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

Docbook to PDF Convertor

A Summer of Code **Proposal**
Prepared for The Fedora Project
March 19th, 2007

Abstract

Several Open Source projects use Docbook as their primary documentation format. The format has been serving quite well for several years now, primarily because of the ability to transform Docbook based documents to a variety of other formats, and it's suitability for collaborative multilingual authoring. There are several Docbook-to-HTML convertors that provide HTML files of excellent quality. However, things on the Docbook-to-PDF front are not so rosy.

This project aims to exploit the versatility of XML by creating a generic, cross-platform Docbook to PDF convertor. Although there are some conversion methods already available, it is found that none of them provide an acceptable amount of control over the quality of output. The project described in this proposal, however, should provide a tool that is flexible enough to convert XML files based on Docbook into PDF files of high quality, the appearance of which can be customized by the end-user to a certain extent.

Why?

PDF [1] is now an open standard and provides users with a high quality representation of content suitable for both online viewing and printing. Providing documentation to end-users in the PDF format, is therefore very beneficial. However, given that most of the documentation efforts in several FOSS projects are based on Docbook [2]; it is imperative that a systematic and clean tool for conversion of Docbook files to their PDF equivalents is developed. This would not only benefit Fedora's users but also documentation efforts of all FOSS projects that use Docbook.

How?

There are several approaches to the Docbook-to-PDF conversion problem, however after evaluation of these methods [3], it is found that using ReportLab's RML [4] format as an intermediate achieves results of highest quality. ReportLab has been providing PDF services based on RML for enterprise customers successfully for quite sometime now. Although ReportLab's RML2PDF convertor is not free software, the emergence of free (as in speech) alternative tools such as OpenReport's TinyRML2PDF [5]; opens up opportunities for a completely free solution to the Docbook-to-PDF conversion problem.

The conversion can therefore be simplified into a 3-step process. The first step involves moving only the content of the Dobook file into a basic RML file. The second step would add styling to the content in the RML file, and the final step would produce the PDF. To elaborate; every RML file contains three basic sections [7]: <template>, <stylesheet> and <story>. The Docbook file can first be converted into equivalent <story> constructs with a default <template> section (while leaving the <stylesheet> section empty). If this RML file is converted to PDF as it is, the output derived would contain all the content, but in a rather bland format. Further refinement to the formatting and coloring of the content can be achieved by modifying the <template> and <stylesheet> portions of the document, which is achieved by the XSL transform in the second step. To Summarize:

- Convert the Docbook XML file to a skeletal RML file (via XSL [6] transforms)
- Apply a custom XSL stylesheet (provided by the user, based on a template) on the RML file
- Use a free RML2PDF convertor to obtain the final PDF

DELIVERABLES

At the end of the project, a set of XSLT files that perform conversion as detailed before will be ready. One set of stylesheets would convert a standard Docbook file to a bare-bones RML file, while the other set would apply a basic set of transformation on the RML file to add styling. The second set of files can serve as a guide to those users wanting to further refine the PDF output.

Also part of the deliverables is a complete software package that accepts a set of Docbook files as input and generates a PDF document as output. This package would contain (or depend on) an XSL processor (such as xsltproc), an XML validator (such as xmllint), an RML2PDF convertor (such as TinyRML2PDF), and a glue script (in Python) to integrate all these components together. Non-code deliverables would include documentation describing the working of the system and an end-user guide.

As a bonus, it would be useful to evaluate the possibility of integrating this system with Plone, the CMS used by Fedora. The Python glue script would make this quite easy, and would result in the ability to generate PDF documentation on the fly and offer it as downloads on the CMS.

When?

The project will be spaced out roughly as follows:

- April 09 - May 01 : Get friendly with the mentors and the Fedora Community ;-)
- May 01 - May 28 : Familiarize self with the Docbook and RML formats, warm-up exercises in XSLT

- May 28 - Jun 15 : Begin! First set of XSL transforms (Docbook to Skeleton RML)
- Jun 15 - Jun 30 : Second set of XSL transforms (Style additions to Skeleton RML)
- Jun 30 - Jul 15 : Create Glue script and assemble a stand-alone package for conversion

- Jul 15 - Jul 31 : Perform mass conversion of Fedora Documentation to PDF (Testing)
- Aug 01 - Aug 10 : Fine-tune stylesheets according to results of testing phase, evaluate Plone module
- Aug 10 - Aug 20 : Write developer and user documentation, finalize project

- 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.

I am an active developer and documentation contributor in several open source projects, 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" [8] for the Gentoo Foundation. The project involved quite a bit of XML weight-lifting, which means I am already quite familiar with the art of making XSL transforms happen - Beacon uses them to convert GuideXML to HTML and vice-versa. My documentation efforts with other projects have also given me a deep understanding of the Docbook Format.

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 software that results 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 Fedora 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.

You can find out more about me and what I do at my personal home page, and you also might want to look at my formal resume [9]. 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://www.adobe.com/devnet/pdf/pdf_reference.html

[2] http://www.docbook.org/

[3] http://thread.gmane.org/gmane.linux.redhat.fedora.documentation/5366

[4] http://www.reportlab.com/rml_index.html

[5] http://freshmeat.net/projects/trml2pdf/

[6] http://www.w3.org/Style/XSL/

[7] http://www.reportlab.com/docs/RML_UserGuide_1_0.pdf

[8] http://code.kix.in/projects/beacon

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

The latest version of this proposal will be available at: http://www.kix.in/soc/07/docbook2pdf-fedora.pdf