Live Visualization and Editing of User Behavior in iObserve

Daniel Banck

Friday 31st March, 2017

Outline

Motivation

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

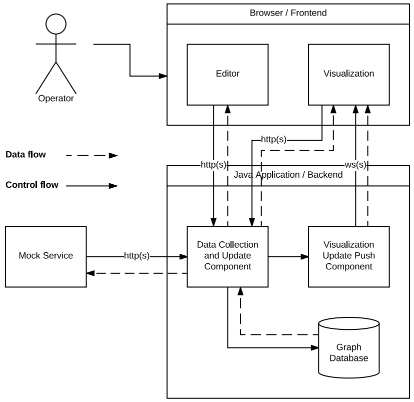
Goals

- 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

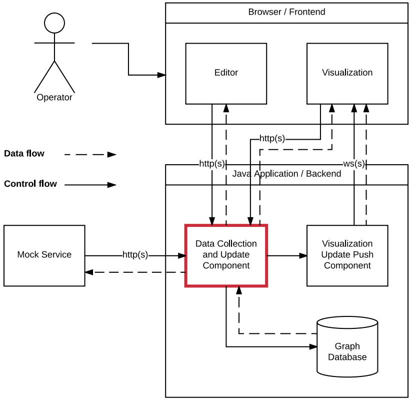
Foundations

- iObserve
- Live Visualization
- Reactive Programming
- More technologies

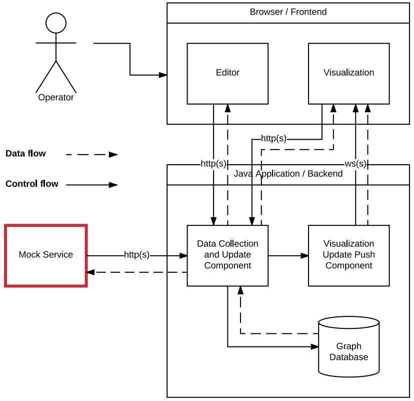
Architecture



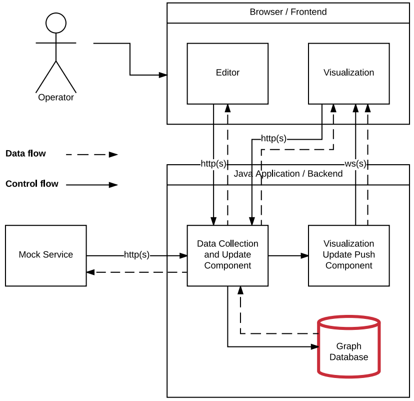
Architecture



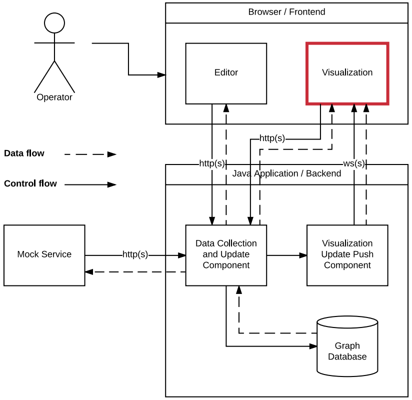
Architecture



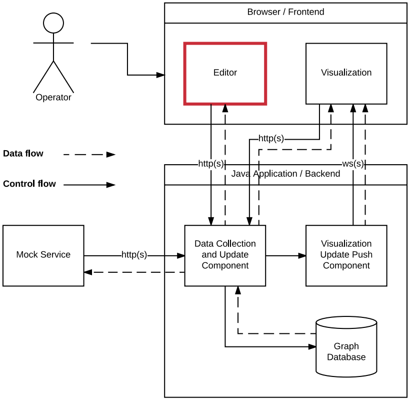
Architecture



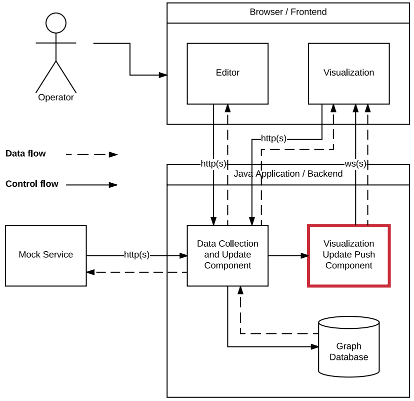
Architecture



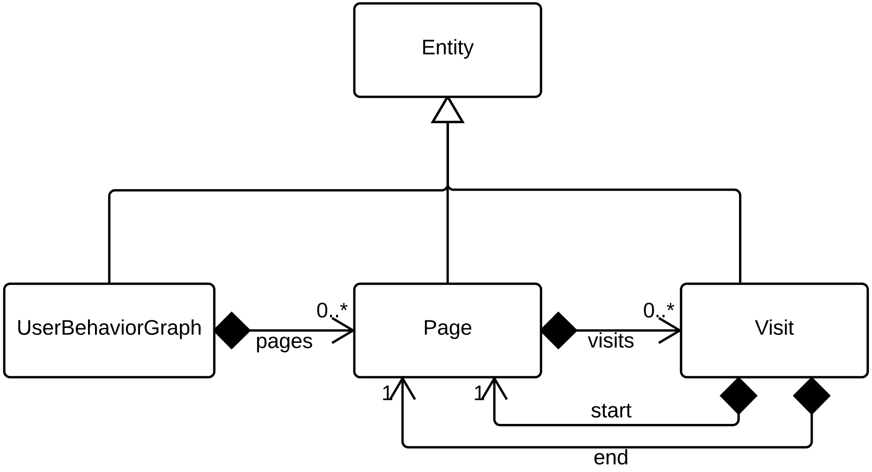
Architecture



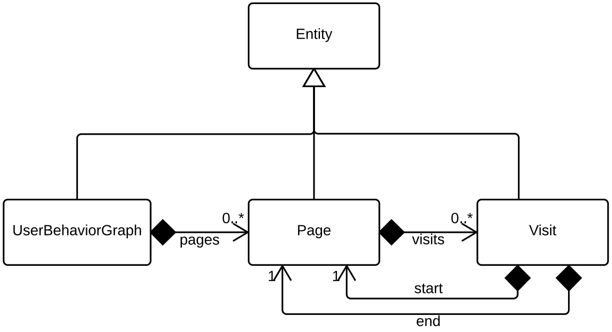
Architecture

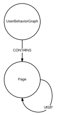


Data Model


Entity
UserBehaviorGraph
Page
Visit
0..*
pages
0..*
visits
1
1
start
end

Data Model Comparison





Demo

Demo

Evaluation

Evaluation Questions

- 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

- 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

```
class ButtonComponent extends
    React.Component {

  constructor(props) {
    super(props);
    state = {count: 0};
  }
  increase() {
    setState({
      count: state.count + 2
    });
  }
  render() {
    <button onClick={increase}>
      Increase
    </button>
  }
}
```

How well is the library maintained?

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

Low complexity component data binding?

Metric	**Elm**	**React**
Does it support push updates?	Yes	Yes
What is the render performance?	2244ms	3553ms

Interoperability with other libraries?

Metric	**Elm**	**React**
Is it compatible with Cytoscape.js?	Might work via a Javascript bridge	Yes
Is there a wrapper for styling via Bootstrap?	Yes, elm-bootstrap-html	Yes, reactstrap
Is there a library for handling WebSockets?	Yes, websocket	Yes, socket.io

Type checking Javascript

Flow

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

```
// @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

- 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

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

Summary

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