

Ob-lesim: Learning Simulator integration for org-Mode

Stefano Ghirlanda

May 16, 2023

Ob-lesim supports editing and running Learning Simulator scripts in Emacs org-mode. It links org-babel (org-mode's code execution system) and lesim-mode, an Emacs major mode for Learning Simulator scripts.

1 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))
```

2 Using

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

- You can embed Learning Simulator code in org-mode files with:

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

- Ob-lesim defines these default header arguments:

- `:results value`, which is used to insert error messages from the Learning Simulator as `#+RESULTS: #`.
- `:exports none`, since you rarely need to export error messages or scripts.
- `:noweb yes`, see 3.
- `:eval no-export` because Learning Simulator scripts can take quite some time to run.

You can override these defaults globally for an org file or locally for a lesim code block with the usual org-mode mechanisms.

- Ob-lesim supports the `:var` header argument to assign parameter values in a script. For example, this code will set `beta` to 2:

```
#+begin_src lesim :var beta=2
...
#+end_src
```

This is useful to run the same script with different settings, especially together with noweb references. Behind the scenes, `:var` values are set at the beginning of the script, and are overwritten by any later assignment.

- Other header arguments have no effect unless handled directly by org-mode. For example, you can use `:tangle file.les` to write a script to `file.les`.
- 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 `lesim-mode` key binding (by default, also `C-c C-c`).
- Lesim-mode features are partly available in the org-mode buffer (for example, `TAB` and `Shift-TAB` to move around), and fully available in org-mode edit buffers (for example, validation).

3 Sharing code across scripts

Ob-lesim supports noweb references to share code across lesim code blocks. For example, this is how you can ensure that two 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
```

Noweb references are automatically resolved when running a code block or tangling it, so that you end up with complete scripts.

4 Bugs and planned features

Bugs and features are tracked as issues on Github.