

# Advancing the Chrome Reactive Inspector

Bachelor-Thesis von Benedikt Gross

Tag der Einreichung:

1. Gutachten: Prof. Dr. Guido Salvaneschi
2. Gutachten: Placeholder second name



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Informatik  
Reactive Programming Technology

Advancing the Chrome Reactive Inspector

Vorgelegte Bachelor-Thesis von Benedikt Gross

1. Gutachten: Prof. Dr. Guido Salvaneschi
2. Gutachten: Placeholder second name

Tag der Einreichung:

---

## Ehrenwörtliche Erklärung

---

Hiermit versichere ich, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in dieser oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Die schriftliche Fassung stimmt mit der elektronischen Fassung überein.

Darmstadt, den 02. Januar 2018

Benedikt Gross



---

## Abstract

---

Abstract Placeholder



---

## Contents

---

1. Introduction	3
1.1. Background . . . . .	3
1.2. Motivation . . . . .	3
1.3. Our Contribution . . . . .	3
1.4. Outline . . . . .	3
2. State of the Art	5
2.1. Implementation of Reactive Systems . . . . .	5
2.1.1. Observer Design Pattern . . . . .	5
2.1.2. Aspect-Oriented Programming . . . . .	5
2.1.3. Callbacks . . . . .	5
2.1.4. Promises . . . . .	5
2.1.5. Iterator Vs Observer Pattern . . . . .	5
2.2. Reactive Programming . . . . .	5
2.2.1. Reactive Programming with JavaScript . . . . .	5
2.2.2. ReactiveX and RxJS . . . . .	5
2.2.3. Important Concepts of RxJS . . . . .	5
2.2.4. Bacon.js . . . . .	7
2.3. Debugging and Tools Support . . . . .	7
2.3.1. Debugging JavaScript . . . . .	7
2.4. Related Work . . . . .	7
List of Figures	9
Listings	11
Bibliography	13
Appendices	15
A. APPENDIX	17
A.1. App#1 - Operators and Events . . . . .	17
A.2. App#2 - Father-Son Wallet War . . . . .	17





---

## 1 Introduction

---

Introduction Header Placeholder

---

### 1.1 Background

---

---

### 1.2 Motivation

---

---

### 1.3 Our Contribution

---

In the last two sections, we introduced reactive systems, ...

As a summary, this thesis makes the following contributions:

- Item 1
- Item 2

---

### 1.4 Outline

---

Outline Placeholder



---

## 2 State of the Art

---

### Chapter Header

---

## 2.1 Implementation of Reactive Systems

---

### 2.1.1 Observer Design Pattern

---

### 2.1.2 Aspect-Oriented Programming

---

### 2.1.3 Callbacks

---

### 2.1.4 Promises

---

### 2.1.5 Iterator Vs Observer Pattern

---

## 2.2 Reactive Programming

---

### 2.2.1 Reactive Programming with JavaScript

---

### 2.2.2 ReactiveX and RxJS

---

### 2.2.3 Important Concepts of RxJS

---

Observable and Observer

Placeholder

Operators

RxJS Code Structure

Placeholder

---

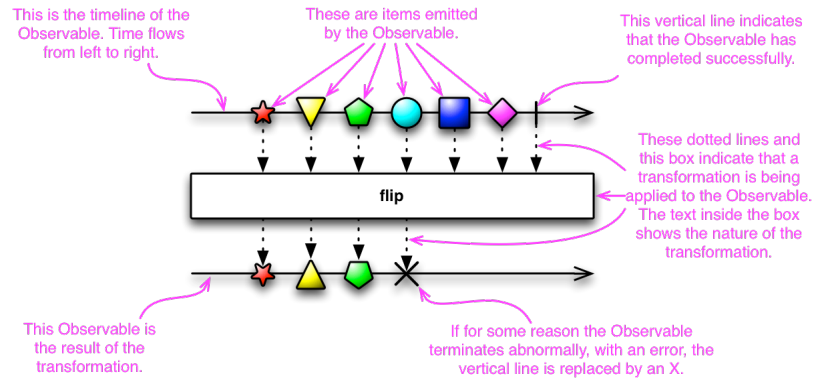


Figure 2.1.: Reactive pattern [1]

```

1 // 1. Srouce Observable Creation
2 var sourceObservable = Rx.Observable.interval(1000);
3 // 2. Transformation by applying different operators
4 var transformedObservable = sourceObservable.map(function(x) {
5     return x * 10;
6 })
7     .filter(function(x) {
8         return x !== 20
9     })
10    .take(5);
11 // 3. Subscribe to desired Observable
12 var subscription = transformedObservable.subscribe(
13     function(x) {
14         console.log('Next: ' + x);
15     },
16     function(err) {
17         console.log('Error: ' + err);
18     },
19     function() {
20         console.log('Completed');
21     });
22 // OUTPUT
23 Next: 0
24 Next: 10
25 Next: 30
26 Next: 40
27 Next: 50
28 Completed

```

Listing 2.1: RxJS Simple Example

---

## 2.2.4 Bacon.js

---

EventStream and Property  
Placeholder

---

## 2.3 Debugging and Tools Support

---

---

### 2.3.1 Debugging JavaScript

---

---

## 2.4 Related Work

---



---

## List of Figures

---

2.1. Reactive pattern [1] . . . . .	6
-------------------------------------	---





---

## Listings

---

2.1. RxJS Simple Example . . . . .	6
------------------------------------	---



---

## Bibliography

---

- [1] Rx: ReactiveX Observable. <http://reactivex.io/documentation/observable.html>. last accessed: 09-03-2017.





---

# Appendices



---

## A APPENDIX

---

Details on the applications used for evaluation in the chapter ?? are presented here. For each application, we present the source from where we took the application. We also mention the required refactoring for each application.

---

### A.1 App#1 - Operators and Events

---

This application contains lot of sub-applications to explain the operators and event handling with Bacon.js library.

Application Source:

This application is one of the examples from the GitHub repository of Bacon.js library.

<https://github.com/baconjs/bacon.js/blob/master/examples/examples.html>

Application Setup/Refactoring:

On page JavaScript code has been moved to separate JavaScript file, which is then included in HTML file.

---

### A.2 App#2 - Father-Son Wallet War

---

This application is based on a rule that father will always have ten dollars more than his son.

Application Source:

Detailed logic of this application can be found here.

<https://alfredodinapoli.wordpress.com/2011/12/24/functional-reactive-programming-kick-starter-guide/>

We found Bacon.js implementation in jsfiddle <http://jsfiddle.net/EeJgZ/> mentioned in one issue of GitHub repository of Bacon.js

<https://github.com/baconjs/bacon.js/issues/51>

Application Setup/Refactoring:

We copied the HTML from the jsfiddle to the HTML file and JavaScript code into a JS file, include it in HTML file with other dependencies like JQuery and Bacon.js libraries.