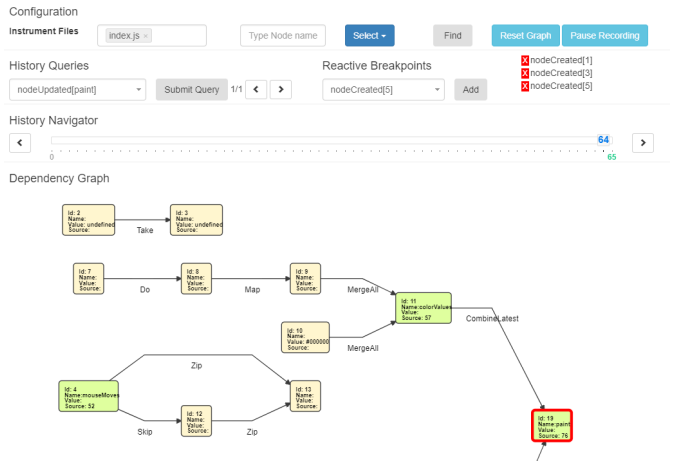
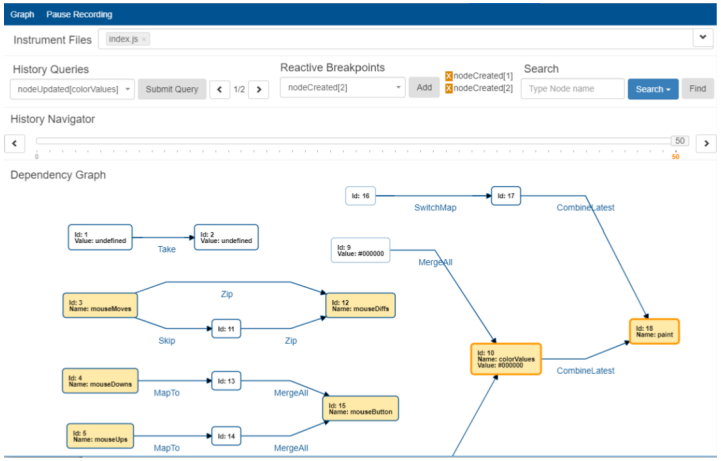
Advancing the Chrome Reactive Inspector

Bachelor-Thesis by Benedikt Gross
Supervisor Prof.Dr.Guido Salvaneschi

TECHNISCHE UNIVERSITÄT DARMSTADT





Reactive Programming

TECHNISCHE UNIVERSITÄT DARMSTADT

- multiple values over time
- observables
- event streams
- Implementations in all major programming languages

ReactiveX
source: http://reactivex.io/

Bacon.js
source: https://baconjs.github.io/

Traditional Debugging

TECHNISCHE UNIVERSITÄT DARMSTADT

- breakpoints
- step by step execution
- event logging
- stack trace inspection
- *printf-debugging*

