Options menu Usability

Introduction

A menu is a window that presents a list of commands to the user. A menu is arranged into lines, known as menu items, each of which contains a text label. Menu provides a convenient way to navigate between different forms and views in an application. When a user selects a menu item, the command handler is invoked which handles the command pressed. Option menu is an efficient way to allow user to perform some actions, especially in the case of soft key based non touch devices.

The Options menu is a menu list displayed in a pop-up window. It is activated by pressing the Options soft key. (Options are the default value of the left soft key.)

Usability tips for options menu are

Group the menu items logically

Logical grouping of the menu items should be used so that the users can recongnize them easily.

The main tasks should be available quickly

Users should be able to locate the main task easily.

No deep menu

The menu structure should not be too deep or too shallow.

Menu should not cover whole screen

Ideally it should not cover more than 70% screen of the application. The maximum size is approximately the size of the standard main pane.

Good option menu example

Bad option menu example

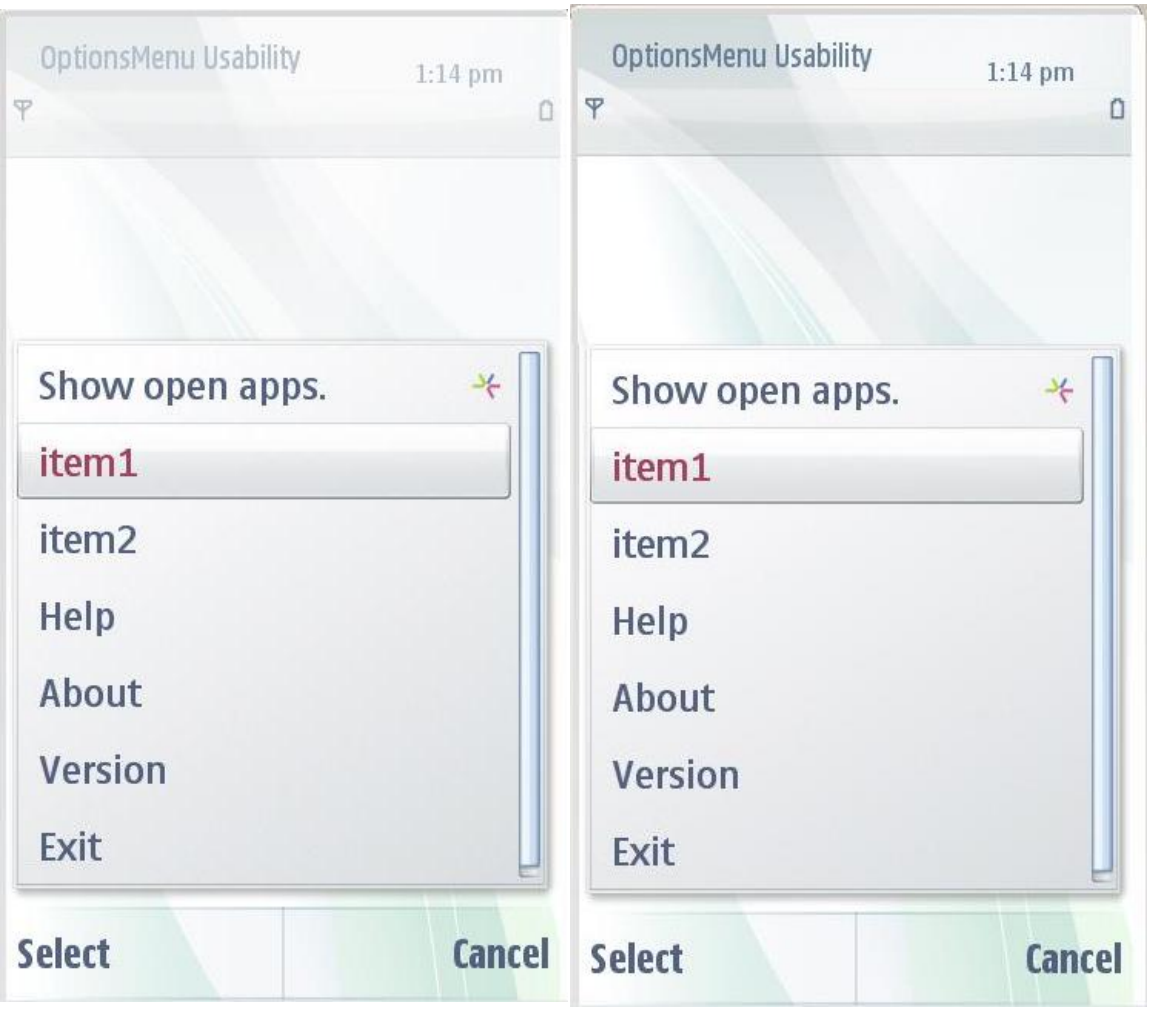
Dim background content

The content on the screen outside the menu pop-up should be dimmed, as otherwise they would hamper the usability of the menu, as the user's attention might be diverted.

Good option menu example

Bad option menu example

(C) Copyright Nokia 2014. All rights reserved.



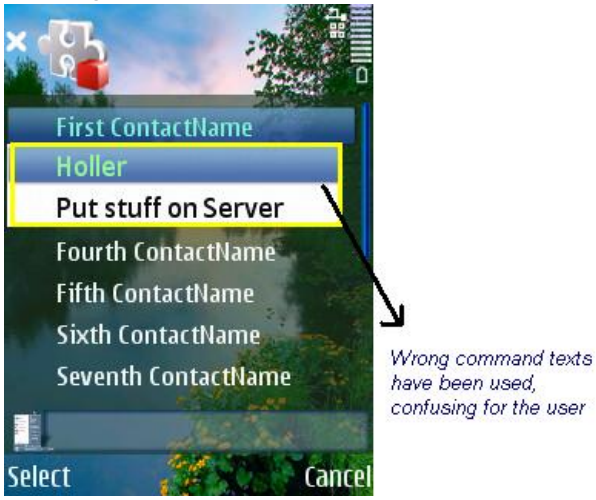
Prefer text over icon

Use icons on the menu only if it is very important or provides some useful information about the state of the application/menu etc, for instance radio buttons/checkboxes etc.

Use short and meaningful/understandable text

Where possible use the conventionally known/used words to denote a menu option command text. Do not use texts which are either too technical or too tough to understand for the end user. Also make sure that the command does what the user expects it to do, do not surprise the user.

Meaning-less text used



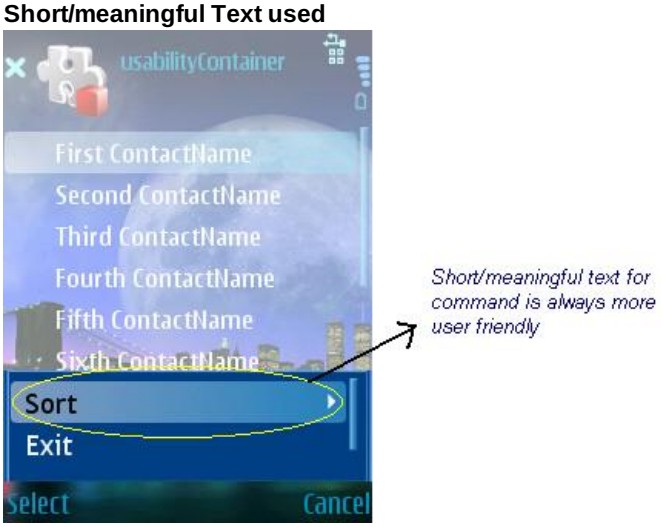
Meaningful Text used

(C) Copyright Nokia 2014. All rights reserved.

The text on the option menu should be short and they should never be truncated or ending with ellipsis. While doing usability testing it is imperative to test this aspect on various screen resolutions and modes.

