

Ob-lesim: Learning Simulator integration for Org-Mode

Stefano Ghirlanda

May 10, 2023

1 Introduction

Ob-lesim supports editing and running Learning Simulator scripts in Emacs org-mode. In short, it is a bridge between org-babel (org-mode's code execution system) and lesim-mode, an Emacs major mode for Learning Simulator scripts.

2 Installing

1. Install lesim-mode.
2. Put `ob-lesim.el` in your Emacs load path, for example, `~/emacs.d/elisp`.
3. Add this to your Emacs init file, for example, `~/emacs.d/init.el`:

```
(require 'ob-lesim)
(add-to-list 'org-babel-load-languages '(lesim . t))
```

3 Using

Ob-lesim works mostly behind the scenes. Editing lesim code blocks in org-mode should feel familiar:

- Embed Learning Simulator code in org-mode buffers with:

```
#+begin_src lesim
...
#+end_src
```

- Ob-lesim defines these default header arguments:
 - `:results silent`, because Learning Simulator does not directly output that makes sense to insert.¹
 - `:noweb yes`, see 4.
 - Other header arguments have no effect unless handled directly by org-mode. For example:
 - * Use `:eval no-export` to prevent evaluation of long-running scripts when exporting.
 - * Use `:tangle file.les` to write a script to `file.les`.

¹This will change when Learning Simulator supports saving figures without human intervention.

- Scripts can be run as code blocks in the org-mode buffer with the usual `C-c C-c` key binding.
- Scripts can also be run from the edit buffer, which has major mode `lesim-mode`, with the regular lesim-mode key binding (by default, `C-c C-r`).
- Lesim-mode features are partly available in the org-mode buffer (for example, `TAB` moves by field in phase blocks), and fully available in org-mode edit buffers.

4 Sharing code across scripts

Ob-lesim supports noweb references to share code across lesim code blocks. For example, this is how you can make sure that two different scripts use the same parameter values:

This is a lesim parameter block:

```
#+name: common-parameters:
#+begin_src lesim
  # lesim parameter block here
#+end_src
```

This is the script 1:

```
#+begin_src lesim
  <<common-parameters>>

  # script 1 here here
#+end_src
```

This is script 2:

```
#+begin_src lesim
  <<common-parameters>>

  # script 2 here here
#+end_src
```

You can also share phase blocks and other script snippets if useful. Thanks to awesome engineering, noweb references are automatically resolved when tangling, so that you end up with complete scripts.

5 Bugs and planned features

Bugs and planned features are tracked as issues on Github.

6 Internals

These are mostly notes to myself. Ob-lesim has two ways of running scripts:

1. Evaluate a code block in an org-mode buffer. This works by saving `full-body` to temporary file, expanding any noweb references, then using `lesim-error` and `lesim-run` on the file.

2. Evaluate an org-src edit buffer with `C-c C-r` (`lesim-mode-run-key`). This works by expanding noweb references, then saving the buffer to a temporary file, using `lesim-error` and `lesim-run`, then collapsing the noweb references again.