Lisp in Summer Projects Submission

Submission Date	2013-10-24 17:27:23
Full Name	Thorsten Jolitz
Country	Germany
Project Name	iOrg
Type of software	web app
General category	framework
LISP dialect	PicoLisp
GitHub URL	https://github.com/tj64/iorg
Did you start this project?	No, I'm modifying or extending an existing project.
Which file or directory contains the majority of your work?	/picoLisp/iorg/lib.l
Briefly describe your modifications	1. Made PicoLisp Wiki work with Org-mode syntax 2. Wrote Elisp wrapper library for PicoLisp's scrape.l 3. Extended the E/R model of the PicoLisp wiki for mapping Org elements to PicoLisp database objects 4. Functionality for communication between Emacs and PicoLisp (e.g. readParseTree for reading parse trees of Org files into PicoLisp DB objects. 5. Patched Emacs inferior-picolisp-mode to enable interactive scraping of iOrg apps in Emacs inferior PicoLisp buffer 6. Patched emacs-w3m to enable its use as Emacs UI for iOrg web-apps. Textareas with contents in Org syntax can now easily be edited in Org-mode buffers
Project Description	I want to describe my project in this form.
Purpose	Merge the PicoLisp web application framework and GNU Emacs Org-mode into one wiki-like framework that offers both - PicoLisp's powerful functionality for creating

interactive webapps (with database) and Org-mode's powerful functionality for creating (and exporting) arbitrarily complex static content.

Function

Replace the PicoLisp Wiki's own html rendering mechanism with calls to Emacs Org-mode html-exporter (i.e. replace PicoLisp wiki syntax with Org-mode syntax). Convert parse trees of Org files produced by the Org-mode parser into PicoLisp database objects that can be edited via the web GUI by non-Emacs users too.

Motivation

PicoLisp and GNU Emacs Org-mode (written in Emacs Lisp) are my two favorite software products that are both 'best in class' in their respective areas. By combining them into one framework I wanted to produce real synergy effects (not just rhethorical ones)

Audience

There is a real demand in the Org-mode community for a wiki/blog like software based on Org-mode syntax and editable via Emacs Org-mode. PicoLisp programmers that use Emacs as editor might enjoy the content creation facilities offered by Emacs Org-mode for webapps that focus on content like wikis and blogs.

Methodology

iOrg is based on the PicoLisp wiki and thus has the same two working areas: the wiki itself and the administration area.

In the wiki, user can read, edit and add wiki articles in Org syntax now that will be exported to html by the new Org exporter framework. Normal users use the browser UI, Emacs user should use emacs-w3m with my patch for editing textareas in Org-mode.

The administration area offers all the usual user, role, document etc administration. Additionally, Org files can now be read from PicoLisp and converted into database objects that can then be editied and administrated in the Org Elements section.

In Emacs, an inferior PicoLisp process can be started and used for interactive web-scraping of an running iOrg webapp in the PicoLisp REPL. I wrote iorg-scrape-minor-mode and iorg-quick-scrape-minor-mode to make this convenient. Its also possible to start an PicoLisp TCP scrape server and do the scraping using an TCP process object in Emacs.

While scraping is for PUTing things on the iOrg server (unless emacs-w3m or a standard browser is used for this), iorg-retrieve-url allows to GET info from the server. This is as easy as putting a DB query into a file and call the file's URL from Emacs.

Conclusion

PicoLisp and Org-mode do complement each other extremely well, and communication between two lisps should be easy. Should be... but the differences in print/read syntax and the somehow unclear protocol for communication via stdout/stdin using emacsclient (the only fast enough option) as well as the extremely rich Org-mode syntax makes it quite complicated to make Emacs Org-mode and PicoLisp talk to each other.

	Producing static html pages for the PicoLisp wiki with Org-mode work so far, except the somehow irrational appearance of surrounding double quotes around the html bodies (a bug in emacsclient?). Reading the parse trees of some Org files into PicoLisp DB objects works, with many other Org-files there are still errors. The web UI for editing Org Elements is only partly existing and functional. However, the benefits to expect do motivate further development of the project: - speed and scalability for iOrg agenda queries - concurrent working for groups due to DB locks - cooperation between Emacs Org-mode users and normal people (that use the browser UI) - fast creation of complex content from within Emacs
Build Instructions	- unpack tar.gz - local install (see /picoLisp/INSTALL)
Test Instructions	See the tutorial that is content of the iOrg wiki, either reading the .org files or the exported html online.
Execution Instructions	just click your way through the application, and don't forget to check emacs-w3m (new patched version!) as Emacs UI for iOrg
Describe any bugs or caveats	- html export surrounded by double quotes - reading parse trees work only with some .org files - GUI for Org Elements rudimentary and not fully functional
Official	I have read rules and have abided by them. I am 18 years of age or older. I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran, Myanmar (Burma), North Korea, Sudan, or Syria.