✗</td>
</tr>
<tr>
<td>ease of programming</td>
<td></td>
</tr>
</tbody>
</table>
### Design Space

<table>
<thead>
<tr>
<th></th>
<th>Anything Goes</th>
</tr>
</thead>
<tbody>
<tr>
<td>consistency</td>
<td>X</td>
</tr>
<tr>
<td>interactivity</td>
<td>⭐</td>
</tr>
<tr>
<td>ease of</td>
<td>X</td>
</tr>
<tr>
<td>understanding</td>
<td></td>
</tr>
<tr>
<td>ease of</td>
<td>⭐</td>
</tr>
<tr>
<td>programming</td>
<td></td>
</tr>
</tbody>
</table>
## Design Space

<table>
<thead>
<tr>
<th></th>
<th>Anything Goes</th>
<th>Blocking</th>
<th>Spinners</th>
<th>Labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>consistency</strong></td>
<td>❌</td>
<td>✭</td>
<td>—</td>
<td>✭</td>
</tr>
<tr>
<td><strong>interactivity</strong></td>
<td>✭</td>
<td>❌</td>
<td>—</td>
<td>✭</td>
</tr>
<tr>
<td><strong>ease of understanding</strong></td>
<td>❌</td>
<td>✭</td>
<td>—</td>
<td>❌</td>
</tr>
<tr>
<td><strong>ease of programming</strong></td>
<td>✭</td>
<td>—</td>
<td>❌</td>
<td>❌</td>
</tr>
</tbody>
</table>
front end engineering?
front end engineering?

or distributed systems?
front end engineering?

or distributed systems?

It’s all data centric programming.
front end engineering?
or distributed systems?

It’s all data centric programming.

The front-end is a database!
front end engineering?

or distributed systems?

It’s all data centric programming.

The front-end is a database!

http://hyperboleandahalf.blogspot.com/
database_techniques(reordering) = consistency & declarativity
Dealing with Inconsistency

- Detection: formal framework
- Prevention:
  - Ensure consistency
  - Communicate inconsistency
A Relational Model

view 1
A Relational Model
A Relational Model

view 1

<table>
<thead>
<tr>
<th>mouse event</th>
<th>event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>down</td>
<td>50</td>
<td>95</td>
</tr>
</tbody>
</table>
A Relational Model

view 1

mouse event

<table>
<thead>
<tr>
<th>event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>50</td>
<td>95</td>
</tr>
</tbody>
</table>
A Relational Model

Mouse event:
- event: x, y
- down: 50, 95
- move: 55, 150
A Relational Model

<table>
<thead>
<tr>
<th>Mouse Event</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>Move</td>
<td>55</td>
<td>150</td>
</tr>
</tbody>
</table>
A Relational Model

<table>
<thead>
<tr>
<th>mouse event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>move</td>
<td>55</td>
<td>150</td>
</tr>
<tr>
<td>up</td>
<td>60</td>
<td>150</td>
</tr>
</tbody>
</table>
A Relational Model

Mouse event

<table>
<thead>
<tr>
<th>Event</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>move</td>
<td>55</td>
<td>150</td>
</tr>
<tr>
<td>up</td>
<td>60</td>
<td>150</td>
</tr>
</tbody>
</table>
A Relational Model

<table>
<thead>
<tr>
<th>event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>move</td>
<td>55</td>
<td>150</td>
</tr>
<tr>
<td>up</td>
<td>60</td>
<td>150</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>view1_rects @ t1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>60</td>
</tr>
<tr>
<td>90</td>
</tr>
</tbody>
</table>
A Relational Model

