



Tasks analysis

Ronny, Charlotte MORICE, Zdenek NOVÝ,
Christophe PINCEMAILLE, Martin VÁVRA

November 12, 2008

Group: Desktop environment

University: Cork Institute of Technology

Department: Department of computer science

Module: Human Computer Interaction

Lecturer: Paul ROTHWELL

Contents

1	Virtual desktops	4
1.1	Activity description and tasks identification	4
1.2	Tasks analysis	4
1.2.1	Switching among Virtual Desktops	4
1.2.1.1	Identify the desktop where to go	4
1.2.1.2	Do the actual switching	5
2	Document management	7
2.1	Activity description and tasks identification	7
2.2	Tasks analysis	7
2.2.1	Creating a folder	7
2.2.1.1	Navigating to the right folder	8
2.2.1.2	Invoking the create folder command	8
2.2.1.3	Providing a name for the new folder	8
2.2.1.4	Actual creation of the folder or an error message	9
2.2.2	Copying a folder/document	9
2.2.2.1	Navigating to the source folder	10
2.2.2.2	Invoking the copy command	10
2.2.2.3	Navigating to the destination folder	10
2.2.2.4	Invoking the paste command	11
2.2.2.5	The actual copying process	11
2.2.3	Moving a folder/document	12
2.2.3.1	Navigating to the source folder	12
2.2.3.2	Invoking the cut command	13
2.2.3.3	Navigating to the destination folder	13
2.2.3.4	Invoking the paste command	13
2.2.3.5	The actual moving process	14
2.2.4	Removing a document/folder	14
2.2.4.1	Navigating to the right folder	15
2.2.4.2	Invoking the remove command	15
2.2.4.3	The actual removing process	15

3	Widgets management	17
3.1	Activity description and tasks identification	17
3.2	Tasks analysis	17
3.2.1	Widget adding	17
3.2.1.1	Go to the dashboard	18
3.2.1.2	Select the action of adding a widget	18
3.2.1.3	Select a widget to add	19
3.2.1.4	Validate the action	19
3.2.2	Widget removing	20
3.2.2.1	Go into the dashboard	20
3.2.2.2	Select the widget(s) to remove	20
3.2.2.3	Select the action of removing	21
3.2.2.4	Validate the action	21
3.2.3	Widget positioning	21
3.2.3.1	Go into the dashboard	21
3.2.3.2	Select a widget	22
3.2.3.3	Choose the final position of the widget	22
3.2.3.4	Validate the action	22
4	Applications	23
4.1	Activity description and tasks identification	23
4.2	Tasks analysis	23
4.2.1	Installing an application	23
4.2.1.1	Go into the repository	23
4.2.1.2	Select an/some application(s) to install	24
4.2.1.3	Drop them to the application folder	24
4.2.2	Launching an application	25
4.2.2.1	Go into the menu containing all the applica- tions installed/Go into a directory containing documents linked to a particular application	25
4.2.2.2	Select the application/file	25
4.2.3	Removing of an application	26
4.2.3.1	Go into the installed application folder	26
4.2.3.2	Select an application to remove	26
4.2.3.3	Drop it to the bin	27
5	User switching	28
5.1	Activity description and tasks identification	28
5.2	Tasks analysis	28
5.2.1	Logging out	28
5.2.1.1	Invoking logout command	29

<i>CONTENTS</i>	3
5.2.2 Logging in	29
5.2.2.1 Logging in	29
A Diagrams	I

Chapter 1

Virtual desktops

1.1 Activity description and tasks identification

The desktop environment supports virtual desktops. So there are several desktops, in which the user can open windows. The user may want to switch between the desktops, in order to have access to the different applications, opened within the different desktops. So, in that activity, we have only one task: switching among virtual desktops.

For the diagram, cf. figure A.1 on page I.

1.2 Tasks analysis

1.2.1 Switching among Virtual Desktops

1. Identify the desktop where to go;
2. Do the actual switching.

1.2.1.1 Identify the desktop where to go

1.2.1.1.1 Information that the user must supply

- *How does the system know in which desktop the user wants to go?*

1.2.1.1.2 Information that the software must supply

- *How does the user know in which virtual desktop is the window he wants to see?* First, we may suppose that the user knows which windows he

has opened in which desktop: the user's memory is already a first source of information. But we can also assume that the user can forget things, so we have to provide it a system which would make the user able to know which windows are on each desktop. This can be a list of opened application by desktop, or sort of a preview of the desktops...