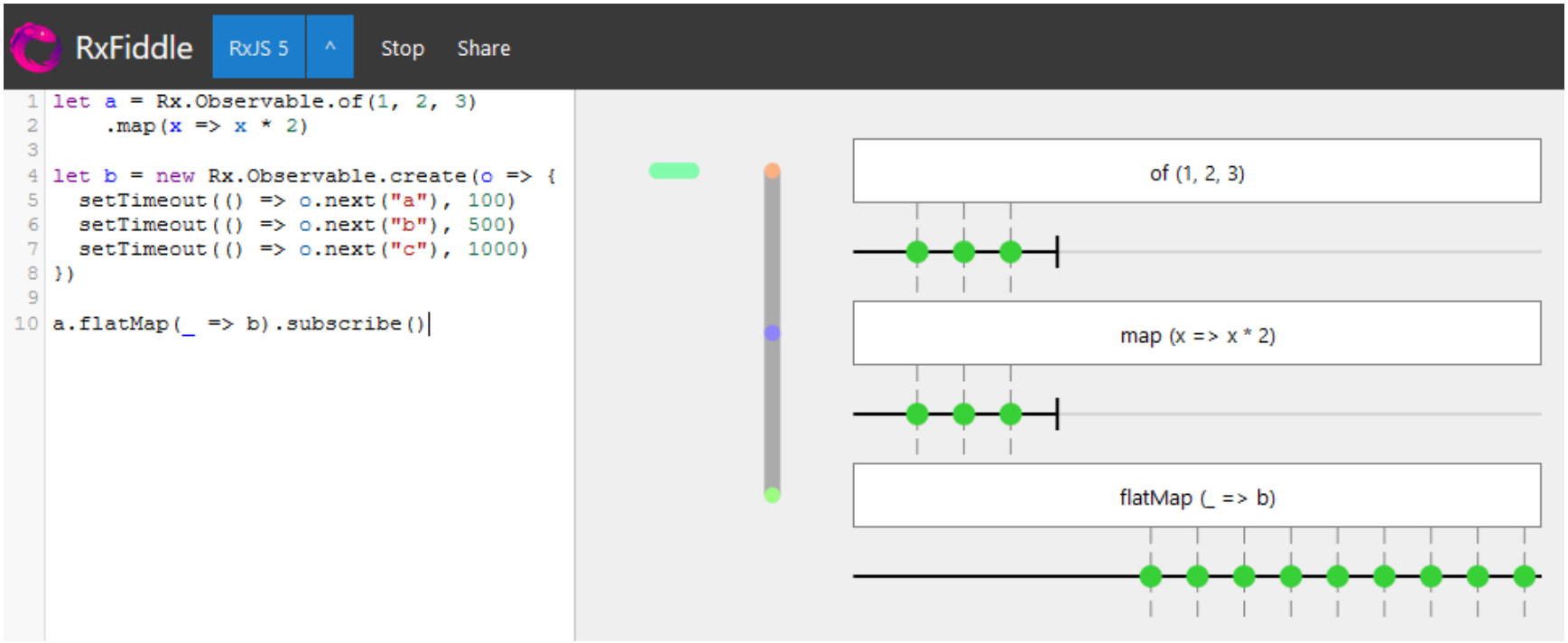
Debuggin Reactive Code

TECHNISCHE UNIVERSITÄT DARMSTADT

- do-debugging
- abstraction of dependencies
- limited number of tools
- Reactive Inspector for Scala
- Chrome Reactive Inspector
- RxFiddle