<table>
<thead>
<tr>
<th>event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>move</td>
<td>55</td>
<td>150</td>
</tr>
<tr>
<td>up</td>
<td>60</td>
<td>150</td>
</tr>
</tbody>
</table>

```
view1_rects @ t1

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>width</th>
<th>height</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>150</td>
<td>5</td>
<td>20</td>
<td>blue</td>
</tr>
<tr>
<td>60</td>
<td>150</td>
<td>5</td>
<td>30</td>
<td>blue</td>
</tr>
<tr>
<td>90</td>
<td>150</td>
<td>5</td>
<td>60</td>
<td>blue</td>
</tr>
</tbody>
</table>
```
A Relational Model

mouse event

view 1

down
move
up

view1_rects @ t1

base data

<table>
<thead>
<tr>
<th>city</th>
<th>stores</th>
<th>sales</th>
<th>reps</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>50</td>
<td>800</td>
<td>5</td>
</tr>
<tr>
<td>NYC</td>
<td>30</td>
<td>300</td>
<td>2</td>
</tr>
<tr>
<td>LA</td>
<td>5</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>BOS</td>
<td>1</td>
<td>50</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>width</th>
<th>height</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>95</td>
<td>5</td>
<td>20</td>
<td>blue</td>
</tr>
<tr>
<td>60</td>
<td>150</td>
<td>5</td>
<td>30</td>
<td>blue</td>
</tr>
<tr>
<td>90</td>
<td>150</td>
<td>5</td>
<td>60</td>
<td>blue</td>
</tr>
</tbody>
</table>
A Relational Model

mouse event

down
move
up

event x y
50 95
55 150
60 150

view1_rects @ t1
x y width height color
50 150 5 20 blue
60 150 5 30 blue
90 150 5 60 blue

view2_rects @ t2
cx cy r color
160 90 5 orange
130 140 2 blue

base data
.city stores sales reps
SF 50 800 5
NYC 30 300 2
LA 5 100 1
BOS 1 50 1
A Relational Model

Base data

<table>
<thead>
<tr>
<th>city</th>
<th>stores</th>
<th>sales</th>
<th>reps</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>50</td>
<td>800</td>
<td>5</td>
</tr>
<tr>
<td>NYC</td>
<td>30</td>
<td>300</td>
<td>2</td>
</tr>
<tr>
<td>LA</td>
<td>5</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>BOS</td>
<td>1</td>
<td>50</td>
<td>1</td>
</tr>
</tbody>
</table>

Mouse event

<table>
<thead>
<tr>
<th>event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>move</td>
<td>55</td>
<td>150</td>
</tr>
<tr>
<td>up</td>
<td>60</td>
<td>150</td>
</tr>
</tbody>
</table>

View1 rects @ t1

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>width</th>
<th>height</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>150</td>
<td>5</td>
<td>20</td>
<td>blue</td>
</tr>
<tr>
<td>60</td>
<td>150</td>
<td>5</td>
<td>30</td>
<td>blue</td>
</tr>
<tr>
<td>90</td>
<td>150</td>
<td>5</td>
<td>60</td>
<td>blue</td>
</tr>
</tbody>
</table>

View2 rects @ t2

<table>
<thead>
<tr>
<th>cx</th>
<th>cy</th>
<th>r</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>160</td>
<td>90</td>
<td>5</td>
<td>orange</td>
</tr>
<tr>
<td>130</td>
<td>140</td>
<td>2</td>
<td>blue</td>
</tr>
</tbody>
</table>
Interaction as a Transaction
Interaction as a Transaction

- Interpreting interaction is a **read**.
Interaction as a Transaction

• Interpreting interaction is a read.
Interaction as a Transaction

- Interpreting interaction is a **read**.
- Rendering the data is a **write**.
Interaction as a Transaction

- Interpreting interaction is a **read**.
- Rendering the data is a **write**.
Data Flow
Data Flow

interaction
Data Flow
Data Flow
Data Flow

interaction -> R

request -> result
Data Flow
Data Flow

interaction → render data → request → result
Data Flow

interaction
confirmation
render data

request
result

R
W
Data Flow

interaction -> confirmation <- render data

R W W

request -> result

Databases
Conflict Example

Pixel data:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>color</th>
<th>lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>70</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>90</td>
<td>140</td>
<td>white</td>
<td>init</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Conflict Example

current_interaction: I1

pixel data

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>color</th>
<th>lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>70</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>90</td>
<td>140</td>
<td>white</td>
<td>init</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Conflict Example

current_interaction: I1

diagram:

view 1

view 2

pixel data:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>color</th>
<th>lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>70</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>90</td>
<td>140</td>
<td>white</td>
<td>init</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Conflict Example

