Fine-grained Language Composition: A Case Study (Artifact)*

Edd Barrett[1], Carl Friedrich Bolz[2], Lukas Diekmann[3], and Laurence Tratt[4]

1 **Software Development Team, Department of Informatics, King's College London.**
`http://soft-dev.org/` `http://eddbarrett.co.uk/`
2 **Software Development Team, Department of Informatics, King's College London.**
`http://soft-dev.org/` `http://cfbolz.de/`
3 **Software Development Team, Department of Informatics, King's College London.**
`http://soft-dev.org/` `http://lukasdiekmann.com/`
4 **Software Development Team, Department of Informatics, King's College London.**
`http://soft-dev.org/` `http://tratt.net/laurie/`

Abstract

This artifact is based on: PyHyp, a language composition of PHP and Python using meta-tracing; and Eco, a language composition editor. The provided package is designed to support the experiments, case studies, and demos detailed in the companion paper.

1998 ACM Subject Classification D.3.4 Processors

Keywords and phrases JIT, tracing, language composition

Digital Object Identifier 10.4230/DARTS.2.1.1

Related Article Edd Barrett, Carl Friedrich Bolz, Lukas Diekmann, and Laurence Tratt, "Fine-grained Language Composition: A Case Study", in Proceedings of the 30th European Conference on Object-Oriented Programming (ECOOP 2016), LIPIcs, Vol. 56, pp. 3:1–3:27, 2016.
`http://dx.doi.org/10.4230/LIPIcs.ECOOP.2016.3`

Related Conference 30th European Conference on Object-Oriented Programming (ECOOP 2016), July 18–22, 2016, Rome, Italy

1 Scope

The artifact allows the user to (in a VirtualBox VM): re-run the experiment; generate the result tables seen in the paper; run the case studies and demos detailed in the paper.

2 Content

The artifact package includes:

- A pre-configured Debian 8 VirtualBox disk image.
- A `README` file documenting how to use the disk image.

The VirtualBox image contains:

- Pre-compiled versions of the programming language VMs used in the paper.
- The Eco editor, ready to run.
- The benchmark suite from the paper, ready to run.
- The case studies and demos from the paper, ready to run.

* This work was funded by the EPSRC COOLER (EP/K01790X/1) and LECTURE (EP/L02344X/1) grants.

- Raw data from our benchmarking run.
- A script to generate the result tables seen in the paper.
- Documentation for all of the above.

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 available at: `http://soft-dev.org/pubs/files/pyhyp/`.

4 Tested platforms

The artifact is intended for use with Oracle VirtualBox (https://www.virtualbox.org/). We used version 4.3.18. Once imported into VirtualBox, the artefact uses about 10GiB of disk space. 4GiB of RAM should suffice to run the artifact in VirtualBox.

5 License

Various licenses. Please consult the enclosed `README` file.

6 MD5 sum of the artifact

3e0dc822d5b36538c1ab86e0c0d799e6

7 Size of the artifact

4.2 GiB

Acknowledgements. We thank Jasper Schulz for testing, and the AEC reviewers for their thorough comments. Any errors and omissions are our own.