Debuggin Reactive Code - RxFiddle

TECHNISCHE UNIVERSITÄT DARMSTADT

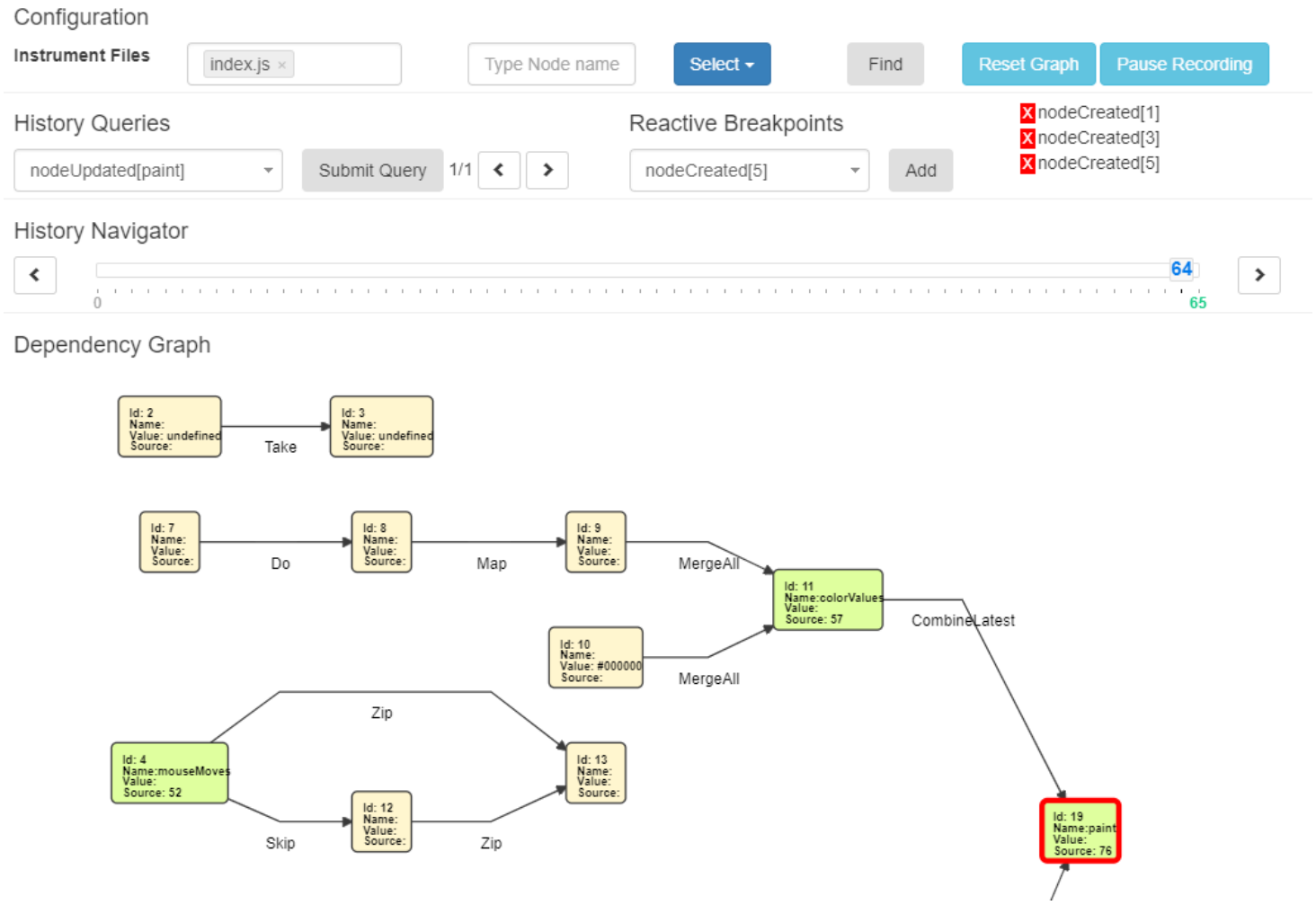


Screenshot of the RxFiddle Demo

Source: https://rxfiddle.net/

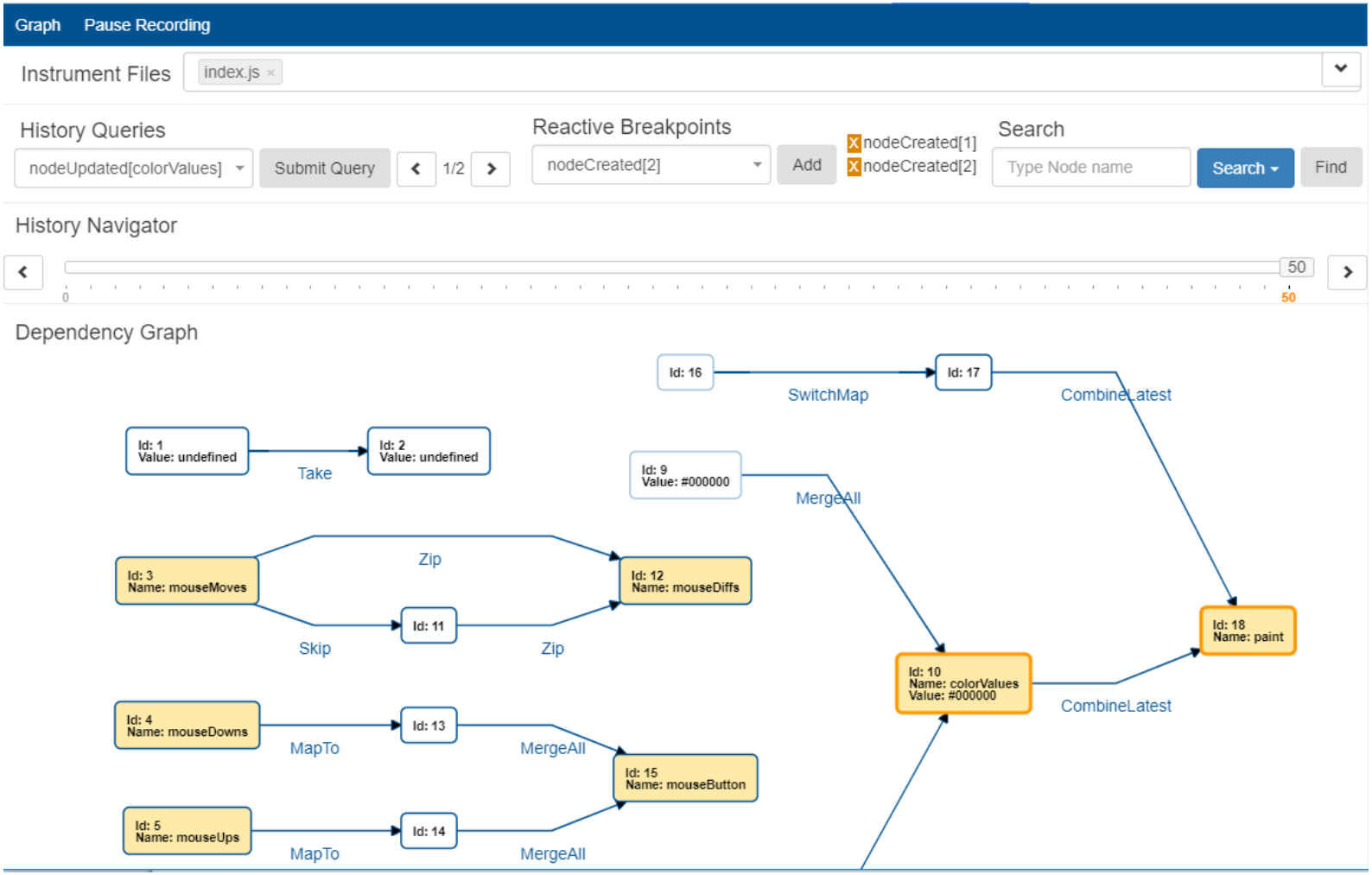
The Chrome Reactive Inspector

The previous User Interface



The new User Interface

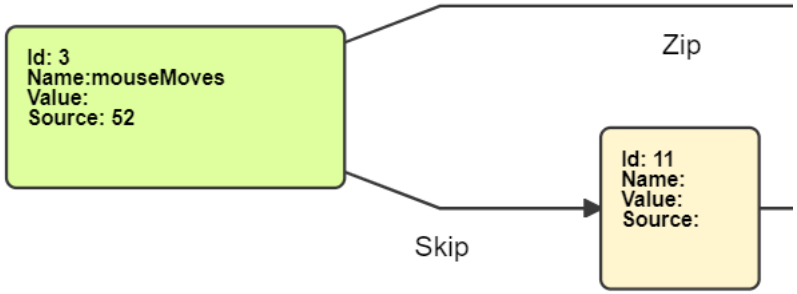
TECHNISCHE UNIVERSITÄT DARMSTADT

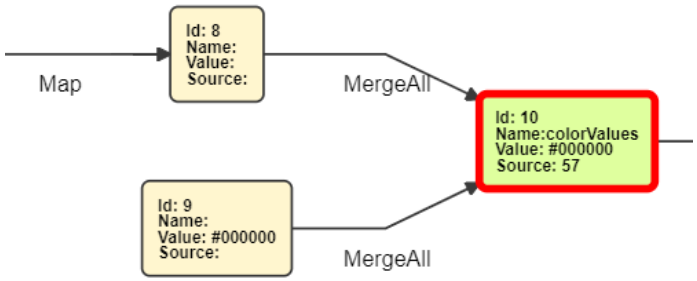


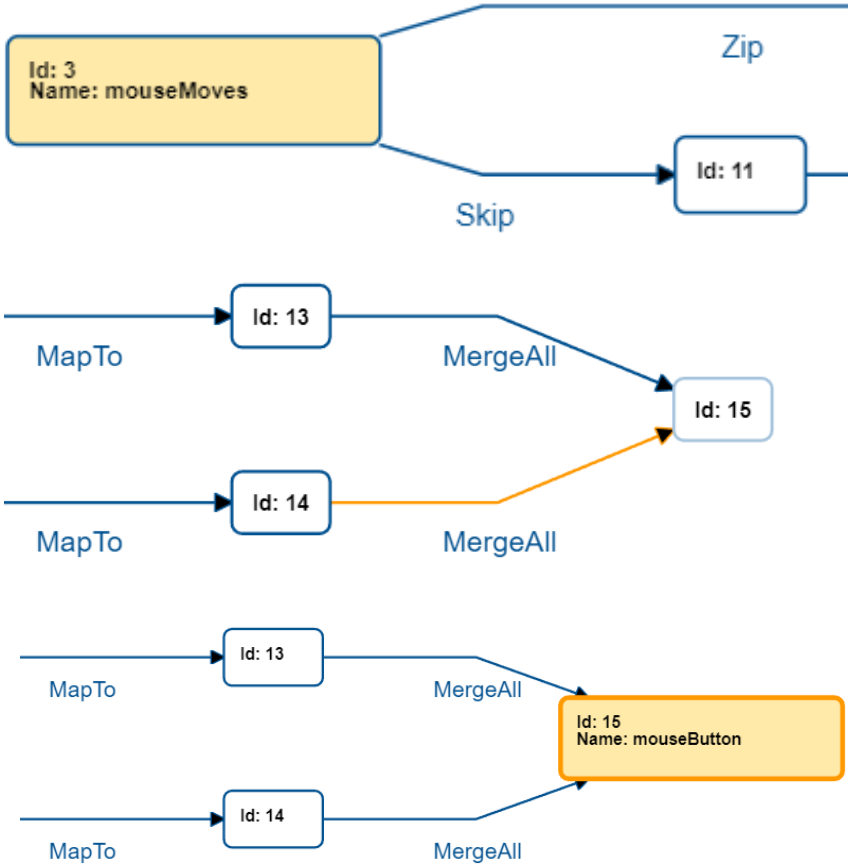
Advancing the User Interface

Reducing the cognitive load

