A comparison of the two most popular word processors from the point of view of their conviviality

Fernando Sáez-Vacas & Carmen de la Torre
Departamento de Ingeniería de Sistemas Telemáticos
E.T.S.I. Telecomunicación
Universidad Politécnica de Madrid
Ciudad Universitaria, 28040, MADRID (Spain)
E-mail: fsaez@dit.upm.es
Fax: + 34 1 2432077

1. Introduction

In another paper submitted to this Conference, one of the authors proposes a theoric framework of technological innovation in companies in the field of the Office Automation (Sáez-Vacas,1991). The strategies defined there focus on the definition and setting up of a group of complexity fittings, according to Cybernetics'Law of Requisite Variety.

But in fact one of the biggest obstacles in applying the above mentioned strategies is the complexity of the office technology, even in the most elemental levels considered, such is the Tool Box.

In this paper, we aim to prove, firstly, that the argument of the excess of complexity is not a whim. We will focous our attention on a particular and widespread case within the Tool Box, word processors, and on the most widely sold products inside this category respectively, the one a few years ago, the other at the present moment: WordStar y WordPerfect. The aspect of their complexity we are interested in is their user interface, because in the first place it is the aspect that most influences the human job.

We will analyse both products in their releases 4.0 and 5.0. The fact that they are not the latest releases and that neither is completely contemporary does not seem to us to be important, because both products have presented with time design features well differentiated and, in essence, little changed for posterior modifications. The fact of even having chosen as guinea pig this kind of tool, instead of any other, is only relevant because, being better understood, becomes easier to communicate the general meaning of the conclusions.

Apart from this, and in coherence with the active principles of the theory mentioned above, we think that technical analysis that we will immediately summarize, have to constitute a previous and unavoidable step in the elaboration of complexity fitting mechanisms, as can be, for example, the design of better teaching methods and of personnel selection for the different jobs related to the office systems.

2. Some interface conviviality criterions

According to Hammer (1983), four are the criterions that a good interface has to meet. It has to be: a) Natural; b) Easy to learn; c) Easy to use; d) Consistent.

In this paper we will consider that these four features define an important part of what we, to quote Illich (1973), call 'conviviality'.

2.1 It must be Natural

It can be said that an interface is natural when it makes us 'feel at home'.

On introducing a word processor into an office, the employees that are obliged to use it will have to modify their way of working and to adapt themselves to the new tool requirements.

These changes could range from the using of the traditional typewriter and file to a word processor or from one word processor to another. The latter is probably less traumatic, as will be analysed at the end of this paper, but in the former change a series of adapting processes and rebounds are generated that will be greatly disminished if the new tool looks as good as posible and does not imply big changes in the old way of working.

It is a fact, nevertheless, that the environment on introducing any one of the word processors previously mentioned into a traditional working atmosphere has little or nothing to do with the environment the employee is used to: a new machine whose basic principles are unknown, a new semantic and a new syntax will have to sustitute the old one and the feeling of being in control of the working environment will be replaced by a feeling of insecurity on facing an unknown and complex mechanism.

Only conviviality in the interface design can supplement this lack of 'naturality' and facilitate this user-machine adapt process.

2.2 It must be Easy to Learn

Every learning process requires a cognitive processes sequence whose nature Psycology is not in agrement about.

But what seems clear for those studious of the relationship of first timers with the word processor is that a good learning of this kind of tool is hard.

One of the more practised researchers into this kind of problems is J.M. Carroll. Carroll (1982) enumerates the typical problems found by novel users on starting work on a word processor, problems applicable to all systems: disorientation, illusiveness, emptiness, mystery messages, slipperiness, side effects, paradox and laissez-faire.

Carroll, (Mack,1983), also describes the principal cognitive boundaries of a novel word processor user: learners lack basic knowledge, learners make Ad Hoc interpretations, learners generalize from what they know, learners have trouble following directions, problems interact, interface features may not be obvious and help facilities do not always help.

Experiments with novel users show that they are naturally inclined to do self-learning or learning by exploration. It is the 'active user' (Carroll,1985). Following his opinions, we will confirm how the interfaces of commercial products sold all over the world do not facilitate this kind of learning.

