Live Visualization and Editing of User Behavior in iObserve

Daniel Banck

Friday 31\textsuperscript{st} March, 2017

se...
1. Introduction

2. Foundations

3. Approach

4. Evaluation

5. Conclusion and Future Work
Motivation

Introduction

- Comprehend how users are interacting with an application
- Find bottlenecks and evolve the system
- Model future user behavior
Goals

Introduction

- Evaluation of Technologies for Live Visualization of User Behavior
- Implementation of User Behavior Visualization in iObserve
  - Live Visualization of User Behavior
  - Editing of User Behavior
iObserve
Live Visualization
Reactive Programming
More technologies
Live Visualization and Editing of User Behavior in iObserve

Daniel Banck

Architecture

Approach

Browser / Frontend

Editor

Visualization

Data Collection and Update Component

Visualization Update Push Component

Graph Database

Mock Service

Operator

Data flow

Control flow
Live Visualization and Editing of User Behavior in iObserve
Live Visualization and Editing of User Behavior in iObserve

Friday 31st March, 2017
Data Model

Approach

Entity

UserBehaviorGraph

Page

Visit

pages 0..* visits 0..*

start

end

Daniel Banck

Live Visualization and Editing of User Behavior

Friday 31st March, 2017
Data Model Comparison

Approach

UserBehaviorGraph

Entity

Page

Visit

pages

0..*

visits

0..*

start

end

Daniel Banck

Live Visualization and Editing of User Behavior

Christian-Albrechts-Universität zu Kiel

Technische Fakultät

Friday 31st March, 2017
Evaluation
Evaluation Questions

Evaluation ⊳ Javascript Libraries for Building User Interfaces

▶ How well is the library maintained?
▶ To what extend does the library foster low complexity component data binding?
▶ To what extend does the library support interoperability with other libraries?
Elm

Evaluation Javascript Libraries for Building User Interfaces

Programming Language by Evan Czaplicki

Compiles to Javascript

Applications should follow the Elm Architecture

Strictly typed

import Html exposing (..)

--- MODEL
type alias Model = { ... }

--- UPDATE
type Msg = Reset | ...

update : Msg -> Model -> Model
update msg model =
case msg of
  Reset -> ...
  ...

--- VIEW
view : Model -> Html Msg
view model =
  ...
React

- Javascript library by Facebook
- For building User Interfaces
- Declarative
- Component-Based
- With optional XML-like syntax called JSX

```javascript
class ButtonComponent extends React.Component {

  constructor(props) {
    super(props);
    state = {count: 0};
  }

  increase() {
    setState({
      count: state.count + 2
    });
  }

  render() {
    <button onClick={this.increase}>
      Increase
    </button>
  }
}
```
# How well is the library maintained?

---

<table>
<thead>
<tr>
<th>Metric</th>
<th>Elm</th>
<th>React</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributors on Github</td>
<td>86</td>
<td>956</td>
</tr>
<tr>
<td>Project age</td>
<td>2012</td>
<td>March 2013</td>
</tr>
<tr>
<td>Downloads on NPM in the last month</td>
<td>35,421</td>
<td>3,382,322</td>
</tr>
<tr>
<td>What kind of release scheme is used?</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>When was the last stable release?</td>
<td>January 23, 2017</td>
<td>January 6, 2017</td>
</tr>
<tr>
<td>Does it follow a versioning scheme?</td>
<td>Semantic Versioning</td>
<td>Semantic Versioning</td>
</tr>
<tr>
<td>How high is the test coverage?</td>
<td>-</td>
<td>82%</td>
</tr>
<tr>
<td>How stable is the API?</td>
<td>Unstable</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

---

Daniel Banck  
Live Visualization and Editing of User Behavior  
Friday 31st March, 2017  
14 / 20
### Low complexity component data binding?

**Evaluation**  
**Javascript Libraries for Building User Interfaces**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Elm</th>
<th>React</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does it support push updates?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>What is the render performance?</td>
<td>2244ms</td>
<td>3553ms</td>
</tr>
</tbody>
</table>
### Interoperability with other libraries?

<table>
<thead>
<tr>
<th>Metric</th>
<th>Elm</th>
<th>React</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is it compatible with Cytoscape.js?</td>
<td>Might work via a Javascript bridge</td>
<td>Yes</td>
</tr>
<tr>
<td>Is there a wrapper for styling via Bootstrap?</td>
<td>Yes, elm-bootstrap-html</td>
<td>Yes, reactstrap</td>
</tr>
<tr>
<td>Is there a library for handling WebSockets?</td>
<td>Yes, websocket</td>
<td>Yes, socket.io</td>
</tr>
</tbody>
</table>
Flow

- Static type checker for Javascript
- Static type annotations
- Type inference
- Third-party library interface definitions

```javascript
// @flow
function square (n: number): number {
  return n * n;
}

square("2", "2"); // Error!

// @flow
function identity <T>(value: T): T {
  return value;
}

type Colors = 'red' | 'blue';
```
Conclusion and Future Work

- Visualization of user behavior models
  - Interactive graph
  - Live updates
- Editor for user behavior models
- Implementations are open source in the iObserve research project
Future Work

Conclusion and Future Work

- Integrate into iObserve
- Handling of fast incoming data
  - Batch updates
  - Less re-rendering of the graph
- Add authentication and authorization
Conclusion

- Visualization of user behavior models
- Editor for user behavior models

Future Work

- Integrate into iObserve
- Handling of fast incoming data
- Add authentication and authorization

Source on Github:
https://github.com/research-iobserve/ubm-visualization