TECHNISCHE UNIVERSITÄT DARMSTADT



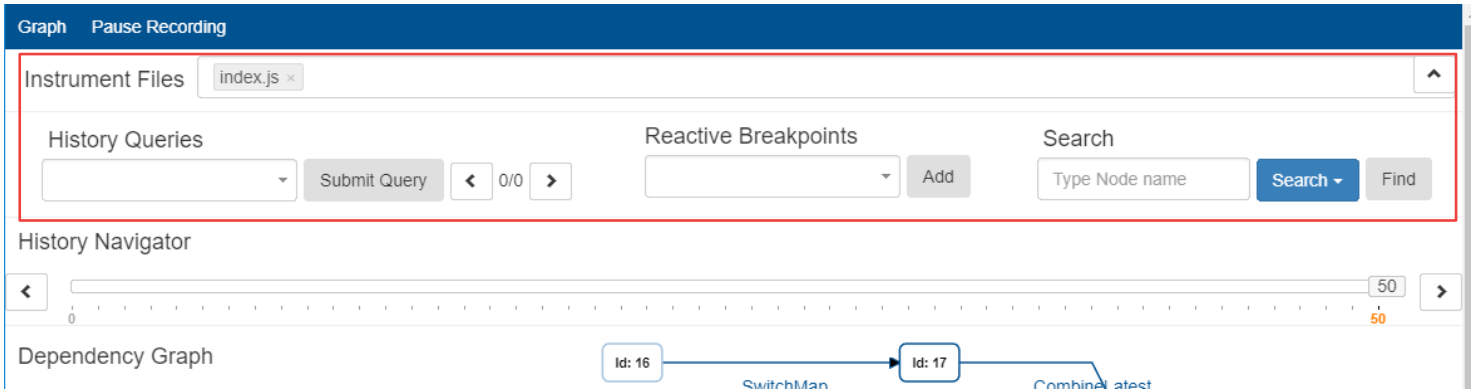


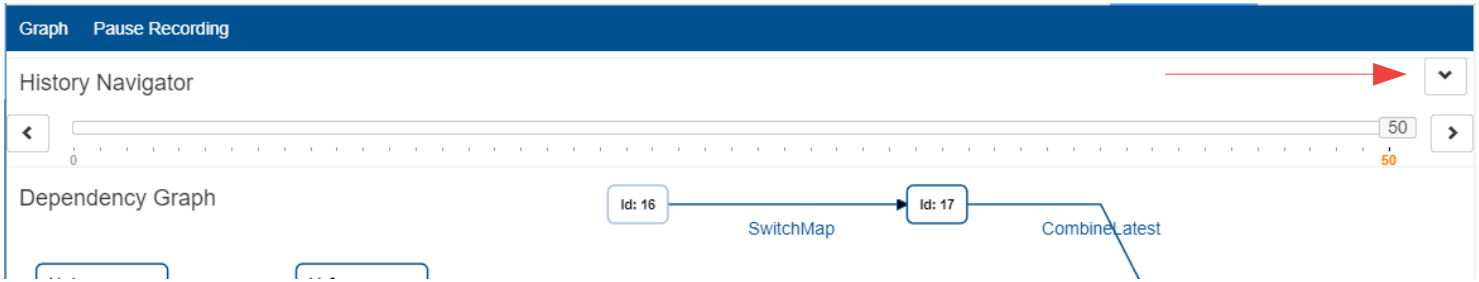


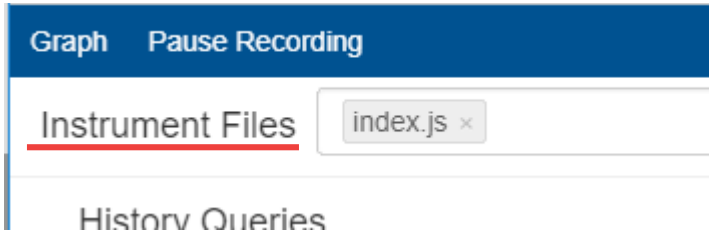
Advancing the User Interface

Reducing the cognitive load

TECHNISCHE UNIVERSITÄT DARMSTADT







Connecting abstract graph with JavaScript code

TECHNISCHE UNIVERSITÄT DARMSTADT

- provide additional context
- merge both views

Method Chaining:

```
var intervalObservable = Rx.Observable.interval(5)
    .timestamp()
    .bufferCount(2, 1)