current_interaction: I1 I2

pixel data

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>color</th>
<th>lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>70</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>90</td>
<td>140</td>
<td>white</td>
<td>init</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Conflict Example

current_interaction: I1 I2

pixel data

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>color</th>
<th>lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>blue</td>
<td>I2, V1, T1</td>
</tr>
<tr>
<td>70</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>90</td>
<td>140</td>
<td>white</td>
<td>init</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Conflict Example

current_interaction: I1 I2

![Diagram showing pixel data and view interactions]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>color</th>
<th>lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>blue</td>
<td>I2, V1, T1</td>
</tr>
<tr>
<td>70</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>90</td>
<td>140</td>
<td>white</td>
<td>init</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Conflicting Example

current_interaction: I1 I2

pixel data

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>color</th>
<th>lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>70</td>
<td>140</td>
<td>blue</td>
<td>I2, V1, T1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>90</td>
<td>140</td>
<td>white</td>
<td>init</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Defining Conflict: Write-Write
Defining Conflict: Write-Write

Two interaction attempting to write to the same pixel during their executions.
Defining Conflict: Write-Write

Two interaction attempting to write to the same pixel during their executions.
Defining Conflict: Write-Write

Two interaction attempting to write to the same pixel during their executions.
Defining Conflict: Read-Write
Defining Conflict: Read-Write
Defining Conflict: Read-Write
Defining Conflict: Read-Write
Defining Conflict: Read-Write
Defining Conflict: Read-Write

<table>
<thead>
<tr>
<th>Event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>move</td>
<td>55</td>
<td>150</td>
</tr>
<tr>
<td>up</td>
<td>60</td>
<td>150</td>
</tr>
</tbody>
</table>
Defining Conflict: Read-Write

<table>
<thead>
<tr>
<th>mouse event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>down</td>
<td>55</td>
<td>150</td>
</tr>
<tr>
<td>move</td>
<td>60</td>
<td>150</td>
</tr>
</tbody>
</table>
Defining Conflict: Read-Write

<table>
<thead>
<tr>
<th>mouse event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>down</td>
<td>55</td>
<td>150</td>
</tr>
<tr>
<td>move</td>
<td>60</td>
<td>150</td>
</tr>
<tr>
<td>up</td>
<td>160</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>130</td>
<td>140</td>
</tr>
</tbody>
</table>
Defining Conflict: Read-Write

Mouse event:
- Event: x = 50, y = 95
- Move: x = 55, y = 150
- Up: x = 60, y = 150

Lines @ t1:
- Line 1: x = 160, y = 90
- Line 2: x = 130, y = 140
Defining Conflict: Read-Write

- **mouse event**
  - event
  - down: x=50, y=95
  - move: x=55, y=150
  - up: x=60, y=150

- **lines @ t1**
  - x | y
    - 160 | 90
    - 130 | 140
Defining Conflict: Read-Write

- Mouse event:
  - Event: x, y
  - Move: 55, 150
  - Up: 60, 150

- Base data:
  - City: SF, stores: 50
  - City: NYC, stores: 30
  - City: LA, stores: 5

- Lines @ t1:
  - x, y
    - 160, 90
    - 130, 140
Defining Conflict: Read-Write

mouse event

<table>
<thead>
<tr>
<th>event</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>50</td>
<td>95</td>
</tr>
<tr>
<td>move</td>
<td>55</td>
<td>150</td>
</tr>
<tr>
<td>up</td>
<td>60</td>
<td>150</td>
</tr>
</tbody>
</table>

base data

<table>
<thead>
<tr>
<th>city</th>
<th>stores</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>50</td>
</tr>
<tr>
<td>NYC</td>
<td>30</td>
</tr>
<tr>
<td>LA</td>
<td>5</td>
</tr>
</tbody>
</table>

lines @ t1

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>160</td>
<td>90</td>
</tr>
<tr>
<td>130</td>
<td>140</td>
</tr>
</tbody>
</table>