- *How many desktops are there?* To make the user know how many desktops there are, the software has to provide some information. If there are enough few desktops (4 or less), the desktops can be represented and viewable any time: by numbered icons in a status bar, by thumbnails, and so forth. If there are more than 4, an icon can be displayed with the number of desktops.
- *How to identify the desktops ?* The desktops can be represented in an icon, or in thumbnails, or in another way. The important thing is that they must be placed in a certain way, for example two "on the top" and two "on the bottom", and must be placed like that everywhere where the desktop representations are. Like that, the user can make a mental model where he knows there is one desktop here, one desktop there, ...
- *How to bring a good mental model to the user, helping him to organize the windows ?* The way for the user to organize the windows can be facilitated by the desktop environment: it can allow to put names on each desktop, so that the user can put windows on the desktop in consequence. For example, the user can name one desktop "music", and then everything related to music (music player, ...) will be placed there. Or another can be named "the Internet", and everything related to the Internet will be placed there (web browser, chat windows, email client...).

1.2.1.2 Do the actual switching

1.2.1.2.1 Information that the user must supply

- *How the system knows to which desktop going?* To provide that information, the user just have to click on the representation of the desktop. It can be the icon of the wanted desktop, or its virtual shape / representation...
- *Is there more than one mean to switch?* The mouse switching is probably the most straightforward, but it can be interesting to provide more than one mean. In particular, the possibility to switch by a keyboard

shortcut is very interesting, because it is definitely quicker, and will be appreciated by advanced user. So a certain key combination, like `ALT + Fi`, where `i` is the desktop number, will notify the system to go to the desktop `i`.

1.2.1.2.2 Information that the software must supply

- *What happens if the user chooses the current desktop?* The best way for that is to do nothing. It is not an error, it must be a slip, and it will certainly be annoying if the computer gives a feed-back to the user (like a message box, or something like that).
- *How the user know the desktop change has been made?* By changing of desktop, the windows change, so the user is already notified that the change has been made. But we can also considerate the case where there are not any windows opened. In that case, there are means to give feedback to the user. In the icons, the current desktop can be represented in another color or shape. Moreover, a very quick signal can be printed on the screen, and immediately removed, indicating the desktop change is progressing (like under Mac OS X).

Chapter 2

Document management

2.1 Activity description and tasks identification

The user needs to manage his/her documents. To keep the documents well arranged the desktop environment has to provide the user with the following functions: creating/moving/copying/removing folders and moving/copying/removing documents. The documents are placed in the folders.

1. creating a folder;
2. copying a folder/document;
3. moving a folder/document;
4. removing a folder/document.

For the diagram, cf. figure A.3 on page II.

2.2 Tasks analysis

2.2.1 Creating a folder

The user can create a new folder in another folder in order to keep his/her documents well arranged.

1. Navigating to the right folder;
2. invoking the create folder command;

3. providing a name for the new folder;
4. actual creation of the folder or an error message.

2.2.1.1 Navigating to the right folder

2.2.1.1.1 Information that the user must supply

- *How does the user provide the system with the information where the new folder is to be created?* The user simply navigates through the hierarchical folder structure all the way to the folder where the new folder is to be created.

2.2.1.1.2 Information that the software must supply

- *How can the user determine he has navigated to the desired folder?* The system provides the user with the the path to the current folder and also with a listing of documents and folders that is contains.

2.2.1.2 Invoking the create folder command

2.2.1.2.1 Information that the user must supply

- *How does the system find out the user wants to create a new folder?* The user chooses a way of invoking the creation process from the options provided by the desktop environment (see below).

2.2.1.2.2 Information that the software must supply

- *How does the system inform the user he/she does not have sufficient permission for creating a folder?* The option to create a new folder is not available (will be disabled).
- *How can the user start the actual creation process?* The desktop environment with different ways to invoke the creation command: context menu, a keyboard shortcut or an icon in the file manager toolbar.

2.2.1.3 Providing a name for the new folder

2.2.1.3.1 Information that the user must supply

- *How does the system ensure that the newly created folder has the name the user wants?* The user inputs the name of the folder or keeps the default name offered by the system (e.g. “New folder”).

2.2.1.3.2 Information that the software must supply

- *How does the system get the name of the new folder?* The system shows a dummy new folder with a default name (e.g. “New folder”) and allows the user to rename it. The default folder name will be marked so the user can simply start writing the new name.
- *How does the system handle characters that are not allowed?* The best way to avoid characters that are not allowed in a folder name is that pressing the key of an invalid character has no effect.

2.2.1.4 Actual creation of the folder or an error message

2.2.1.4.1 Information that the user must supply All the information have already been supplied by the user.

2.2.1.4.2 Information that the software must supply

- *How does the user find out the folder was already created?* The system shows the new folder in the current folder having the name inputted by the user and no error messages are shown.
- *How does the user find out there were any problems while creating the folder?* The problems can occur if the folder of the same name already exists or if there happen to be a system error.

2.2.2 Copying a folder/document

The user may want to copy a document or a folder or a set of these to some different location.

1. Navigating to the source folder;
2. invoking the copy command;
3. navigating to the destination folder;
4. invoking the paste command;
5. the actual copying process.