    .map(function (w) {
        return w[1].timestamp - w[0].timestamp;
    })
    .share();
```

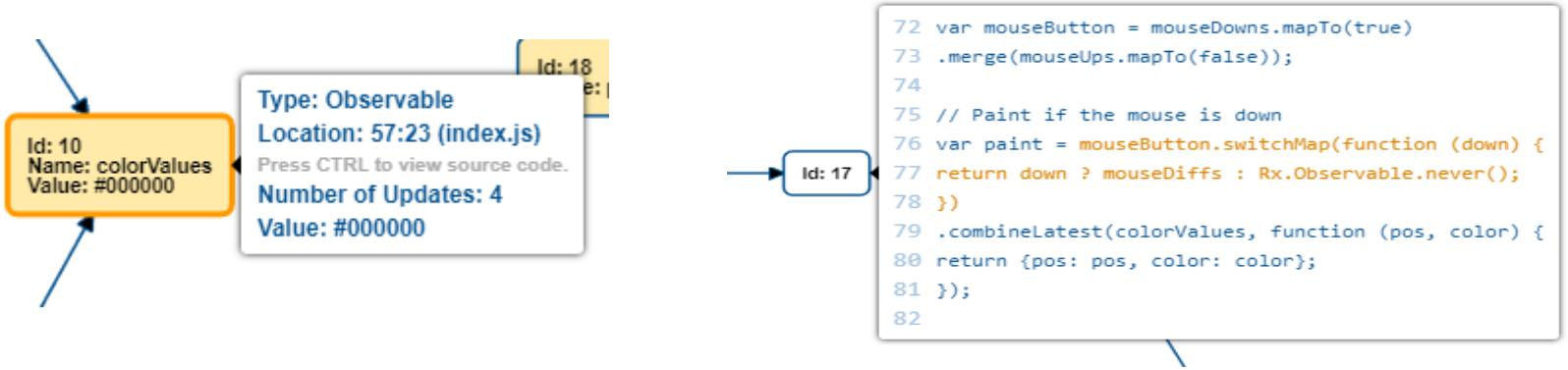
Connecting abstract graph with JavaScript code

TECHNISCHE UNIVERSITÄT DARMSTADT

- difficulty: instrumented JavaScript code

```
$sonWallet = J$.W(593, '$sonWallet', J$.F(585, J$.I(typeof $ === 'undefined'
    ? $ = J$.R(569, '$', undefined, true, true) : $ = J$.R(569, '$', $, true,
     true)), false)(J$.T(577, '#wallet-son', 21, false)), J$.I(typeof
```

- covers *almost* all nodes



Connecting abstract graph with JavaScript code

TECHNISCHE UNIVERSITÄT DARMSTADT

- *Source Code Tooltips*

  Pro

  - compliments dependency graph
  - easily accessible
  - highlight code
  - existing workflow

  Con

  - only snippets
  - no search feature
  - limited when using references to functions

Function reference example:

```
function createFunction(add) {
  return (value) => value + add;
}
let fatherWalletValue = sonWalletValue
  .map(createFunction(10));
```

