



**NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS**

**FACULTY OF EXACT SCIENCES  
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS**

**BACHELOR THESIS**

**RHEA: A Reactive, Holistic, Extensible, Abstract Framework  
for Dataflow Programming**

**Orestis Melkonian**

**Supervisors: Panos Rondogiannis, Professor EKPA  
Angelos Charalambidis, Researcher NCSR**

**ATHENS**

**APRIL 2016**

## **BACHELOR THESIS**

RHEA: A Reactive, Holistic, Extensible, Abstract Framework for Dataflow Programming

**Orestis Melkonian**

**A.M.:** 1115201000128

**SUPERVISORS:** **Panos Rondogiannis**, Professor EKPA  
**Angelos Charalambidis**, Researcher NCSR

## ABSTRACT

Summary here

**SUBJECT AREAS:** Dataflow programming, Stream Processing

**KEYWORDS:**  $\text{\LaTeX}$ , κλάσεις εγγράφων, πτυχιακές εργασίες, τμήμα πληροφορικής και τηλεπικοινωνιών, πανεπιστήμιο αθηνών

*”τὰ ὄντα ἰέναι τε πάντα καὶ μένειν ουδέν”*  
*(all entities move and nothing remains still)*  
*- Heraclitus*

## **ACKNOWLEDGEMENTS**

I would like to thank Angelos Charalambidis for his immensely helpful supervision and guidance throughout the whole period of 6 months that I was present in NCSR.

I would also like to thank Professor Panos Randogiannis for being a major influence in my current research interests through the undergraduate courses "Theory of Computation" and "Principles of Programming Languages", which cultivated a certain appeal to declarative languages, formal methods and generally abstraction.

# CONTENTS

<b>PROLOGUE</b>	<b>10</b>
<b>1. INTRODUCTION</b>	<b>11</b>
1.1 Motivation	11
1.1.1 Declarative languages	11
1.1.2 Abstraction as a mean to overcome complexity	11
1.1.3 Data versus Computation	11
1.2 Dataflows in Robotics	11
1.3 Dataflows in Big Data	11
1.4 Structure	11
<b>2. BACKGROUND</b>	<b>12</b>
2.1 Functional Reactive Programming	12
2.2 The Dataflow Computational Model	12
2.3 Stream processing	12
2.4 Flow-based programming	12
2.5 Model-driven architecture	12
2.6 Robotics	12
<b>3. APPROACH</b>	<b>13</b>
3.1 Generality	13
3.2 Extensibility (Evaluation + Distribution)	13
3.3 The Reactive Streams Standard	13
<b>4. IMPLEMENTATION</b>	<b>14</b>
4.1 General Documentation	14
<b>5. DEPLOYMENT</b>	<b>15</b>
<b>6. OPTIMIZATION</b>	<b>16</b>
<b>7. USE-CASES</b>	<b>17</b>

7.1	Robot Hospital Guide . . . . .	17
7.2	Robot Control Panel . . . . .	17
7.3	Camera Surveillance . . . . .	17
7.4	Hamming Numbers . . . . .	17
8.	RELATED WORK . . . . .	18
8.1	GoogleDataflow . . . . .	18
8.2	TensorFlow . . . . .	18
8.3	Akka . . . . .	18
8.4	dispel4py . . . . .	18
8.5	Flowstone . . . . .	18
8.6	Spark . . . . .	18
8.7	Naiad . . . . .	18
8.8	NoFlo . . . . .	18
9.	FUTURE WORK . . . . .	19
10.	CONCLUSIONS . . . . .	20
	TERMINOLOGY TABLE . . . . .	21
	ABBREVIATIONS, INITIALS AND ACRONYMS . . . . .	22

## LIST OF FIGURES



## LIST OF TABLES

## **PROLOGUE**

This bachelor project is a continuation of my internship at the National Centre for Scientific Research "Demokritos", particularly in the Software and Knowledge Engineering Laboratory (SKEL).

The main task I was assigned was the implementation of a framework for robot programming using a dataflow approach. During that internship, I came to realize that my work could be easily generalized to cover a much broader application area than just robot software.

## **1. INTRODUCTION**

### **1.1 Motivation**

#### **1.1.1 Declarative languages**

#### **1.1.2 Abstraction as a mean to overcome complexity**

#### **1.1.3 Data versus Computation**

### **1.2 Dataflows in Robotics**

### **1.3 Dataflows in Big Data**

### **1.4 Structure**

## **2. BACKGROUND**

- 2.1 Functional Reactive Programming**
- 2.2 The Dataflow Computational Model**
- 2.3 Stream processing**
- 2.4 Flow-based programming**
- 2.5 Model-driven architecture**
- 2.6 Robotics**

## **3. APPROACH**

### **3.1 Generality**

### **3.2 Extensibility (Evaluation + Distribution)**

### **3.3 The Reactive Streams Standard**

## **4. IMPLEMENTATION**

### **4.1 General Documentation**

## **5. DEPLOYMENT**

## 6. OPTIMIZATION



## **7. USE-CASES**

**7.1 Robot Hospital Guide**

**7.2 Robot Control Panel**

**7.3 Camera Surveillance**

**7.4 Hamming Numbers**

## **8. RELATED WORK**

**8.1 GoogleDataflow**

**8.2 TensorFlow**

**8.3 Akka**

**8.4 dispel4py**

**8.5 Flowstone**

**8.6 Spark**

**8.7 Naiad**

**8.8 NoFlo**

## **9. FUTURE WORK**

## **10. CONCLUSIONS**

## TERMINOLOGY TABLE

A table of used scientific terms follows.

κλάση	class
εντολή	command
περιβάλλον	environment

## ABBREVIATIONS, INITIALS AND ACRONYMS

A table of all abbrevations used throughout the thesis follows [1].

FRP	Functional Reactive Programming
-----	---------------------------------

## REFERENCES

[1] I. Freely, “A small paper,” *The journal of small papers*, vol. -1, 1997. to appear.