2.2.2.1 Navigating to the source folder

2.2.2.1.1 Information that the user must supply

- *How does the user provide the system with the information about the folder which contains the documents and/or folders to be copied?* The user simply navigates through the hierarchical folder structure all the way to the folder which contains the documents and/or folders to be copied.

2.2.2.1.2 Information that the software must supply

- *How can the user determine he has navigated to the desired folder?* The system provides the user with the the path to the current folder and also with a listing of documents and folders that is contains.

2.2.2.2 Invoking the copy command

2.2.2.2.1 Information that the user must supply

- *How does the system find out the user wants to copy files and/or folders?* The user chooses a way of invoking the copy process from the options provided by the desktop environment (see below).

2.2.2.2.2 Information that the software must supply

- *How does the system inform the user he/she does not have sufficient permission for copying files and/or folders?* The option to copy files and/or folders is not available (will be disabled).
- *How does the system find out that the user wants to copy files and/or folders?* The user chooses an option from a context menu or presses a keyboard shortcut or simply starts dragging the selected items.
- *How does the system find out which files and/or folder the user wants to copy?* The user selects the files and/or folders before he/she chooses the option to copy them.

2.2.2.3 Navigating to the destination folder

2.2.2.3.1 Information that the user must supply

- *How does the user provide the system with the information about the folder which should contain the files and/or folders being copied?* The user simply navigates through the hierarchical folder structure all the way to the folder which should contain the files and/or folders being copied.

2.2.2.3.2 Information that the software must supply

- *How can the user determine he has navigated to the desired folder?* The system provides the user with the the path to the current folder and also with a listing of documents and folders that is contains.

2.2.2.4 Invoking the paste command

2.2.2.4.1 Information that the user must supply

- *How does the system find out the user wants to paste files and/or folders?* The user chooses a way of invoking the paste command from the options provided by the desktop environment (see below).

2.2.2.4.2 Information that the software must supply

- *How does the system inform the user he/she does not have sufficient permission for pasting files and/or folders?* The option to paste files and/or folders is not available (will be disabled).
- *How does the system find out that the user wants to paste files and/or folders?* The user chooses an option from a context menu or presses a keyboard shortcut or simply drops the selected items. When the items being dragged are dropped the desktop environment shows a menu from which the user selects the “copy” command.

2.2.2.5 The actual copying process

2.2.2.5.1 Information that the user must supply All the information have already been supplied by the user.

2.2.2.5.2 Information that the software must supply

- *How does the user find out the files and/or folders were already copied?* The system shows the copied files and/or folders in the destination folder.

- *What happens if there already exists a file of a same name as one of the files being copied?* The user will be provided with a list of options. Besides the common options like “Overwrite” or “Overwrite all” the user will be able to ask the system to compare the files by content. The result of the comparing process would be in a form of “yes” (files are identical) or “no” (files are different).
- *What happens if there already exists a folder of a same name as one of the folders being copied?* The user will be provided with a list of options containing “Merge” or “Replace”.
- *What happens if the destination drive is full?* The user will be informed about this situation and the whole operation will be rolled back.

2.2.3 Moving a folder/document

The user may want to move a document or a folder or a set of these to some different location.

1. Navigating to the source folder;
2. invoking the cut command;
3. navigating to the destination folder;
4. invoking the paste command;
5. the actual moving process.

2.2.3.1 Navigating to the source folder

2.2.3.1.1 Information that the user must supply

- *How does the user provide the system with the information about the folder which contains the documents and/or folders to be moved?* The user simply navigates through the hierarchical folder structure all the way to the folder which contains the documents and/or folders to be moved.

2.2.3.1.2 Information that the software must supply

- *How can the user determine he has navigated to the desired folder?* The system provides the user with the path to the current folder and also with a listing of documents and folders that is contains.

2.2.3.2 Invoking the cut command

2.2.3.2.1 Information that the user must supply

- *How does the system find out the user wants to move files and/or folders?* The user chooses a way of invoking the move process from the options provided by the desktop environment (see below).

2.2.3.2.2 Information that the software must supply

- *How does the system inform the user he/she does not have sufficient permission for moving files and/or folders?* The option to move files and/or folders is not available (will be disabled).
- *How does the system find out that the user wants to move files and/or folders?* The user chooses an option from a context menu or presses a keyboard shortcut or simply starts dragging the selected item.
- *How does the system find out which files and/or folders the user wants to move?* The user selects the files and/or folders before he/she chooses the option to move them.

2.2.3.3 Navigating to the destination folder

2.2.3.3.1 Information that the user must supply

- *How does the user provide the system with the information about the folder which should contain the files and/or folders being moved?* The user simply navigates through the hierarchical folder structure all the way to the folder which should contain the files and/or folders being moved.

2.2.3.3.2 Information that the software must supply

- *How can the user determine he has navigated to the desired folder?* The system provides the user with the the path to the current folder and also with a listing of documents and folders that is contains.

