COORDINATION'26 Artefact Evaluation Guidelines

The artefact evaluation (AE) differs according to the type of paper submitted:

- Accepted regular papers may be invited to submit an accompanying artefact for evaluation by the AE committee.
- **Tool papers** must submit an artefact that at least satisfies the requirements for the **functional** badge (see Section 4). Papers not meeting this clause are rejected.

1 Badge claims

Authors should indicate the EAPLS badges[1] they intend to claim with justification why their artefact qualifies for the claimed badges. Section 4 details the badges available. Papers may claim up to two badges.

Example 1 (Claiming badges)

We claim two badges (i) Artefacts Available and (ii) Artefacts (Functional and) Reusable. The reasons why our tool fulfils the requirements set out by the EAPLS scheme are outlined below.

2 Submission format

When packaging their artefact for submission, authors should provide specific scenarios describing how it can be concretely used. Authors can also consider variations on those scenarios to showcase the robustness or reusability of their artefact. Section 3 describes the template authors should follow when structuring their AE submissions.

Every submitted artefact must include the following:

- The URL from where the artefact can be downloaded.
- A quick start guide explaining how the artefact environment is set up.

The AE committee expects artefacts to be executable on major operating systems (macOS, Linux, Windows). Authors are strongly encouraged to package their artefacts as **Docker images** (also **OVA/OVF virtual machines** are accepted but Docker is preferred) since this facilitates the AE set-up process.

3 Evaluation instructions

The AE instructions should be attached as a post-bibliography appendix in the original paper. It must detail the steps reviewers should follow to confirm the functionality or reusability claims. This appendix is reserved for the AE procedure and will not be published with the camera-ready version of the paper. We recommend the appendix format outlined in Section A.1 to Section A.4.

[1] please take a look at `https://eapls.org/eapls/artifact-badges/`

4 Badge Definitions

4.1 Available

To claim the *Available* badge, authors must provide a link to a publicly accessible repository that permanently hosts artefacts. If this is not possible at the time of submission, authors should indicate the repository on which they intend to host the artefact (e.g., Zenodo, Software Heritage). The hosting platform should not track the identities of downloaders; using non-compliant hosting platforms is valid grounds for outright artefact rejection.

4.2 Functional

To claim the *Functional* badge, the submitted artefact should be reasonably documented and include evidence of proper validation. Authors should identify all the individual outcomes in the paper, which must be reproducible through the artefact. Each functional outcome should be numbered, $F_1, \ldots, F_n$, and accompanied by a short explanation that links that outcome obtained via the artefact to the one in the paper.

Example 2 (Functional outcomes)

F_1 Experiment 4 shows that our aggregate computing simulation scales linearly with the number of devices.(Outcome) The throughput and latency values printed by this experiment correspond to rows 1–4 listed in Tbl. 3.(Link with paper)

4.3 (Functional and) Reusable

To claim the *Reusable* badge, the submitted artefact should be of high quality that exceeds the requirements for the functional badge. Authors should outline individual reusability claims, $R_1, \ldots, R_n$, that demonstrate how the artefact may be used in scenarios beyond those covered in the paper. Each scenario should be presented as an example that AE reviewers can replicate.

Example 3 (Reusability claims)

R_1 Our first example shows how the coordination simulator handles custom network topologies besides the ones studied in Sec. 4 of the paper.(Claim) These instructions configure a new topology and launch the simulation, which produces a convergence plot after step 4. . . (Replicable scenario)

A.1 Badge claims

Authors should state the EAPLS badges they intend to claim, motivating why the submitted artefact qualifies for each badge. Tool papers must claim the functional badge and meet its requirements. Refer to Section 4 for examples on how to specify the functional outcomes and reusability claims. Functional outcomes and reusability claims can be organised under these sub-headings.

Functional outcomes

F_1 This outcome...

Reusability claims

R_1 This claim...

A.2 Quick start

This phase confirms whether the artefact and its environment are fully operational prior to the main evaluation. The quick start guide should consist of a **short** sequence of sanity checks that reviewers execute. Ideally, authors provide a script that automates these checks. Authors are advised to give a high-level directory structure of their submitted artefact since this helps AE reviewers familiarise themselves with the artefact.

- `tool/`
  - `simulator/` (*Simulation engine root*)
  - `doc/` (API documentation)
  - `src/` (Sources)
  - `test/` (Unit tests)
  - `build.sbt` (Build definition)
- `examples/` (*Reusability claims examples*)
  - `custom-topology/` (Custom network topology scenario)
- `experiments/` (*Functionality outcomes examples*)
  - `gradient/` (Gradient self-organisation case study)
  - `collect/` (Data collection protocol)

Figure 1: Tool directory structure example

Example 4 (Directory structure)

Figure 1 describes the directory layout of our submitted artefact. The directories used in Sections A.3 and A.4 are underlined. AE reviewers may safely ignore the other directories.

The quick start guide aims to facilitate the "kick-the-tyres" phase. At this stage, reviewers may communicate with authors for early feedback or help with difficulties.

A.3 Functional evaluation

This evaluation replicates the functional outcomes $F_1, \ldots, F_n$ listed in Section A.1. Authors should structure this appendix section as a sequence of steps AE reviewers follow to verify all the functional outcomes. Reference to individual functional outcomes should be made by authors when arguing why or how each outcome is met. Justification should be given whenever the results obtained through the artefact differ from those in the paper.

Example 5 (Justifying difference in results)

The rendered plot produced by experiment 5 (functional outcome F_5) reflects the same convergence trends of Fig. 7 in the paper, albeit on a smaller network. This difference between the artefact plot and Fig. 7 results from the *lower* number of simulated devices we use for our artefact set-up to keep execution time reasonable, per the AE guidelines.

A.4 Reusability evaluation

This evaluation validates the reusability claims $R_1, \ldots, R_n$ identified in Section A.1. Authors should structure this section of the appendix as a sequence of steps AE reviewers follow to explore the reusability of the submitted artefact. These steps can *(i)* direct AE reviewers to inspect content packaged with the artefact (e.g., documentation), *(ii)* give scenarios where the artefact may be reused (e.g., imported as a library into other software), or *(iii)* follow instructions which showcase how the artefact can be used out-of-the-box to tackle use cases besides those given in the paper (e.g., a short tutorial with examples).

Example 6 (Reusing the artefact on other examples)

Our simulation engine exposes a configuration API that enables users to define custom coordination programs (reusability claim R_2). This tutorial shows how an arbitrary aggregate computing program written in ScaFi can be simulated on a user-defined network topology by invoking the simulator with the `--topology` and `--program` flags as follows...

General expectations

AE reviewers should be able to complete the evaluation in $\approx$ 4 hours. If the evaluation relies on long-running experiments (e.g., to gather empirical data), the evaluation harness should provide a push-button method for executing experiments with *minimal* intervention from AE reviewers (e.g., using a script). Artefacts packaged as Docker images or OVA/OVF virtual machines reduce the AE set-up overhead.