

Functional/Implementation Specification for Aporia

Jeffrey Greenberg

3/17/89

*ADD:
change font size
resize icons*

TODO

A scheme for selecting several icons as arguments to another icon. How to lasso them, and drag them.
Drag one, then hold down shift key and drag over another, put it diagonally underneath, (a drag/arg list), then finally when click up, pass list to selected icon.

DESK

There are many desks, one for each job.
Each desk's tools are represented by ICONS.
Each desk is an ICON, in particular, the DESK_ICON.
There can be no less than one DESK_ICON present at a time since it is needed to switch to other desks or to end aporia.
There is a FIRST_DESK which has all the BUILTIN_ICONS. From these new desks can be created, old desks can be removed.
Each desk has the same layout on the Windows screen: starting from the top upper right going down, left adjusted, and as each column fills the next on the left is filled. Once the user moves the icons around, there characteristics are saved in the desk.

*- Batch files
: startup BATCH*

ICONS

Icons represent some action. To do the action you must RUN_THE_ICON.
Icons can either run on their own, or require another icon as an argument.
Icons are associated with a desk, and only one desk at a time.
Each icon has a list of DATA_FIELDS and a corresponding DATA_HANDLER.
The current list of pairs is:

path + arg list

PURPOSE	DATA_HANDLER	(ExecIcon ip)
DISPLAY	DATA_HANDLER	(DisplayIcon ip) <i>on</i>
CWD	DATA_HANDLER	(null)
COPY	DATA_HANDLER	(null)
HELPERFILE	DATA_HANDLER	(ExecIcon ip)
NOTESFILE	DATA_HANDLER	(ExecIcon ip)
PERMISSION	DATA_HANDLER	(PermissionIcon ip)
PATH	DATA_HANDLER	(null)

ADD OR COPY FLAG

There are several BUILTIN_ICONS that deal with these:

- RUN_ICON
- DISP_ICON
- DIR_ICON
- COPY_ICON
- TRASH_ICON
- HELP_ICON
- NOTE_ICON
- PERM_ICON
- PATH_ICON

options icon

There is one other kind of icon, a generic icon representing a file.

FILE_ICON

Other builtins that might be provided for common functions:

- diff
- cmp
- date (also touches files dragged onto it)

DATA_HANDLER

Each data handler takes an ICON as an argument, and extracts from that ICON the actual arguments the icon is interested in.

They also get a second argument, the POINT on which the icon was

clicked. This is so that the icon can figure out where it was clicked... it may be that different parts of the icon mean different things.

RUN THE ICON

There are two ways of doing this, DRAGGING and DOUBLE_CLICKING_LEFT.

DOUBLE_CLICKING_LEFT

This will execute the icon with a null argument if you have click permission.

DRAGGING

This will execute the icon that is just below the upper left corner of the dragged icon. The dragged icon is an argument to exec'd icon. The icon is moved back to its original position when an exec is attempted. If the passed icon doesn't not have drag permission, the exec is not attempted and icon is moved back.

If there is no icon below, the icon is simply moved to the new position.

PURPOSE

This is basic reason for having the ICON. It is the action the ICON represents.

A purpose boils down to either another program, hence an exec string, or an internal function, hence a function address. If these are blank, it does nothing.

DISPLAY

A display is the general appearance of an individual icon.

An entire display can be _VISIBLE or _INVISIBLE.

A display can be composed of a NAME, a PICTURE, a CD_TEXT, and FILE_INFO.

The name is on top of the bitmap, the cd text on the bottom.

The FileInfo is appended after the name.

The default is that all but the fileinfo are displayed, and that each, but not all, may be turned off. (Something must be displayable).

NAME

This is a word or two that describes the icon.

PICTURE

This is a bitmap that illustrates the icon.

CD_TEXT

This is the current position in the directory tree.

FILE_INFO

This is a long string

CWD

This is where the icon is in the directory tree currently. The "Current Working Directory" (CWD).

HELPPFILE

This is a file that generally cannot be edited which contains information about the program the icon represents.