Rapidly updated Observables

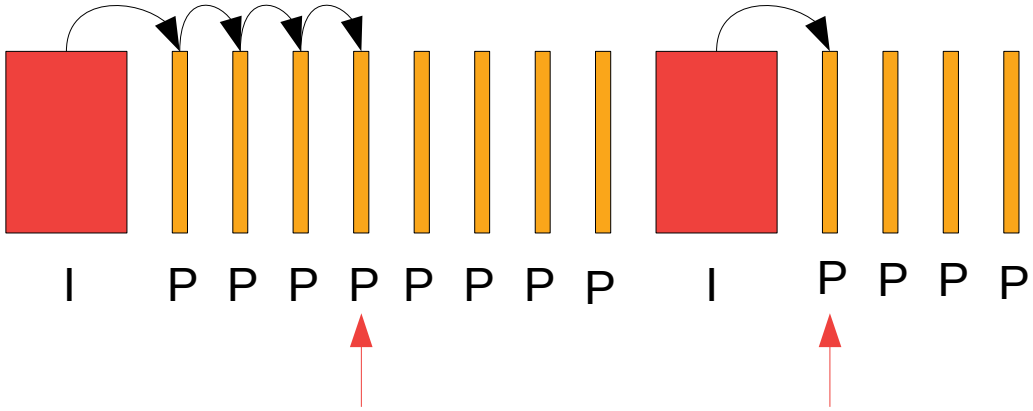
TECHNISCHE UNIVERSITÄT DARMSTADT

- timers, mouse movement, network traffic
- generate thousands of steps
- limited by recording performance
- may render CRI useless

Rapidly updated Observables
improving performance

TECHNISCHE UNIVERSITÄT DARMSTADT

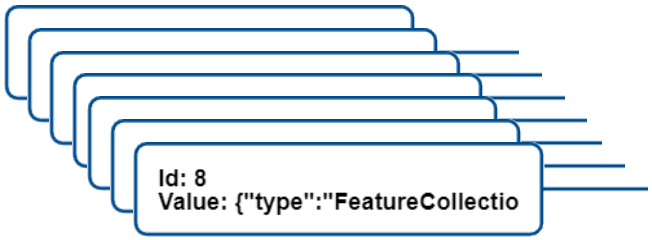
- UI update throttling
- (software) Paging
- Delta Encoding
- improved similar to Video Compression



Excessively created Observables

TECHNISCHE UNIVERSITÄT DARMSTADT

- describes a group of observables
- created as part of a loop
- created multiple times temporarily
- all originating from the same code
- require detection and representation in the graph