2.2.3.4 Invoking the paste command

2.2.3.4.1 Information that the user must supply

- *How does the system find out the user wants to paste files and/or folders?* The user chooses a way of invoking the paste command from the options provided by the desktop environment (see below).

2.2.3.4.2 Information that the software must supply

- *How does the system inform the user he/she does not have sufficient permission for pasting files and/or folders?* The option to paste files and/or folders is not available (will be disabled).
- *How does the system find out that the user wants to paste files and/or folders?* The user chooses an option from a context menu or presses a keyboard shortcut or simply drops the selected items. When the items being dragged are dropped the desktop environment shows a menu from which the user selects the “move” command.

2.2.3.5 The actual moving process

2.2.3.5.1 Information that the user must supply All the information have already been supplied by the user.

2.2.3.5.2 Information that the software must supply

- *How does the user find out the files and/or folders were already moved?* The system shows the moved files and/or folders in the destination folder.
- *What happens if there already exists a file of a same name as one of the files being moved?* The user will be provided with a list of options. Besides the common options like “Overwrite” or “Overwrite all” the user will be able to ask the system to compare the files by content. The result of the comparing process would be in a form of “yes” (files are identical) or “no” (files are different).
- *What happens if there already exists a folder of a same name as one of the folders being moved?* The user will be provided with a list of options containing “Merge” or “Replace”.
- *What happens if the destination drive is full?* The user will be informed about this situation and the whole operation will be rolled back.

2.2.4 Removing a document/folder

The user may want to remove some of his documents and/or folders.

1. Navigating to the right folder;
2. invoking the remove command;
3. the actual removing process.

2.2.4.1 Navigating to the right folder

2.2.4.1.1 Information that the user must supply

- *How does the user provide the system with the information about the folder which contains the documents and/or folders to be removed?* The user simply navigates through the hierarchical folder structure all the way to the folder which contains the documents and/or folders to be removed.

2.2.4.1.2 Information that the software must supply

- *How can the user determine he has navigated to the desired folder?* The system provides the user with the path to the current folder and also with a listing of documents and folders that is contains.

2.2.4.2 Invoking the remove command

2.2.4.2.1 Information that the user must supply

- *How does the system find out the user wants to remove files and/or folders?* The user chooses a way of invoking the remove command from the options provided by the desktop environment (see below).

2.2.4.2.2 Information that the software must supply

- *How does the system inform the user he/she does not have sufficient permission for removing files and/or folders?* The option to remove files and/or folders is not available (will be disabled) or a “no entry” sign will be shown if the user decided to remove the file by the drag&drop way. If the user used a keyboard shortcut for invoking the remove command while not having sufficient permissions an error message box will be shown to the user.
- *How does the system find out that the user wants to remove files and/or folders?* The user chooses an option from a context menu or presses a keyboard shortcut or simply drops the selected items into the Recycle bin.

2.2.4.3 The actual removing process

2.2.4.3.1 Information that the user must supply All the information have already been supplied by the user.

2.2.4.3.2 Information that the software must supply

- *How does the user find out the files and/or folders were already removed?* The system does not show the removed files and/or folders in the folder.

Chapter 3

Widgets management

3.1 Activity description and tasks identification

In the desktop environment, there is the possibility to insert widgets. These are little applications which provide some services, such as the weather, a calculator, date and time, and so forth. There are several tasks that can be done.

1. adding a widget;
2. moving a widget;
3. delete a widget.

For the diagram, cf. figure A.4 on page III.

3.2 Tasks analysis

3.2.1 Widget adding

The user can add widgets in his dashboard.

1. Go to the dashboard;
2. select the action of adding a widget;
3. select a widget to add;
4. validate the action.

3.2.1.1 Go to the dashboard

3.2.1.1.1 Information that the user must supply

- *How the user can notify the system that he wants to go to the dashboard?* Here, there can be several means. The first is by the mouse: by putting the mouse on a corner or a certain part of the screen, the dashboard opens. Thanks to that, it can be very quick to open the dashboard. But this possibility should be parametrized (and removed), because it can appear annoying to users who are not used to that (and make the dashboard appear when they don't want). Then, there can be a keyboard shortcut to make the dashboard appear, which can be also very useful for users who only use the keyboard. Finally, there can be an icon in the status bar, which opens the dashboard.

3.2.1.1.2 Information that the software must supply

- *In the case of it takes too much time, how to notify the user the system is going to the dashboard?* In that case, the system opens a textbox with a message, and an estimation of the remaining loading time. It is important here to provide feedback to the user, so that he can see that the system is available, and that there is no problem.
- *How to notify the user that the dashboard has been opened?* The dashboard is very different of the work space (the current desktop), so this is straightforward. The feedback is provided by the display of the dashboard itself.

