



KDE And Accessibility

KDE Contributor Conference 2003 - “Kastle”

Gunnar Schmidt

University of Paderborn



Overview



- Definition of “Accessibility”
- Accessibility aids for KDE
- Technologies developed for possible inclusion in kdelibs
- Rules for Avoiding Accessibility Issues



What is “Accessibility”?



- Providing accessibility means to reduce barriers for handicapped people.
- A computer, an application or a web site is fully accessible if it can be used by all kind of different devices, including braille devices, screen readers, head point devices, special keyboards, buttons or mice, etc.
- The term accessibility is also more generally used to describe projects aiming at making computers fully accessible, or at offering computer software as a tool for dealing with physical handicaps.
- Technologies that assist the user with dealing with a handicap are often called assistive technologies or accessibility aids.



Accessibility Aids for KDE



Accessibility Features





Accessibility Features



- Many accessibility features are already included in the kdelibs and kdebase packages.
- These features are configurable through kcontrol.



kdeaccessibility



- The kdeaccessibility package is a set of assistive technologies for KDE.
- It currently contains KMagnifier, KMouseTool, and KMouth.
- Two accessibility-related programs currently developed in kdenonbeta are Klaviatura and the KDE Text-to-speech daemon (kttsd).



Technologies Developed for Possible Inclusion in kdelibs



Speech Synthesis



- The KDE Text-To-Speech Daemon (ktttsd) is currently developed in kdenonbeta as a text to speech DCOP service for KDE.
- ktttsd supports plug-ins for different text-to-speech systems. Currently it is possible to install different plug-ins for various languages.
- As ktttsd is not stable yet the DCOP APIs are not yet frozen. The plug-in APIs will most likely change when new features are added.
- It is planned that ktttsd will interoperate with gnome-speech.



Speech Synthesis API



- `kttsd` currently knows about warnings, messages and long texts.
- For issuing a warning you use
`kspeech::sayWarning (QString warning,
 QString language)`
- For issuing a message you use
`kspeech::sayMessage (QString message,
 QString language)`
- Currently there are 9 methods for handling long texts. These include starting, stopping and pausing the text and navigation within the text.



Word Completion



- The word completion currently is a part of KMouth.
- Base class for the word completion is the KWordCompletion class which is derived from the KCompletion class.
- The current version supports multiple word lists from which the user can choose which one he wants to use.



Word Completion API



- Support for multiple word lists:
 - `QStringList wordLists()`
 - `bool setWordList(QString wordlist)`
 - `QString currentWordList`
`(QString wordlist)`
- Language-related information for the word lists:
 - `QStringList wordLists(QString language)`
 - `QString languageOfWordList`
`(QString wordlist)`



Rules for Avoiding Accessibility Issues



General Rules



A good usability is especially important for people who use the computer slower or with different devices.

Examples are:

- Don't use hard-coded timeout values or other time-based functions.
- Don't show or hide information based on the movement of the mouse pointer. Exception: tooltips.
- Never use the same label two times in the same window. If the same label occurs in two different windows make sure that the corresponding objects serve for the same purpose.



Keyboard Navigation



- Provide keyboard shortcuts for all important features of the application.
- Make sure that all GUI elements can be handled by only using the keyboard.
- Provide a logical tab order.



Mouse Interaction



- Make sure that all actions can be performed by using the mouse.
- Make mouse targets large enough.
- Don't depend on input from mouse buttons 2 and 3.



Graphical Elements



- Don't hard-code graphic attributes such as line thickness, font sizes or colors.
- Don't use the color as the only way to distinguish information.



Final Remarks



- Making the KDE desktop accessible to all users is much work.
- Much can be achieved by using the existing features and applications and by following the rules of this presentation.
- Further than that it is possible to write different accessibility themes.
- In the long term we need to provide a service provider interface for assistive technologies.