Compositional Design, Implementation, and Verification of Swarms (Artifact)

John Q. Public
Dummy University Computing Laboratory, [optional: Address], Country
My second affiliation, Country

Joan R. Public[1]
Department of Informatics, Dummy College, [optional: Address], Country

Abstract

Swarm protocols are a recently-introduced formalism for specifying and verifying the behaviour of distributed ensembles of machines that interact in a swarm. The machines communicate by emitting events that propagate asynchronously throughout the swarm while adhering to a *local-first* paradigm — meaning that a machine can make progress even without always-active connections to other machines. A swarm behaves correctly if its machines are *eventually faithful* to a swarm protocol, i.e., if they eventually reach a consensus on a valid execution of the protocol. Previous work presents verification methods to ensure eventual fidelity.

Unfortunately, the existing techniques for designing, implementing, and verifying swarm applications based on swarm protocols are *not compositional*. Consequently, such techniques do not support the modular design and development of large and complex swarm applications as compositions of simpler swarms. They also do not support the reuse of machine implementations and they lead to inefficient verification performance. The paper introduces novel theory and techniques for the compositional specification and verification of swarm protocols, and for the composition of swarms. It also present a compositional method to ensure that a swarm is eventually faithful to a composition of swarm protocols, allowing the reuse and adaptation of pre-existing machines. The artifact, correspondingly, contains software tools for the compositional design, implementation and verification of swarms together with runnable examples from the paper. Additionally, the artifact contains scripts that run the experiments reported in the paper and generates corresponding figures.

2012 ACM Subject Classification General and reference → General literature; General and reference

Keywords and phrases Dummy keyword

Digital Object Identifier 10.4230/DARTS.3.2.1

Funding *John Q. Public*: (Optional) author-specific funding acknowledgements
Joan R. Public: [funding]

Acknowledgements I want to thank ...

Related Article John Q. Open and Joan R. Access, "A very nice paper", in Proceedings of the 30th Conference on Very Important Topics (CVIT 2016), LIPIcs, Vol. 0, pp. 0:1–0:2, 2016.
https://doi.org/10.4230/LIPIcs.xxx.xxx.xxx

Related Conference 42nd Conference on Very Important Topics (CVIT 2016), December 24–27, 2016, Little Whinging, United Kingdom

1 Scope

What is the scope of the artifact? What claims of the related scholarly paper are intended to be backed by this artifact?

[1] Optional footnote, e.g. to mark corresponding author

2 Content

The artifact package includes:

- ...

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: `https://to.be.specified`.

4 Tested platforms

Please specify the platforms on which the artifact is known to work, including requirements beyond the operating system such as large amounts of memory or many processor cores.

5 License

The artifact is available under license ....

6 MD5 sum of the artifact

xxx

7 Size of the artifact

x.xx GiB

References

1. *16th Annual Symposium on Foundations of Computer Science, Berkeley, California, USA, October 13-15, 1975*. IEEE Computer Society, 1975.
2. Edsger W. Dijkstra. Letters to the editor: go to statement considered harmful. *Commun. ACM*, 11(3):147–148, 1968. doi:10.1145/362929.362947.
3. Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.
4. John E. Hopcroft, Wolfgang J. Paul, and Leslie G. Valiant. On time versus space and related problems. In *16th Annual Symposium on Foundations of Computer Science, Berkeley, California, USA, October 13-15, 1975* [1], pages 57–64. doi:10.1109/SFCS.1975.23.
5. Donald E. Knuth. Computer Programming as an Art. *Commun. ACM*, 17(12):667–673, 1974. doi:10.1145/361604.361612.