3.2.1.2 Select the action of adding a widget

3.2.1.2.1 Information that the user must supply

- *How does the user notify the software of the action he wants to do ?* By the mouse, by clicking on a reserved space on the dashboard ("add a widget"). This could be also possible by keyboard shortcut.

3.2.1.2.2 Information that the software must supply

- *How does the user know that the action is available?* If the action is not available, it appears in grey, showing that the function is inactivated.
- *What if the widget have already been added?* The widget application can enable several instances of the same widget. If it is the case, the

widget icon appears like the others. If it is not the case, the widget icon appears like an inactivated widget. This provides the good information to the user.

3.2.1.3 Select a widget to add

3.2.1.3.1 Information that the user must supply

- *How does the software know the widget to add?* The user will make a click on the widget he wants to add. Thanks to that, the software knows exactly which widget to add.
- *How does the software can be sure that the user has provided the good information?* It is always possible that the user slips, and click the wrong widget. But in spite of that, there is no alert when clicking on a widget to add it, because it would become very heavy and annoying to add a widget. So no window “Are you sure to add this widget?”.

3.2.1.3.2 Information that the software must supply

- *Is there any repository containing the list of the widget available?* Yes, the desktop environment will provide a repository manager, in order for the user to be able to add easily a widget.
- *How does the user know what widgets are available?* In the repository, there can be an icon, showing which widgets are available, which are not.
- *Does the user can add a link to indicate where to find more widgets?* A link in the repository provides the user the information to where find other widget. This gives the possibility to add new repositories, for example.

3.2.1.4 Validate the action

3.2.1.4.1 Information that the user must supply There is no really a task or an information the user has to send to validate the action. The adding validation has already occurred by the widget selection.

3.2.1.4.2 Information that the software must supply

- *What if the user does not have necessary grants to add widgets?* Then, the widget should have appeared as a inactivated widget. So in that stage of the process, this should not happen.

- *What if there is no enough space to add supplementary widget?* The computer, when displaying the widgets, should calculate the required space (or get it from the widget, more likely). When the widget is too large to be added, the system should make it appear inactivated, but with another color (red for example). Like that, the widget cannot be added (and then, no annoying alert window), and still, the user has the information why it can't be added.

3.2.2 Widget removing

The user can remove widgets of his dashboard.

1. Go into the dashboard;
2. Select the widget(s) to remove;
3. Select the action of removing;
4. Validate the action.

3.2.2.1 Go into the dashboard

This is exactly the same as in the widget adding (cf. subsection 3.2.1.1 on page 18).

3.2.2.2 Select the widget(s) to remove

3.2.2.2.1 Information that the user must supply

- *How the user can select the widget he wants?* By clicking on a widget, this is selected.
- *Does the user can deselect widgets?* It should be possible to deselect widget, by clicking again on it.

3.2.2.2.2 Information that the software must supply

- *How the user know that a widget has been select?* When a widget is selected, the system should make it appear differently: clearer, or with a distinct border. . .
- *What feedback is provided when a widget is deselected?* Then, the widget comes back to it original appearance state.

3.2.2.3 Select the action of removing

3.2.2.3.1 Information that the user must supply

- *How the system knows which widget remove?* By selecting the good widget, the user has given that information.
- *How the computer knows that a remove is asked?* There should be a remove button, in which the user can click. Then, the system knows that a removing is asked. That should be also possible by keyboard shortcut.

3.2.2.3.2 Information that the software must supply

- *How does the user knows the actions available?* When selecting the widget, the different actions are displayed, by displaying the buttons (for example, the presence of the remove button shows that removing a widget is possible).

3.2.2.4 Validate the action

3.2.2.4.1 Information that the user must supply There is no really a task or an information the user has to send to validate the action.

3.2.2.4.2 Information that the software must supply

- *How to know the widget has been removed?* The widget doesn't appear any more, that provides the sufficient feedback.

3.2.3 Widget positioning

The user can move widgets on his dashboard

1. Go into the dashboard;
2. Select a widget;
3. Choose the final position of the widget;
4. Validate the action.

3.2.3.1 Go into the dashboard

This is exactly the same as in the widget adding (cf. subsection 3.2.1.1 on page 18).

3.2.3.2 Select a widget

This is exactly the same as in the widget removing (cf. subsection 3.2.2.2 on page 20).

3.2.3.3 Choose the final position of the widget

3.2.3.3.1 Information that the user must supply

- *How does the user inform the system of the new position?* The user makes a drag-and-drop with the widget, to the final place.

3.2.3.3.2 Information that the software must supply

- *How the user can follow the drag-and-drop of the widget?* When the drag-and-drop is performed, the system makes a grey representation of the widget, to create the mental model of dragging the object.
- *What if a widget is already present at the position?* To make things easier, we will consider that widget can not overlap. Consequently, a widget can't be put into the position of another widget; when moving to it, the widget is stopped by the other already there.
- *Do all the widget can be moved? (e.g Any possibility to block them?)* In our desktop environment, we will consider that all widgets can be moved. But it would be a further improvement to make, with the adapted feedback to the user.