Truncating text used

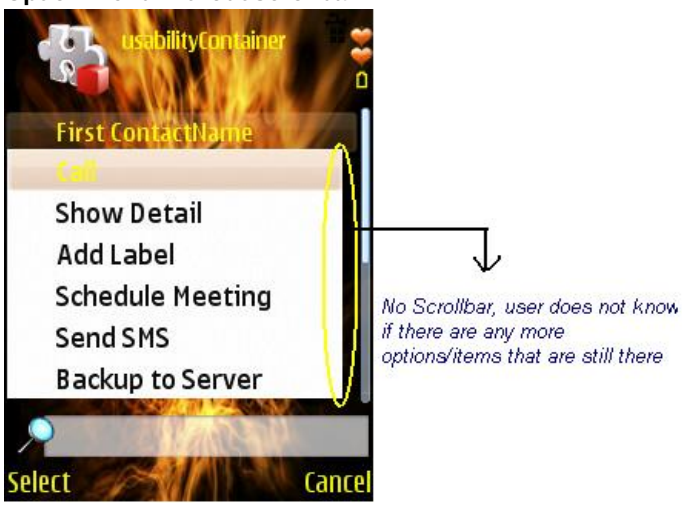
Short/meaningful Text used



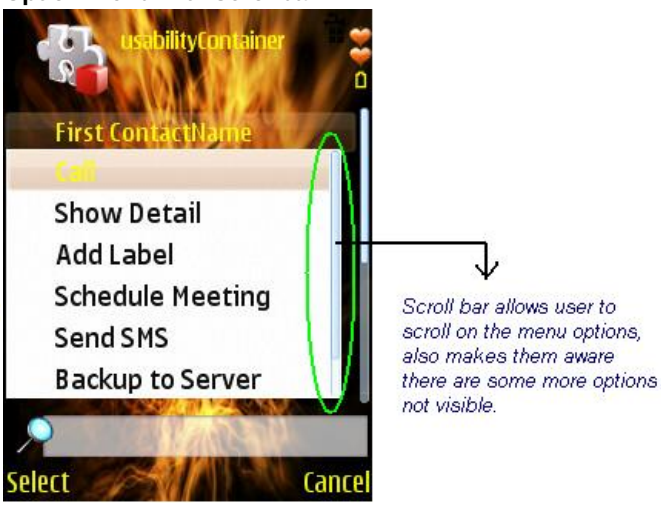
Provide scrollbar

The first choice should be to place only as many options on the menu as can be displayed at one time, without having to have a scrollbar for navigating up/down. In case it is not possible to avoid having more options than it is possible to display in one shot, a scrollbar should be provided to the user, that ways they would know there is some more menu option which is not visible right now.

