

DEMO: Virtual Obeya

Heikki Virtanen

1 DEMO: Virtual Obeya

Basically Obeya is a room dedicated for a single purpose. The walls are for visualising relevant information, for example, the state of the designed product and the development process, or organisation's strategy and business. Obeya is also equipped with furniture and accoutrements which enables efficient (group) work in the room. (<http://blog.maskell.com/?p=374>)

Virtual Obeya is a platform for (open) development communities which can not gather to one place physically. It is not a collaboration platform in traditional sense but a versatile and variable framework obeying local software and liquid data approach. And more importantly, there can be multiple views for single data and it is easy to switch between views, as easy as to turn around and look another wall.

This is only a proof of concept demo.

2 Setup and use

2.1 Required software

Experimenting with this demo, you will need:

- X11 and Fvwm window manager with at least 4x4 pages virtual desktop.
- Gnu Emacs with org-mode.
- Bash

2.2 Configuring Fvwm

Module FvwmCommandS have to be loaded, for example, in StartFunction. This is required by FvwmCommand program.

For easy switch between walls, the numeric keypad can be used by adding following lines to Fvwm config:

```
PointerKey KP_Home      A      M      GotoPage 1 1
PointerKey KP_Up        A      M      GotoPage 2 1
PointerKey KP_Prior     A      M      GotoPage 3 1

PointerKey KP_Left      A      M      GotoPage 1 2
PointerKey KP_Begin     A      M      GotoPage 2 2
PointerKey KP_Right     A      M      GotoPage 3 2

PointerKey KP_End       A      M      GotoPage 1 3
PointerKey KP_Down      A      M      GotoPage 2 3
PointerKey KP_Next      A      M      GotoPage 3 3
```

2.3 Starting Virtual Obeya

(I) Open this file with Gnu Emacs and evaluate follown code scrap which loads required org-mode initialisations.

```
(require 'ob-python)
(require 'ob-sh)
(setq org-confirm-babel-evaluate (lambda (lang body)
  (not (or (string= lang "sh")
            (string= lang "python")
            (string= lang "emacs-lisp")))))
```

The actual Virtual Obeya realisation code (`entertoobeya`) is attended at the end of this file.

(II) Define the configuration of the walls using table. For example

strategic _{plans} .org	Obeya-Plan	1	2
activities.org	Obeya-Do	2	1
status _{andoutcomes} .org	Obeya-Check	3	2
initiatives.org	Obeya-Act	2	3
generic _{notes} .org	Obeya-Notes	2	2
readme.pdf		1	3

The columns are

1. File to open. Can be empty, if a wall is supposed to be occupied by application already running. Those can be started by ObeyaInitialisationHook scrap which is evaluated at the beginning.

2. The name of X11 window as seen by Fvwm. If file is opened, this defines the name of the X11 window of the Emacs frame.
3. Coordinates of the desktop page. There can be several X11 windows open on a wall.

Here is the skeleton for `ObeyaInitialisationHook`. It is required even if it is not used. Note one way to start applications to be run background is using `FvwmCommand`.

```
echo "This is currently empty."  
FvwmCommand "Exec evince $(pwd)/readme.pdf"
```

(III) Realise the Obeya by evaluating following

Note that the parameter is name of the table defining what is shown on walls.

3 Virtual Obeya Realisation Code