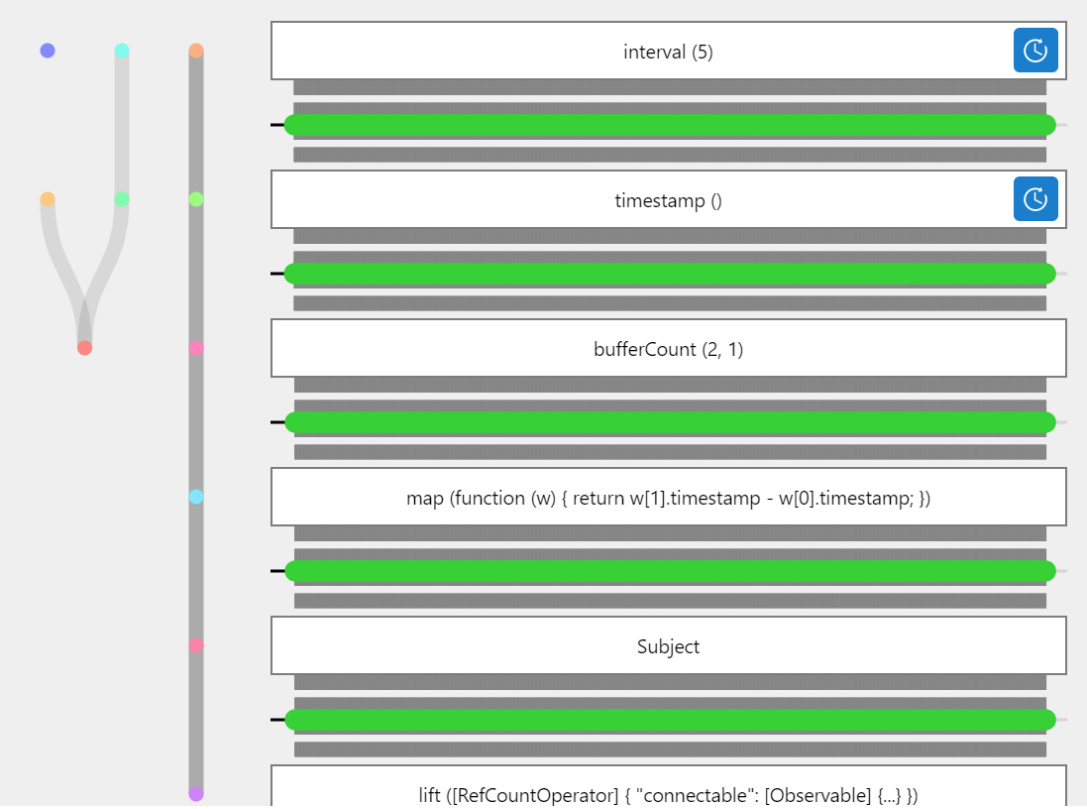
Evaluation

TECHNISCHE UNIVERSITÄT DARMSTADT

- changes to the UI
- Source Code Tooltips cover 75% of all nodes
- CPU Performance
  - Version 2: 154s
  - Version 3: 5.5s
- Memory
  - Version 2: 54mb - depending on steps
  - Version 3: 15mb - constant

Evaluation
Comparison to RxFiddle

TECHNISCHE UNIVERSITÄT DARMSTADT



Screenshot of RxFiddle with the test application

Evaluation
Test Applications

TECHNISCHE UNIVERSITÄT DARMSTADT

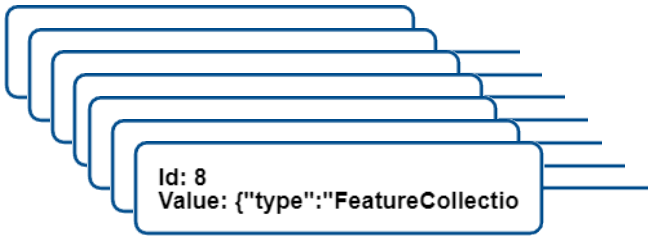
- verify robustness
- state of the project
- 9 specifications tested for each applicaiton
- 388 of 420 checks were successful
- notable findings:
  - Ambiguity
  - JavaScript in HTML attributes
  - breakdown of recording for a specific pattern

Future

TECHNISCHE UNIVERSITÄT DARMSTADT

- module loaders
- ES6 and TypeScript support
- extend node exclusion
- excessively created observables

Source: http://requirejs.org/logo.png



Summary – Q and A

TECHNISCHE UNIVERSITÄT DARMSTADT

Advancing the Chrome Reactive Inspector

- Reducing the cognitive load on the user
- Connecting the abstract graph with JavaScript code
- Rapidly updated observables
- Excessively created Observables
- Evaluation

Advancing the Chrome Reactive Inspector

TECHNISCHE UNIVERSITÄT DARMSTADT

Thank you

Advancing the Chrome Reactive Inspector

TECHNISCHE UNIVERSITÄT DARMSTADT

Backup Slides:

Reactive Programming

TECHNISCHE UNIVERSITÄT DARMSTADT

- very similar source code

Rx.js

```
addClick = Rx.Observable.fromEvent($addSon, 'click')
    .mapTo(1);
removeClick = Rx.Observable.fromEvent($removeSon, 'click')
    .mapTo(-1);
eventClick = addClick.merge(removeClick);
function plus(a, b) {
    return a + b
}
sonWalletValue = eventClick.scan(plus, 0)
fatherWalletValue = sonWalletValue
    .map(function (value) {
        return value + 10
    })
sonWalletValue.subscribe(
    function (data) {
        $sonWallet.val(data);
    });
```

Bacon.js

```
addClick = $addSon.asEventStream('click');
addClickMap = addClick.map(1);
removeClick = $removeSon.asEventStream('click');
removeClickMap = removeClick.map(-1);
eventClick = addClickMap.merge(removeClickMap);
function plus(a, b) {
    return a + b
}
sonWalletValue = eventClick.scan(0, plus);
fatherWalletValue = sonWalletValue
    .map(function (value) {
        return value + 10
    });
sonWalletValue.assign($sonWallet, "val");
```

Advancing the User Interface
Reducing the cognitive load

TECHNISCHE UNIVERSITÄT DARMSTADT