

oar (or `~/u/oar`) (short for **o**bjective **a**rchive **r**eflection) is a *static website compiler*—an eDSL for describing the construction and deployment of your statically-generated website.

The concept of oar isn't new: it is an mashing of multiple sources of inspiration:

- `oar.fsbrowse` (a submodule of oar) generates a front-end for a repository, inspired by Git front-ends like `cgit`¹. However, in oar we simplified this idea to support a normal, non-VCS folder repository, as well as Git repository².
- The syntax and typing design of the oar eDSL is heavily inspired by the works of Daniel Bünzli (`dbuenzli`) especially his build system `b0`³. oar's construction units are declared with and referred to by string-based name identifiers, just like in `b0`.
- For oar's source code, the writing style is also heavily inspired by Bünzli's, where the implementation is composed of top-scope let-bindings as much and “clean”-ly as possible.

Some other related works that we have considered:

- `OCurrent`
- The Dune build system⁴ is technically expressive enough for user to, through Dune build rules, describe the construction of a statically-generated websites. However, oar is used by the author Kinten Le who uses an OCaml distribution⁵ where Dune is not the first-class build system.

The development of oar also birthed some interesting OCaml programming patterns. Namely, we experimented with `ppx-memberinclude`, a syntax extension for OCaml for defining module value members from top-scope, to ease module nesting.

Up till the writing of this article, oar has been successfully used in the making of Kinten Le's portfolio website and TTKB-HCMUT's introduction website.

Future work⁶

- `pyar`: oar port for Python

Acknowledgement

¹<https://git.zx2c4.com/cgit/about/>

²Support for Git is work in progress

³<https://erratique.ch/software/b0>

⁴<https://dune.build/>

⁵<https://baorepo.web.app/~kinten/u/utop>

⁶This section is being constantly updated.