Anant **Narayanan**
anant@kix.in
http://www.kix.in/

Annotation Support for Evince

A Summer of Code **Proposal**
Prepared for The Gnome Foundation
March 20th, 2007

Abstract

PDF is now an open standard and provides users with a high quality representation of content suitable for both online viewing and printing. Evince, Gnome's document viewer provides support for a variety of document formats, including PDF. It, however, lacks support for annotations as described in section 8.4 of the PDF specification.

This project aims to add support for parsing, rendering, adding and modifying annotations as per the Adobe PDF specification to Evince. This task would involve extending the Poppler Glib bindings to support annotations, modifying Evince to properly render those annotations, and then furthering the effort by modifying Evince to enable modification, addition and deletion of annotations. How this will be done is detailed in this proposal.

Why?

As explained in the Gnome Live! Wiki [1]:

"We all do annotations on paper when we read documents. We do annotations to allow us to locate important parts of the text when passing over pages. Our brain dispatches an alarm when our eyes look that fluorescent yellow marks over a lot of blank characters"

PDF [2] is an extremely popular form of document sharing, and with the recent "opening up" of the standard itself, becomes a *defacto* way of sharing content. And of the most important features for any document format that is meant to be shared is that of the ability to add notes. Even in the cases where the document is not shared, it is certainly useful to mark certain areas of the document for later review: like notes in a research paper (all of which are in PDF these days!)

Since the PDF standard offers support for annotations, it is important for Evince [3] to exploit those features and benefit its users by providing them with a powerful and free (as in speech) PDF viewer that is on par with the commercial alternatives.

How?

Some of the legwork for adding annotation support has already been done by the Evince developers in the 'evince-form' branch [4]. This branch adds rudimentary support for interactive forms embedded in PDF files. Some of the code from this branch of Evince can be used for rendering annotations.

This task will primarily consist of four distinct steps:

- Update the evince-form branch to incorporate the latest changes in trunk. Update the Glib bindings for Poppler [5], the PDF library currently used by Evince; to add support for annotations. Poppler already supports annotations to some extent [6], and the Qt4 bindings already incorporate them [7].

- Modify the evince-forms branch to render annotations while viewing PDF files that contain them. Once the annotations are parsed through glib-poppler, they can be rendered on the screen using Cairo [8]. Cairo also supports transparencies, which can really spruce up the display of these annotations.

  The PDF specification requires any annotation to be either in a "closed" or "open" state. The default behavior can be controlled via Evince's "View" options. The state of each annotation may be toggled by clicking its corresponding icon located next to the object the annotation is attached to.

- Make further changes to the evince-form branch to add support for the creation, modification and deletion of annotations. This will be made possible by creating a temporary XML [9] file that represents these annotations, whose Schema will be formally defined during the course of the project. A "Save" and "Save As" functionality will be added to Evince, which, when called, will invoke the glib-poppler system to generate the final PDF with the annotations added. The temporary files created will be removed when Evince is closed. The user will be provided an interface in Evince to Add (Main Menu, Right-click), Modify (Double-click on existing annotation), and Delete (Main menu, Right-click, Delete key) annotations.

  This system will give the user the ability to undo/redo changes in annotations. When a PDF file with annotations will be opened in Evince, the temporary XML file will be created, and any changes in annotations will be made to this file and rendered on the screen. Changes will be written to the PDF document only when the file is "saved". (Note: Poppler's 'saveAs' action is sufficient to implement the described system). The libxml2 [10] library, already part of Gnome, will be used to interact with the file.

- Add the "tiger-bar" [11] functionality to Evince so that user's may locate annotations in the document easily. Also implement a shortcut key command to iterate over all annotations in a document as required by the PDF specification: a) In Row Order, b) In Column Order, and c) In Structure Order.

DELIVERABLES

At the end of the project, the user will be able to view, add, modify and delete the following types of annotations (as described in the PDF specification): Text, Free Text, Underline, Strike-Out, Line, Square, Circle, Highlight, and Popup. The first four types can be implemented as extensions to the current evince-form code, while the latter will be rendered from scratch using Cairo.

