DocumentationTasks

One of the possibilities for the documentation (both technical and user) is to split it into a more interface independent format. I (RichardJones) have been playing around with the possibility of doing this in DocBook or some other semantic XML format. I have also looked at Apache Forrest and I am also considering TEI (as it's quite popular here in Bergen), although I haven't taken any time to get into this at the moment.

Forrest seems very clever, but I am concerned that it is overkill. Docbook is a common standard, and seems to work nicely, especially using the chunking stylesheet to create multi-page books separated into chapters and sections.

I converted some of the docs to docbook XML and then on to XHTML (download the source and results here: ^dspace-docbook.tar.bz2) as an experiment, and I quite like the result. I have tried converting the technical documentation into docbook as per:

http://wiki.docbook.org/topic/Html2DocBook

But there seems to be some issues with the table conversions after bringing the HTML 4.0 up to XHTML 1.0 (strict) using Tidy. I'm not sure what the problem is, but it would be a nuisance to have to do this all by hand in the first instance.

What do people think of these approaches, and does anyone have any other suggestions?

side note: the docbook licence appears acceptable for distribution with DSpace. It contains the wording: "Permission to use, copy, modify and distribute the DocBook XML DTD and its accompanying documentation for any purpose and without fee is hereby granted in perpetuity, provided that the above copyright notice and this paragraph appear in all copies."

(SY) Other alternatives?:

- Given that the University of Texas guys are going to be releasing the Cocoon UI, maybe Cocoon could be used for the transforms. I think both Forrest and Cocoon are both overkill for just the doco and help (since we'll then also have to keep an eye on the updates for an entire app) but since Cocoon will probably turn up in 1.5 we could always start using it for the doco.
- Alternatively we could also just have a servlet which given an entry point to the doco (some id) could run the stylesheet using the jaxp API. Once the Cocoon UI turns up, we could then move the doc transforms into Cocoon.
- build and deploy static xhtml at install time (can this be done via ant with jaxp/saxon/xalan?)
- it's a shame all browsers don't apply stylesheets in the way IE and Firefox does, then you could probably let the browser do all the transform work and just point it to the stylesheet! Maybe this will become standard over time.
- Leave as xhtml for now but add content-descriptive class attributes (this method can also allow you to convert to other formats, but keep the master format in a presentable way)