Option menu without scrollbar

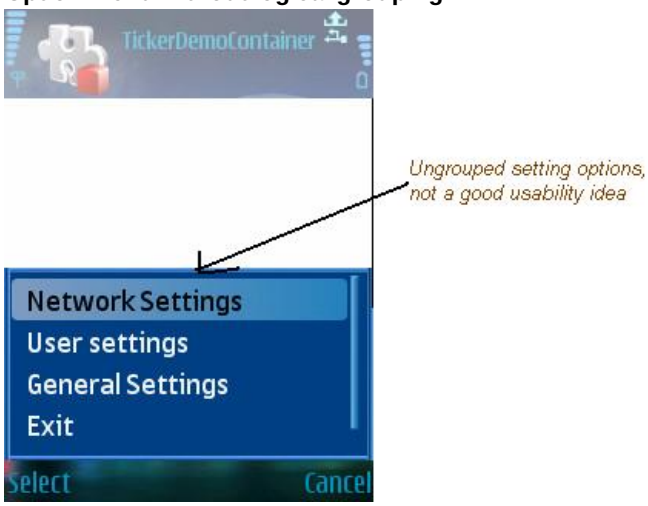


Option menu with scrollbar

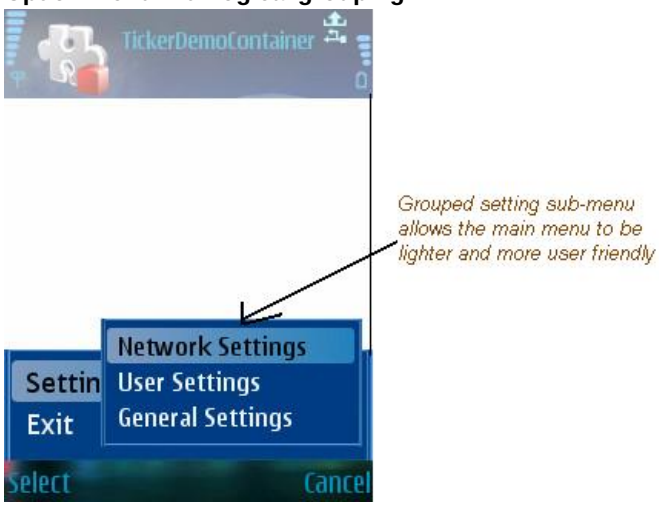


Add sub-menu

Option menu without logical grouping



Option menu with logical grouping

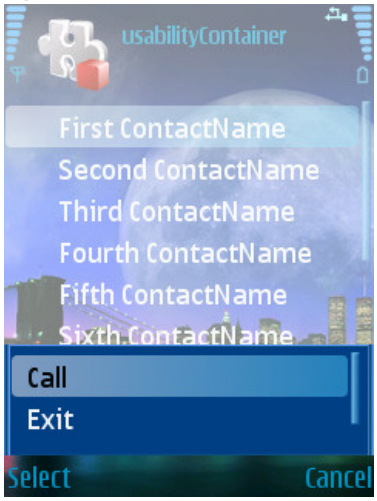


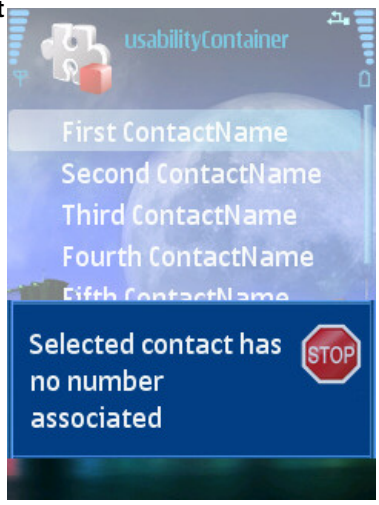
(C) Copyright Nokia 2014. All rights reserved.

Hide menu item based on state

Since the space available for showing the menu is limited it makes good usability sense to hide the items from the menu which are not relevant for a given state of the application. For example no need to show delete menu item if list does not contain any item, or provide option to download content when there is no connectivity etc.

Option menu with command not hidden as per context





Honor navigation key default action(s)

The navigations keys, Left Soft Key (LSK)/Right Soft Key (RSK) and Centre Soft Key (CSK) should follow the conventions, whereby the LSK should select the currently focused command; RSK should cancel the option menu and return the user to the view. The CSK should correspond to Select unless it is customized to bring up a context sensitive/drop down menu.

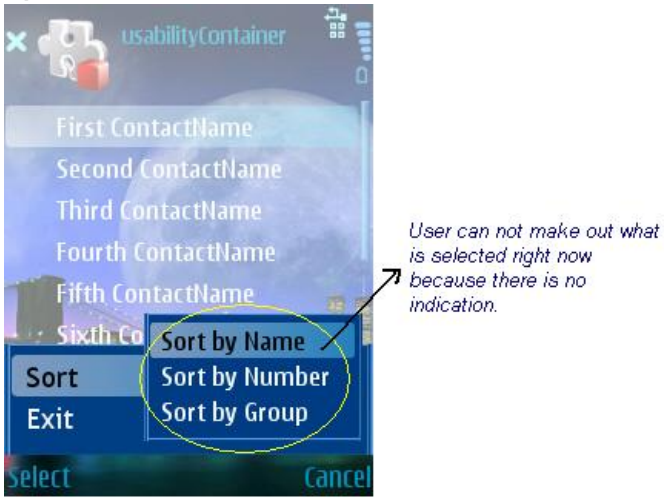
Provide Context-sensitive menu if required

Context-sensitive menus are secondary menus that are typically launched by pressing the Selection key (as opposed to the primary Options menu accessed via the left soft key). These menus are sensitive to the currently displayed view, and furthermore can be sensitive to the internal state of that view.