lines @ t2

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>90</td>
</tr>
<tr>
<td>300</td>
<td>140</td>
</tr>
<tr>
<td>130</td>
<td>140</td>
</tr>
</tbody>
</table>
Defining Conflict: Read-Write

mouse event

event | x | y
--- | --- | ---
down | 50 | 95
move | 55 | 150
up   | 60 | 150

lines @ t2

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>130</td>
<td>140</td>
</tr>
<tr>
<td>200</td>
<td>90</td>
</tr>
<tr>
<td>30</td>
<td>140</td>
</tr>
</tbody>
</table>

base data

city | stores
--- | ---
SF   | 50
NYC  | 30
LA   | 5

25

50
Defining Conflict: *Write-Read

*human latency: ~250ms
Defining Conflict: *Write-Read

*human latency: ~250ms
Defining Conflict: *Write-Read

*human latency: ~250ms
Defining Conflict: *Write-Read

*human latency: ~250ms
Defining Conflict: *Write-Read

*human latency: ~250ms
Dealing with Inconsistency

- Detection: formal framework
- Prevention:
  - Ensure consistency
  - Communicate inconsistency
Seen Technique: Blocking
Seen Technique: Blocking

Inspiration from systems algorithms!
Seen Technique: Blocking

Inspiration from systems algorithms!
Can we draw more insights from algorithms for systems concurrency?
Interaction Constraints

I.pins.render IF I.maps.render == TRUE
Interaction Constraints

I.pins.render IF I.maps.render == TRUE
Merge Functions

view 1

view 2

pixel data

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>color</th>
<th>lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>70</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>90</td>
<td>140</td>
<td>white</td>
<td>init</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
**Merge Functions**

merge_func(p1,p2): [p1,p2].map(
    p -> p.lineage.mark.transparency = 0.5)

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>color</th>
<th>lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>70</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>140</td>
<td>orange</td>
<td>I1, V2, T2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>90</td>
<td>140</td>
<td>white</td>
<td>init</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
merge_func(p1,p2): [p1,p2].map(
p  ->  p.lineage.mark.transparency = 0.5)
**Merge Functions**

merge_func(p1,p2): [p1,p2].map(
  p -> p.lineage.mark.transparency = 0.5)
merge_func(p1,p2): [p1,p2].map(
  p -> p.lineage.mark.transparency = 0.5)
Visualization Snapshots

Multi Version Concurrency Control
Visualization Snapshots

Multi Version Concurrency Control
Visualization Snapshots
Example: Cross Filter

http://square.github.io/crossfilter/
Visualization Snapshots
Example: Cross Filter

http://square.github.io/crossfilter/
Visualization Snapshots
Example: Cross Filter

http://square.github.io/crossfilter/
Visualization Snapshots
Example: Cross Filter

http://square.github.io/crossfilter/
Visualization Snapshots
Example: Cross Filter

scroll

http://square.github.io/crossfilter/
Dealing with Inconsistency

- Detection: formal framework
- Prevention:
  - Ensure consistency
  - Communicate inconsistency
Dealing with Inconsistency

- Detection: formal framework
- Prevention:
  - Ensure consistency
  - Communicate inconsistency
labels: data from <selection description>
No Callback Hell Please!
No Callback Hell Please!

labels: data from <selection description>

know the global state!
Dealing with Inconsistency

**Declaratively**
API for Front-End State
Shim Layer for Consistency

API for Front-End State
Shim Layer for Consistency

API for Front-End State

interaction & request handlers
Shim Layer for Consistency

API for Front-End State

conflict detector

interaction & request handlers
Shim Layer for Consistency

API for Front-End State

Conflict detector

Prevent

Interaction & request handlers
conflict detector

Shim Layer for Consistency

communicate    prevent

conflict detector

Shim Layer for Consistency

API for Front-End State

interaction & request handlers
the front-end is a database
Related Work

- **Paper**: A DeVIL-ish approach to inconsistency in interactive visualizations.

- Prof Eugene Wu’s [Data Visualization Management Systems](#).

- Prof. Joe Hellerstein and Peter Alvaro’s work on [Bloom](#).
Thanks!

@yifanwu

to share your comments!

or get slides, references, and project updates!