

Help System for GNUstep

overview and goals

Outline

- *Why?*
- *Different help possibilities*
- *HelpViewer*
- *Future*

Why?

- *While GNUstep aims to implement an OpenStep API, some of us want a GNUstep desktop ! (backbone)*
- *An help system is then obviously needed - it's a basic component of any user-friendly desktop environment*
- *We want to provide better help experience to the user*

Different help possibilities

- *Tooltips*
- *“Plain” on-line help books*
- *Contextual help*
- *System-wide help center*

Tooltips

- *Only for short descriptions of a widget (ie, use them for unlabelled buttons - see ProjectCenter for an example)*
- *Simply set and unset by the setToolTip: method of NSView*
- *Need to be incorporated in Gorm (...and in -gui)*

“Plain” on-line help books

- *It's the common method used to provide help in applications (MacOSX, Windows, KDE, etc.)*
- *Easily done with GNUstep: just provide a YourApp.rtf file in your application resources. But you don't have structured help, and a general RTF viewer isn't really ideal for browsing on-line help.*

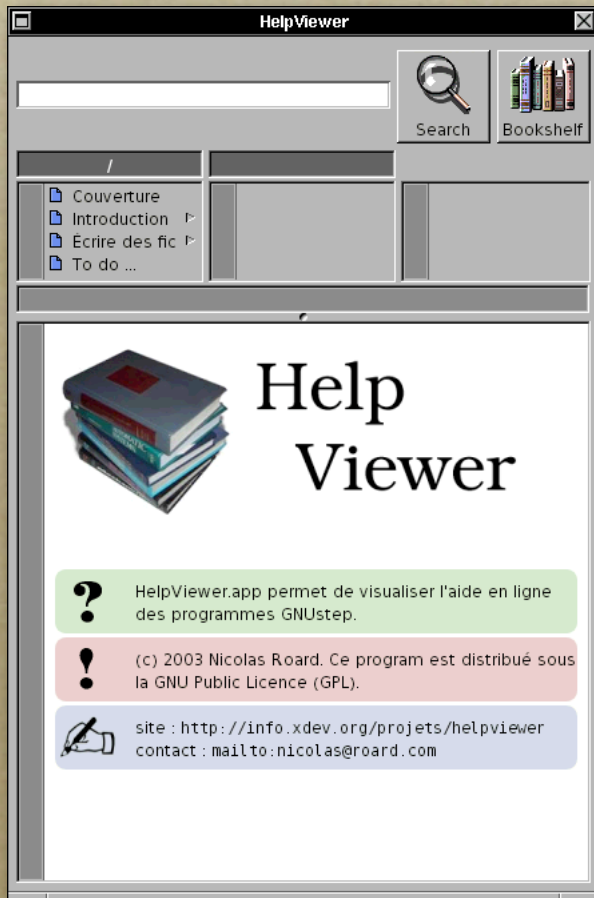
Contextual Help

- *Similar yet different than tooltips:*
 - *it should be used to provide more detailed help about the general use of the pointed widget*
 - *it points to a specific section of the help manual - the manual should appear on screen*
- *Not implemented...*

System-wide Help Center

- *We need to provide to the user a single entry-point for accessing help in addition to the applications info->help menu item*
- *Should provides access to common help formats used by non-gnustep applications: man, info, etc.*
- *Must be done in a separate application*

HelpViewer



It's a standalone GNUstep application used to present structured help.

You could already use it instead of RTF files.

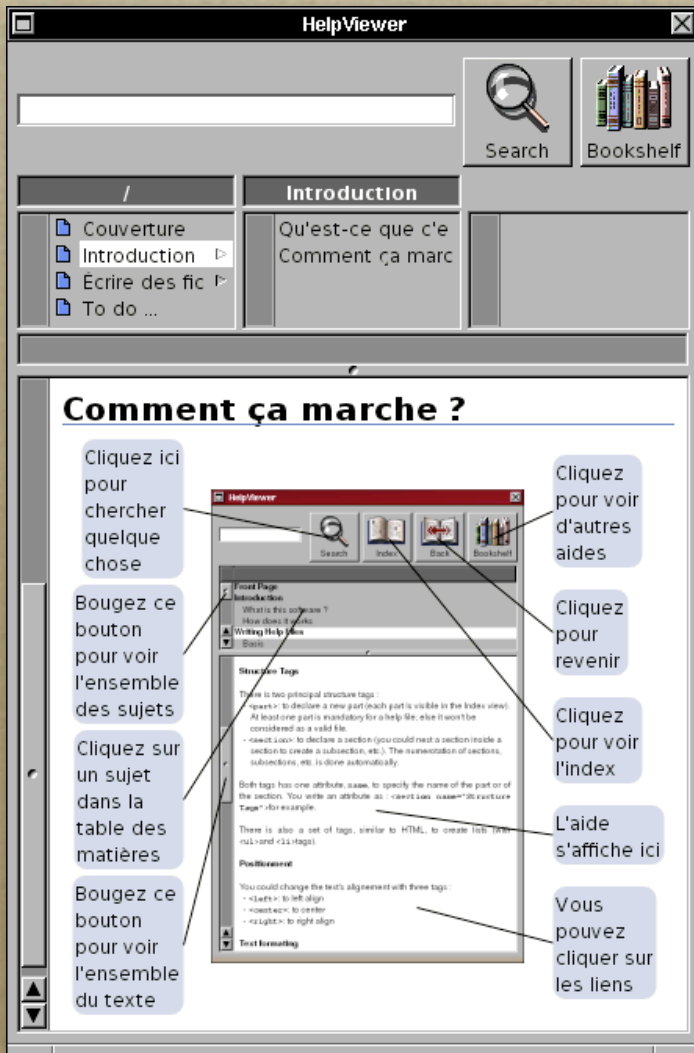
Help Viewer

- *Help books are bundles: you could then store in one entity your entire help manual in different languages and with its resources (images...)*
- *Help files are written using a simple XML syntax*

Help Viewer

- *Not only display help, but should help you to create good help books:*
 - *use of a standard help style*
 - *provides help-centered tags*

Help Viewer



<legendfig src="helpviewer-200.tiff">

<legend x="113" y="31">

Click here to see the index

</legend>

...

</legendfig>

Help Viewer

- *Problems:*

- *Yet another syntax! (and no graphic editor)*

- *It only reads .help format*

- *the current version lacks some feature previously implemented (and others that should be here as well :-)*

To sum up

- *HelpViewer is useable, but it still need some common features (mainly search and links...)*
- *It enhance the “plain help manual” possibility, but doesn’t provide contextual help nor could be used as a system-wide help application*

The Future

- *Some incremental ameliorations:*
 - *use of bundles to load other filetypes*
 - *add new help-centered tags (figures, references, footnote, etc.)*
 - *add parser for man and info files, possibly also display rtf files and html pages (using WebKit)*
 - *etc.*

The Future

- *More important design changes:*
 - *Move actual help rendering in a framework*
 - *Use that framework to implement NSHelpManager, Contextual Help in gui (would be perhaps done at a desktop-bundle level)*
 - *Implement a separate application for system-wide help using the framework*

Roadmap

- *Version 0.4 (march 2004)*
 - *use skyrix-xml instead of gsxml*
 - *add filetypes bundles*
 - *add search and links*

Questions?