3.2.3.4 Validate the action

3.2.3.4.1 Information that the user must supply

- *What if the user change his mind? Any possibility undo the action?* It could be possible by a keyboard shortcut, but the “normal” method is to to it again by moving the widget to its original place.

3.2.3.4.2 Information that the software must supply

- *How does the user know that the widget has been moved?* When finalizing the move (end of the drag-and-drop), the system makes the widget appear in it new place, and disappear from its original place. Like that, the user has the appropriate feedback and knows that the action has been correctly done.

Chapter 4

Applications

4.1 Activity description and tasks identification

The desktop environment allows to manipulate applications. Three tasks are involved :

1. Installing an application;
2. launching an application;
3. removing an application.

For the diagram, cf. figure A.5 on page III.

4.2 Tasks analysis

4.2.1 Installing an application

1. Go into the repository;
2. select an/some application(s) to install;
3. drop them to the application folder.

4.2.1.1 Go into the repository

4.2.1.1.1 Information that the user must supply

- *How does the user notify the software that he wants to open the repository?* An icon is displayed in the tool bar and a keyboard shortcut is also provided.

4.2.1.1.2 Information that the software must supply

- *How does the user know that his action has been understood by the software?* A list of all the applications in the repository is displayed. This list is sorted by categories.

4.2.1.2 Select an/some application(s) to install

4.2.1.2.1 Information that the user must supply

- *How does the user notify the software which applications/files he want to select?* The user selects an application or a file with a click on its icon. If he wants to select several files he needs to use the keyboard (Ctrl/Shift) or the mouse (drawing of a rectangle).

4.2.1.2.2 Information that the software must supply

- *How does the user know if an application is already installed?* Applications which are already installed in the user's machine will have a sign on their icons (like a small green check sign or anything).
- *How does the user know if he has the grants to install an application?* If the user can not install an application, all icons will be in a grey tint, so the user will know that actions are inactivated.

4.2.1.3 Drop them to the application folder

4.2.1.3.1 Information that the user must supply

- *How does the user indicate to the software the folder where to put the applications?* The user has to put the mouse over the icon of the folder.

4.2.1.3.2 Information that the software must supply

- *How does the user actually know when he can release his mouse for his action to be taking into consideration?* An animation on the icon of the application folder occurs.
- *What happens if the user drops an application which is already installed?* A message will be displayed, allowing the user either to cancel his action or to continue to do a reinstallation of the application.
- *How does the user know that the application has actually been installed?* An icon of the application will appears inside the installed application folder.

4.2.2 Launching an application

An application can be directly launched by clicking an item in a hierarchical menu or launched by opening a document or dynamic menu.

1. Go into the menu containing all the applications installed/Go into a directory containing documents linked to a particular application;
2. Select the application/file;

4.2.2.1 Go into the menu containing all the applications installed/Go into a directory containing documents linked to a particular application

4.2.2.1.1 Information that the user must supply

- *How does the user notify the software that he wants to open the menu?* An icon is displayed in all the virtual desktops and it is quick accessible. The user has also the possibility to use a keyboard shortcut (like key Windows in MS, Apple in MacOS).

4.2.2.1.2 Information that the software must supply

- *How does the user know that his action has been understood by the software?* A list of all the installed applications is displayed. Maybe sorted by categories.

4.2.2.2 Select the application/file

4.2.2.2.1 Information that the user must supply

- *How does the user notify the software which applications/files he want to select?* The user selects an application or a file with a click on its icon. If he wants to select several files he needs to use the keyboard (Ctrl/Shift) or the mouse (drawing of an rectangle).

4.2.2.2.2 Information that the software must supply

- *How does the user know wich application is associated with a file?* A small icon of the application associated can be displayed over the icon of the file.

- *What happened if several applications are available for the same file?*
A list of applications associated can be displayed if the user right clicks on the icon of the file. A default application exists and it's this one that will be launched if the user does not specify any particular application to be launched.
- *How does the user know if he has the grants to read/execute the file/application ?* If the user can not read a file, its icon is in a grey tint, so the user knows that he will not be able to do any actions on it.

4.2.3 Removing of an application

An application can be directly removed by selecting the application to remove in the application folder and drag and drop it to the trash.

1. Go into the installed application folder;
2. Select an application to remove.
3. Drop it to the bin.

4.2.3.1 Go into the installed application folder

4.2.3.1.1 Information that the user must supply

- *How does the user notify the software that he wants to open the folder?*
An icon is displayed in the tool bar for a quick access but the user can also use a keyboard shortcut.

4.2.3.1.2 Information that the software must supply

- *How does the user know that his action has been understood by the software?* A list of all the installed applications is displayed. Maybe sorted by categories.

