Tackling the Awkward Squad for Reactive Programming: The Actor-Reactor Model (Artifact)

Sam Van den Vonder
Software Languages Lab, Vrije Universiteit Brussel, Belgium

Thierry Renaux
Software Languages Lab, Vrije Universiteit Brussel, Belgium

Bjarno Oeyen
Software Languages Lab, Vrije Universiteit Brussel, Belgium

Joeri De Koster
Software Languages Lab, Vrije Universiteit Brussel, Belgium

Wolfgang De Meuter
Software Languages Lab, Vrije Universiteit Brussel, Belgium

Abstract

This artefact provides runnable versions of the code samples given in the main publication. An interpreter for the Stella language is provided together with a basic web-based IDE (syntax highlighting + running programs) which is able to run all Stella code given in the main publication. Also included are runnable implementations of the running example from the main publication (a simple wind turbine simulator) implemented in Stella and 6 other languages and frameworks (Akka, Flapjax, FrTime, ReactJS, REScala, and RxJS). While we do not discuss how these other technologies work, we highlight the interesting parts of the implementations of the running example: the difficulties we had, and any particular points of interest related to the claims made in the main publication.

2012 ACM Subject Classification Software and its engineering → Data flow languages; Software and its engineering → Multiparadigm languages

Keywords and phrases functional reactive programming, reactive programming, reactive streams, actors, reactors

Digital Object Identifier 10.4230/DARTS.6.2.7

Funding *Sam Van den Vonder*: Research Foundation – Flanders (FWO) grant No. 1S95318N
Thierry Renaux: Flanders Innovation & Entrepreneurship (VLAIO) "Cybersecurity Initiative Flanders" program
Bjarno Oeyen: Research Foundation – Flanders (FWO) grant No. 1S93820N

Related Article Sam Van den Vonder, Thierry Renaux, Bjarno Oeyen, Joeri De Koster, and Wolfgang De Meuter, "Tackling the Awkward Squad for Reactive Programming: The Actor-Reactor Model", in 34th European Conference on Object-Oriented Programming (ECOOP 2020), LIPIcs, Vol. 166, pp. 19:1–19:29, 2020. https://doi.org/10.4230/LIPIcs.ECOOP.2020.19

Related Conference 34th European Conference on Object-Oriented Programming (ECOOP 2020), November 15–17, 2020, Berlin, Germany (Virtual Conference)

1 Scope

This artefact provides all code samples that are used in the paper, as well as the code samples that were used to guide the evaluation of other reactive languages and frameworks (Akka, Flapjax, FrTime, ReactJS, REScala, and RxJS). Its purpose is not to go into the details of each individual implementation, but rather to guide the user to run the different code samples and to observe the expected outcomes. While we will highlight interesting parts of the implementation of projects, we will not go into the details of how they work exactly.

2 Content

This package includes:

Folder "`code_supplements`": This folder contains 7 different software projects that implement the common example of a simple wind turbine simulator, which corresponds to the example used in the main publication. The different projects are contained within the subfolders called Akka, Flapjax, FrTime, ReactJS, REScala, RxJS, and Stella. In the Stella project we provide all code examples used in the main publication. Additionally, in the REScala project we provide the 2 code samples related to identifying the "Reactive Thread Hijacking Problem" and the "Reactive Update Order Leak". For a more detailed overview of all files and folders included in the corresponding projects, please consult the user guide.

Folder "`VirtualBox`": If running the provided code natively is not possible or not desirable, a virtual machine image is provided in the "Open Virtualization Format 2.0". This image has all software packages and dependencies pre-installed. For instructions on how to import and use this image, please consult the user guide.

Document "`ECOOP 2020 Artefact User Guide.pdf`": This user guide walks the user through the contents of the artefact, how to run the different projects, and how to observe the intended results. It also details the required software and their versions for each project, and provides instructions on how to use the provided virtual machine image via VirtualBox.

Document "`LICENSE`": This artifact includes software that is distributed under various licenses. Please consult this license file or the license information included in this document.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also archived on Zenodo and can be found via DOI "`10.5281/zenodo.3862954`" (`https://doi.org/10.5281/zenodo.3862954`).