The code will be developed in a modular fashion so that it may be used later for adding generalized annotation support for other documents. The rendering methods and interaction method with the temporary XML file will be especially useful for this task. Although this Summer of Code proposal deals only with adding annotation support for PDF files, I am confident that the outcome will be most useful as a general framework for annotations in Evince; a task which I plan to personally take up on conclusion of the SoC.

When?

The project will be spaced out roughly as follows:

- April 09 - May 01 : Get friendly with the mentors and the Gnome Community ;-)
- May 01 - May 28 : Familiarize self with the evince-form branch, PDF specification and glib-poppler

- May 28 - Jun 15 : Begin! Update Glib bindings to Poppler adding annotation support
- Jun 15 - Jun 20 : Design Schema for temporary XML file representing annotations
- Jun 20 - Jun 30 : Modify evince-form branch to parse temporary XML file
- Jun 30 - Jul 15 : Add rendering functions to evince-form to display annotations
- Jul 15 - Jul 25 : Add functions to add, modify and delete portions of the XML file
- Jul 25 - Aug 05 : Add code to provide an interface for the user to add, modify and delete annotations
- Aug 05 - Aug 20 : Add "tiger-bar" and keyboard shortcut commands to iterate over annotations.

- Party!

Why Me?

I am an undergraduate student at the Malaviya National Institute of Technology, Jaipur, India; pursuing my Bachelor of Technology in Computer Engineering. I have been involved in the FOSS community for almost 3 years now, and have been an avid Gnome user for about the same time.

I am an active developer and documentation contributor in several open source projects [12], the most prominent of them being Gentoo Linux, GNU Parted and PHP-GTK. I became a Gentoo Developer as a result of my participation in last year's Summer of Code; during which I developed a web-based GuideXML editor: "Beacon" for the Gentoo Foundation. My work with PHP-GTK has given me a thorough understanding of the GObject-Glib-Gtk+ system, and I am confident that I will have no trouble in wading through the glib-poppler or Evince source code. In any case, I grasp concepts and implement things pretty quickly; skills which I have displayed in all my previous and current projects, and this will be no exception.

I had applied to six organizations last year, of which four selected me - the highest number of selections for a student in that year. Although I could officially perform only one of those projects, I am continuing work on the other three outside of the SoC. FOSS is something that is already very dear to me, and I can assure you that I will take full responsibility for the maintenance of the modules that result from this project even after the Summer of Code concludes. I take every Summer of Code as an opportunity to "infiltrate" and become part of another new community. Since I am already well-versed in the community dynamics of open source projects, I will have absolutely no trouble in mingling with the Gnome community and working with the infrastructure (Mailing Lists; Version Control Systems - I've extensively worked with CVS, SVN and Git; IRC etc.) already in place. In other words, I can get started almost immediately, giving me an effective coding time of almost 4 months, as opposed to the allotted 3. I sincerely hope that my work during the Summer of Code with Gnome will eventually lead me to become a full-time Gnome developer - something that I've always dreamed of!

You can find out more about me and what I do at my personal home page [13], and you also might want to look at my formal resume [14]. Please don't hesitate to get back to me if any part of this proposal is not clear to you. Thanks for considering this proposal and for your time!

REFERENCES

[1] http://live.gnome.org/Evince/Annotations

[2] http://www.adobe.com/devnet/pdf/pdf_reference.html

[3] http://www.gnome.org/projects/evince/

[4] http://live.gnome.org/Evince/Forms, http://svn.gnome.org/viewcvs/evince/branches/evince-form/

[5] http://poppler.freedesktop.org/

[6] http://webcvs.freedesktop.org/poppler/poppler/poppler/Annot.cc

[7] http://webcvs.freedesktop.org/poppler/poppler/qt4/src/poppler-annotation.cc

[8] http://cairographics.org/

[9] http://www.w3.org/XML/

[10] http://xmlsoft.org/

[11] http://www.gnome.org/~clarkbw/images/evince-bookmarks.png

[12] http://code.kix.in/

[13] http://www.kix.in/

[13] http://www.kix.in/personal/resume.pdf

The latest version of this proposal will be available at: http://www.kix.in/soc/07/evince-annotation-gnome.pdf