4.2.3.2 Select an application to remove

4.2.3.2.1 Information that the user must supply

- *How does the user notify the software which applications/files he want to select?* The user selects an application or a file with a click on its icon. If he wants to select several files he needs to use the keyboard (Ctrl/Shift) or the mouse (drawing of a rectangle).

4.2.3.2.2 Information that the software must supply

- *How does the user know the application which are installed?* All the applications installed in the user's machine appears in the installed application folder.
- *How does the user know if he has the grants to remove an application ?* If the user can not remove an application, all icons will be in a grey tint, so the user will know that actions are deactivated.

4.2.3.3 Drop it to the bin

4.2.3.3.1 Information that the user must supply

- *How does the user indicate to the software that he wants to put the file to the bin?* The user has to put the mouse over the icon of the trash.

4.2.3.3.2 Information that the software must supply

- *How does the user actually know when he can release his mouse for his action to be taking into consideration?* An animation on the icon of the trash occurs.
- *How does the user know that the application has actually been removed?* The icon of the application will disappears from the installed application folder and appears into the trash.

Chapter 5

User switching

5.1 Activity description and tasks identification

The operating system and also the desktop environment should provide support for three ways of switching among users. First, the user who is currently logged in logs out and then the other user logs in. In this case the session of the first user will be closed. The second way is to keep the session of the user who was logged in previously. In this case the session of the logged-in user will be saved. The information for the session state will contain the list of applications that were running in the time of logging out and the opened documents. The third possibility would work only for switching among users on one computer and would be quite similar as the fast switching feature which is implemented in Windows XP system. In this case logging out does not mean closing the running applications because they stay running in the memory.

1. Logging out;
2. logging in.

For the diagram, cf. figure A.2 on page I.

5.2 Tasks analysis

5.2.1 Logging out

There are three ways to log out, see the section 5.1.

1. Invoking logout command.

5.2.1.1 Invoking logout command

5.2.1.1.1 Information that the user must supply

- *How does the system determine the way the user wants to log out?* The desktop environment provides a menu from which the user can select the desired way (one of the three possible, see above) to log out.

5.2.1.1.2 Information that the software must supply

- *If the user chooses the first way of logging out, what happens to the open documents?* The desktop environment tries to close the applications and if there are unsaved documents the user will be asked by the applications to save them.
- *If the user chooses the second way of logging out, what happens to the open documents?* The session information will be saved (see above) and then the same procedure as in previous point (way 1) will take place.
- *If the user chooses the third way of logging out, what happens to the open documents?* All the running applications and all the documents will stay in the memory. This is a way of fast switching between users.

5.2.2 Logging in

1. Logging in.

5.2.2.1 Logging in

5.2.2.1.1 Information that the user must supply

- *How does the system determine which user wants to log in?* The user types in a username or chooses the username from a menu (this depends on the system configuration).
- *How does the system ensure that the user exists?* The user must be registered previously on the system.
- *How does the system authorize the user?* The user must supply his/her username and password.

5.2.2.1.2 Information that the software must supply

- *How can the user log in?* The system provides a log in screen where the user can either write his/her username or select it from a list. The password must be then supplied.