4 Tested platforms

This artefact was created on a MacBook Pro (15 inch, 2016) running macOS 10.15.4. All software required by this artefact is available on multiple platforms and has no specific minimum system requirements. When running the software natively (not in a virtual machine) a Windows, macOS, or Linux device is required that is capable of running a web browser, DrRacket (an IDE) and the Oracle JDK or OpenJDK. The projects that make use of the Scala SBT (Akka and REScala) will download additional dependencies when they are first run (the Akka and REScala runtime), and thus a network connection is required.

A virtual machine image is provided with all software dependencies pre-installed. To run this virtual machine a device is needed capable of running virtualised software (e.g. via VirtualBox). More specifically, the device needs:

- At least 1GB of free memory (RAM). More memory results in a better experience.
- At least 11GB of free disk space. When we import the virtual machine image into VirtualBox, the disk space consumed by the image is approximately 10.6GB
- Any reasonably powerful x86 hardware, e.g. any Intel or AMD x86 CPU for laptops or desktops released after 2010 should do, possibly except for the very low end of the performance spectrum.

Instructions to run specific code samples on different platforms are provided in the user guide with their respective projects.

5 License

This artefact includes software that is distributed under various licenses. Please see the licenses of the corresponding software packages for complete and up-to-date information.

In our own code we make use of certain software (e.g. tools or libraries) that fall under the following licenses:

- Flapjax is distributed under the "3-Clause BSD License", also known the "New BSD License" or "Modified BSD License".
- Babel is distributed under the "MIT License".
- React-DOM is distributed under the "MIT License".
- React is distributed under the "MIT License".
- RxJS is distributed under the "Apache License 2.0".
- Bulma is distributed under the "MIT License".
- CodeFlask is distributed under the "MIT License".

The following files are written by ourselves and are licensed under the "Creative Commons Attribution 3.0 Unported (CC-BY)". This includes the following files:

- `code_supplements/Akka` (all files in this directory and its subdirectories)
- `code_supplements/Flapjax/wind-turbine-simulator.html`
- `code_supplements/FrTime/wind-turbine-simulator.rkt`
- `code_supplements/ReactJS/wind-turbine-simulator.html`
- `code_supplements/REScala` (all files in this directory and its subdirectories)
- `code_supplements/RxJS/wind-turbine-simulator.html`
- `code_supplements/Stella/size-change-termination/css/style.css`
- `code_supplements/Stella/size-change-termination/size-change-termination.html`
- `code_supplements/Stella/wind-turbine-simulator/css/style.css`
- `code_supplements/Stella/wind-turbine-simulator/wind-turbine-simulator.html`
- `ECOOP 2020 Artefact User Guide.pdf`

The "Stella" language interpreter is written by ourselves and is distributed under the "Creative Commons Attribution NonCommercial (CC-BY-NC)". This includes the following files:

- `code_supplements/Stella/size-change-termination/stella/stella-evaluator.min.js`
- `code_supplements/Stella/wind-turbine-simulator/stella/stella-evaluator.min.js`

We distribute an image of a virtual machine in the "Open Virtualization Format 2.0" which includes preinstalled software. This software is distributed under the following licenses:

- The Kubuntu operating system is distributed under the "GNU GPL version 2 or later".
- Mozilla Firefox is distributed under the "Mozilla Public License".
- Racket (DrRacket editor) is primarily distributed under the "Apache License 2.0" and the "MIT License". The compiled "racket" executable for the "Regular" variant of Racket and packages that distribute third-party libraries, are distributed under the "GNU Lesser General Public License (LGPL) version 3.0".
- OpenJDK is distributed under the "GNU General Public License, version 2, with the Classpath Exception".
- Scala SBT is distributed under the "Apache License 2.0".
- Akka is distributed under the "Apache License 2.0".

- REScala is distributed under the “Apache License 2.0”.
- Logback is distributed under the “Eclipse Public License v1.0” and the “GNU Lesser General Public License version 2.1”.

6 MD5 sum of the artifact

MD5 (Artefact.zip) = 53103f0bff3d935104a11abc82bdd411

7 Size of the artifact

4.48 GB (4,480,124,959 bytes)

Please note that operating systems can report file sizes using the decimal system (base 10) or the binary system (base 2). The number given here is in the decimal system. Please consult the number of bytes for the correct file size.