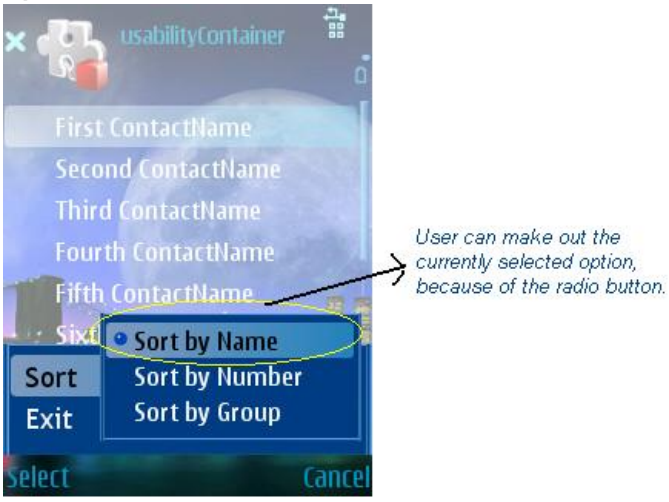
Use radio button/checkboxes only in the sub-menu

Use mark able menu options like radio button/checkboxes only in the sub-menu, especially if they are more like custom setting choices. For instance if you are giving the user choice to select a specific number of the 3 pre-defined numbers, you might want to show an indication against the number chosen. This way the user can know what is currently selected/chosen.

Option sub-menu without indication



Option sub-menu with indication



For details of drop down/context sensitive menu, check:-

How to display drop down/fly out menu

Loop the options menu

(C) Copyright Nokia 2014. All rights reserved.

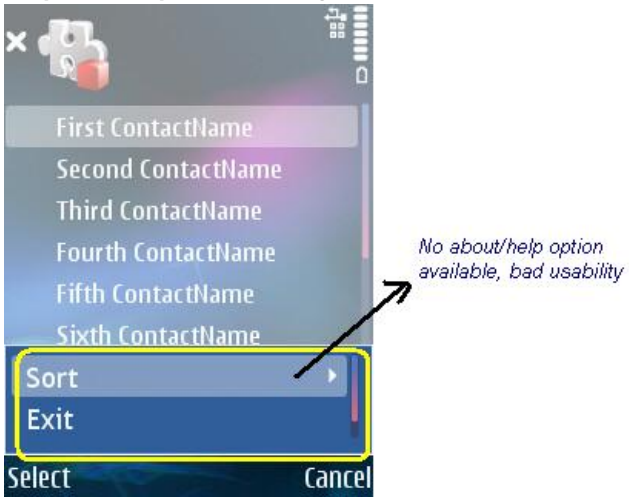
Provide a looping mechanism to allow accessing the top and bottom commands on the menu, this would save the user the time/effort to scroll up/down to get to the corresponding menu options.

Provide About/Help options

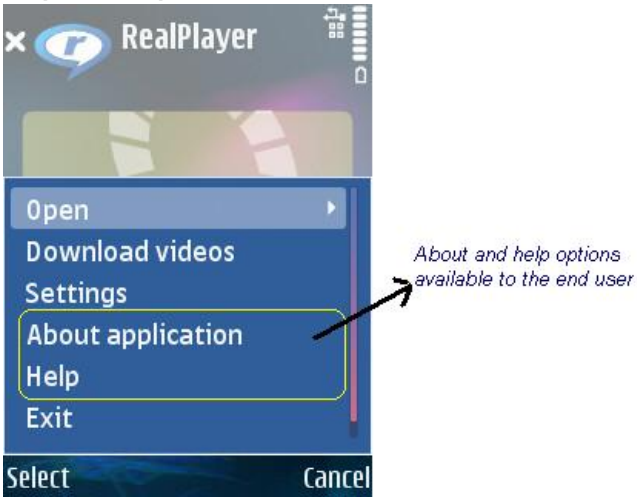
It is very important to provide the user a way to access the context sensitive through the application. The help would assist the user in making full use of the functionalities. User should also be able to know about the application, which should be accessible using the about option. More details can be had from *Things to remember when writing Help text or Manuals*. If the application has multi views, help should be possibly available on all view option menus, while about should be available only on the main view option menu.

An application without help/about options places a serious usability limitation.