NOTEFILE

This is a file that can be edited which contains anything the user wants to put in it, such as shortcuts about the program the icon represents, or a todo list for the program.

PERMISSION

These are execute permissions for each of the items in the list.

Each function either can or cannot be run by the user.

There are CLICK_PERMISSIONS and DRAG_PERMISSIONS. Click e

There is also a password here. A user can change an icon only if it knows the password.

FIRST_DESK

This desk has the basic functions for the entire program.

BUILTIN_ICONS

These icons perform the basic operations of maintaing a desk and icons icons.

They are a member of the first desk.

Only the directory icon can be duplicated.

RUN_ICON

This icon runs the PURPOSE field if you have permission.

The default is to ~~run a priori~~ again. *nothing*

Cannot be duplicated.

provided for consistency since all icons can be run by DBC-clicking

DISP_ICON

This icon edits the DISPLAY if you have permission. Can edit itself.

Cannot be duplicated.

DIR_ICON

The idea of the icon is quick movement through the directory trees.

To accomplish rapid directory changing, the first thing is to avoid dialog boxes. The main activities one does is to go up one in the directory tree (cd ..), jump to a given directory, to go back and forth between two or three directories comparing them, and to view what is in a directory. These functions are encoded in a special way into the icon surface itself so that with a few clicks one is where one wants to be: the icon is divided into 2 regions activated by double-clicking:

- up - up one in the directory tree (at top it beeps)
- view - (dlg) go to where you specify.

There is no directory stack since one can COPY_ICON this directory, and leave it to jump to it.

When 'view' is selected a dialog box with two windows is displayed:

- a directory window (or tree),
 - double clicking on a directory 'cds' there.
- a files window listing the files in the current directory.
 - selecting multiple files or double clicking on a file iconizes them: a FILE_ICON is created for each selected file.

Dragging a FILE_ICON on top of a directory, copies that file into that directory. If that name already exists, a RENAME_DLG is run. *move*

COPY_ICON

This is used to duplicate another icon. Does nothing without an argument.

Especially useful for duplicating the directory icon. Since when duplicated, it marks the current directory. Thus one can jump right to it by clicking on it. *move*

Copying a FILE_ICON, causes a RENAME_DLG to run since the file exists.

Does nothing if run without an argument. (Cannot copy itself).

Copying a DESK_ICON copies everything in the desk.

move RENAME_DLG

A dialog box for renaming a FILE_ICON purpose path, renaming the

*sets window icons
Non default icons
window size, pos.
Disappears after running
command*

Options

file in the filesystem.

FILE_ICON

This represents any file: executable, data, text, batch, etc. The user makes these.

The files path name is in the PURPOSE, the DISPLAY is an image of a folder and the display name is 'file', the CWD is the cwd.

TRASH_ICON

This icon removes icons dragged on top of it. Its default is nothing.

Removing a FILE_ICON, causes its removal from the filesystem.

Does nothing if run without an argument.

No CWD.

DESK_ICON

Running a desk without an argument, gives a dialog box for selecting other desks. You can either jump to another desk, clearing the others first, or add another desk so that tools can shared or recombined.

Dragging something onto a desk saves it as part of the desk. Can be deleted if not last visible desk.

HELP_ICON

The icons HELPFILE is displayed with the icons specified display program. If there is no icon (if clicked to exec), then the default help for aporia is displayed via the default icon display program, 'browser'.

NOTE_ICON

The icons NOTEFILE is displayed with the icons specified edit program. If there is no icon (if clicked to exec), then the default notes for aporia is displayed via the default icon display program, 'notepad'.

PERM_ICON

This icon changes the permissions on the icon.

To change permission one must know the password.

Permissions are only used if there is a password and the function is not permitted. The built-ins have a password and certain things are not permitted.

If there is no password then anything can be done without a query.

If there is a password and there is no permission for doing this function, it can be done if you know the password.

PATH_ICON

This icon changes the exec path for the icon. In this way, icons can have there own unique paths for different sets of programs.

DATA_HANDLER

This is an internal function which is passed an icon. It extracts from that icon that data it wants. It also supplies the defaults.

The handler can be null.

OPTIONS