Appendix A

Diagrams



Figure A.1: Virtual desktops

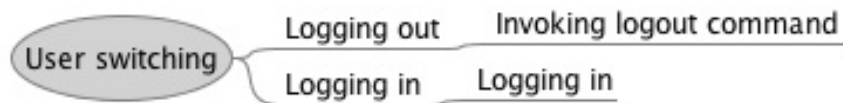


Figure A.2: User Switching

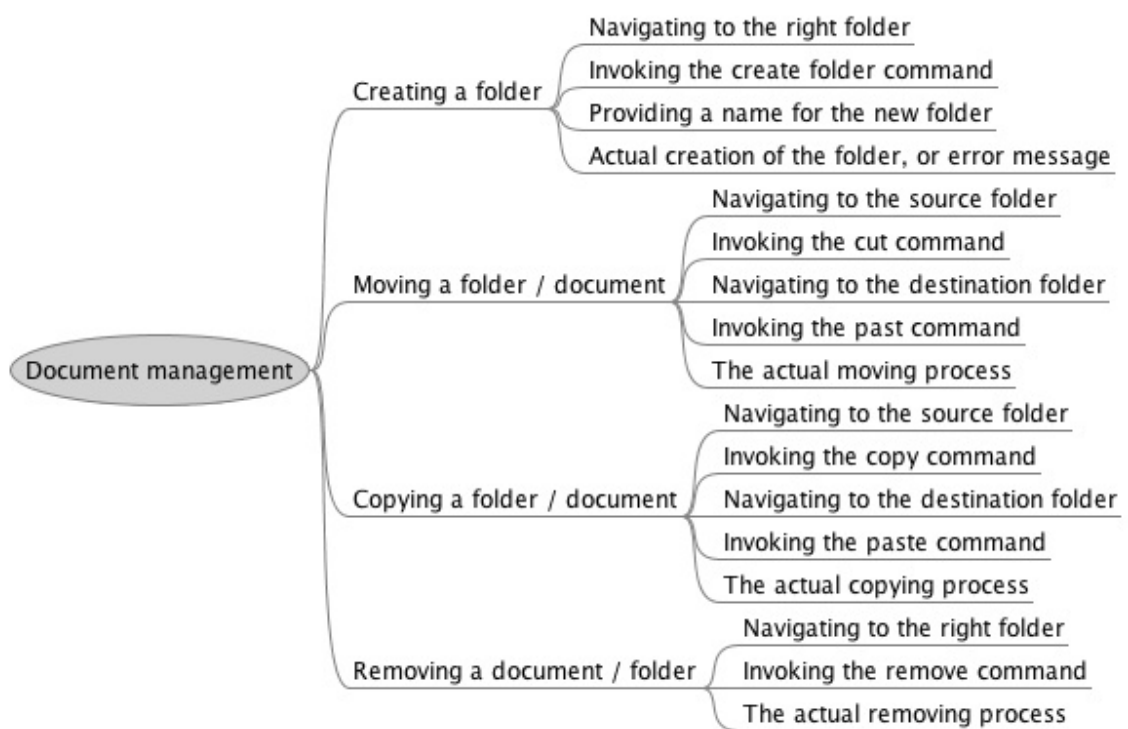


Figure A.3: Documentation management

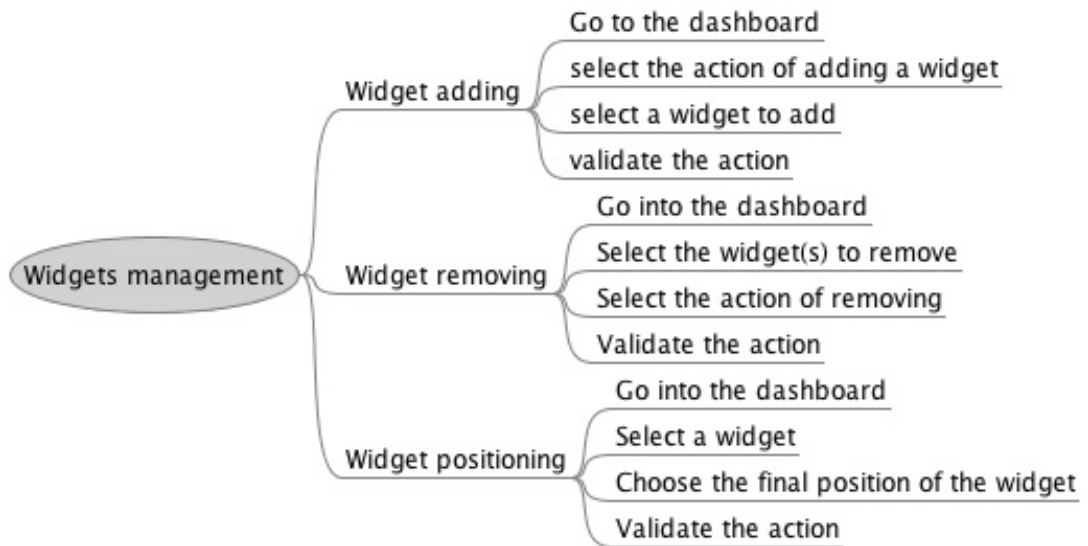


Figure A.4: Widgets

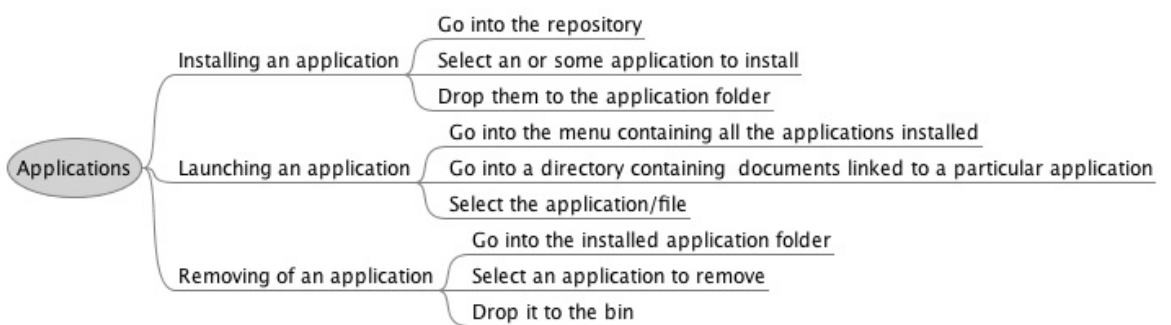


Figure A.5: Applications