Help/About options missing



Help/About options available



Sub-Menu guidelines

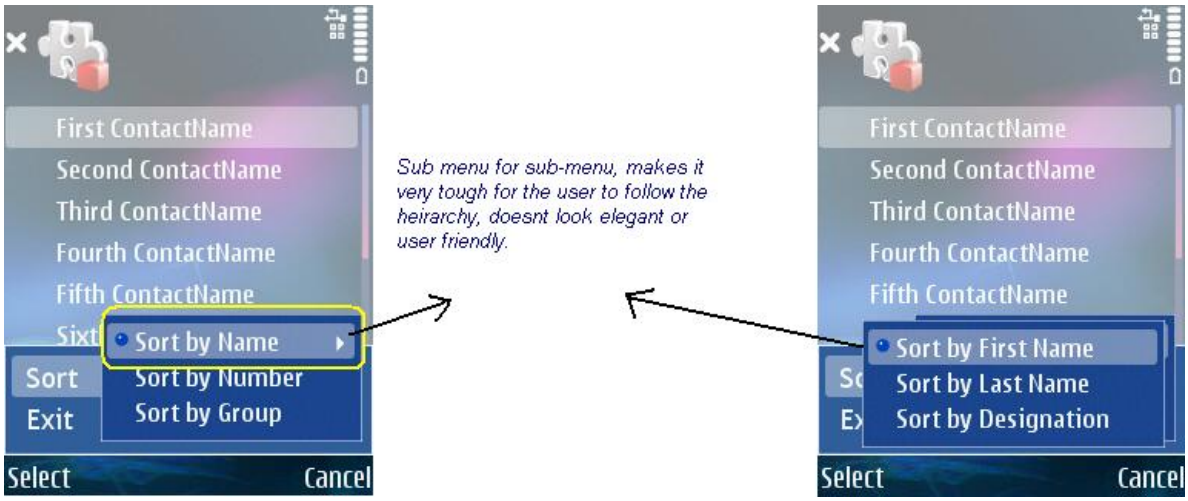
Minimize items

Minimize the number of items in the sub-menu; ideally speaking the sub-menu should always have lesser items than the main menu.

Avoid using submenus for sub-menus

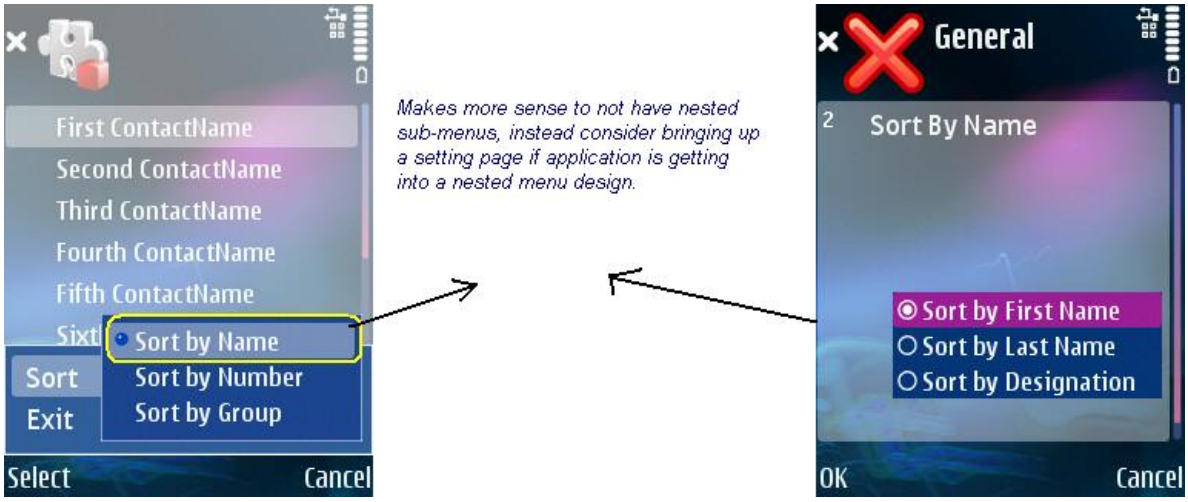
The deeper the navigational hierarchy, the more cumbersome it is from a user's perspective to be able to use them. Avoid using sub-menus for sub-menus, in those cases it would be more relevant to see how the menus can be re-arranged, or possibly breaking down the functionalities into multiple views with single level menu layout.

Nested menu structure, not usable



Single level menu structure, more usable

(C) Copyright Nokia 2014. All rights reserved.



--- Edited by Mayank on 26/06/2009 ---

(C) Copyright Nokia 2014. All rights reserved.