Among Carroll's proposals to remedy this situation, we find his 'Training Wheels Word Processor' that basically is a variety reducer (see companion paper for this conference) because it only allows to access to the basic operation set[1] and leaves those operations who more frequently lead to error situations inaccesible (i.e. operations which refer to a disk name).

For Hammer (1983) the fact that a tool is easy to learn lies in that its interface should supply enough help to beginners in his or her learning task. For this it should have at least mechanisms such as help commands, feedback mechanisms which provide the user with the state of current execution of his task or disaster prevention mechanisms (loss of data).

We will proceed to analyse immediately the facilities and difficulties that studied word processors offer to an inexpert 'active' user.

[1] This basic operations subset is being studied now by the authors of this paper with an object to designing better teaching methods.

First contact with the word processor

First of all, it is desirable to analyse the 'making contact', the very first feeling of a user who plunges into one word processor or the other.

The first thing the WP 5.0 user will find is an empty screen with only a hieroglyph message on the bottom right-hand corner with shortened information about cursor position (Document is worded Doc.,Page with Pg., Line with Ln, and the place within the line Pos.).

No introducing message to 'break the ice', no indication about what to do immediately, neither a simple indication on how call for help. If guided by his experience or by what he knows, as Carroll states, he types F1 remembering that most programs he knows select help like that, he will find an option line corresponding to the 'Restore erased text' operation, so his inmediate problem will aggravated.

In WS 4.0, the first screen the user sees is the so called 'Menu without a file' what could be considered as the WS 4.0 hall because those operations related to files, WS 4.0 utilities, system commands and the so called 'Opening files commands' are selected from it.

The aforecited menu consists of an avalaible operation list together the letter that has to be typed to select it.

Like WP 5.0, not only is there no message about how to succed in reaching his overall goal, i.e. to write a letter ... there is none his first partial goal neither: to reach the Edition Screen.

The first problem that faces the novel WS 4.0 user is not the WP 5.0 emptiness, but a semantic problem, because he will have to distinguish between a 'file document' and a 'no file document'.

The problem, then, for the 'active' user will therefore be one of 'disorientation' defined by Carroll as: "The user doesn't know what to do in the system environment" (Carroll,1982,p.50).

Help mechanisms

Carroll and his colleages, (Mack,1983), manifest the paradox that the help facilities 'do not always help'. The main problem is that users do not always know what to ask for exactly and that help information is not always focused on the learner's specific difficulty.

The only help facilities which have been useful in Carroll's experiments are those which have some sensitivity to the problem context from which they are called (thus relieving users, at least in principle, of having to describe what help they require).

In this kind of help facilities the users can position the cursor in a field of a menu about which they want information and request help. They can also request help when an error message appears on the screen which they don't understand.

Neither WP 5.0 nor WS 4.0 [2] have such facility help.

WP 5.0 help not only is not context sensible, but can only be requested from the Edition Screen, i.e. no help can be requested from any menu, option line or specific edit screen not even when there is a message on the screen.

On selecting help, a screen is presented to the user in which there are explanations on how to manage it; the user can type one letter to obtain a list with all the operations which begin with this letter in alphabetical order or he can type whatever function key to obtain the description of the operation associated with this key. Some keys show the information organized on identical menus of those which appear when the operation is selected from the Edition Screen.

He can also have access to a template which includes the operations associated with the function keys and their combinations with the Ctrl, Shift and Alt keys.

The offered help is then purely syntactyc.

The WS 4.0 help is either context sensible and like WP 5.0 can only be reached from the Main Menu which is the state equivalent to the WP 5.0 Edition Screen. Neither can it be called from another menu nor when there is a message on the screen.

The help facility is selected from the Main Menu and gives place to the Help Menu which is organized in operation groups.

The information offered is syntactic too, as it show how to do certain operations (i.e. to place documents's margins) and semantic, as it shows what some indications means (i.e. what the status line indications or the rightest column marks mean).

[2] WS release 5.0 has a windowed sensible context help but not for error messages.

Feedback mechanisms

When we say 'feedback mechanisms' we refer to all of those mechanisms which provide the user with all current information regarding the state of his work.

Both WP 5.0 and WS 4.0 have a 'status line' where there is information about the cursor position within the document, including the page number, line number, column number, etc., the name of the file currently being edited, if it is in insert mode or not, and the tabulator and margin rule.

In some operations, like file storing, the user is shown a message indicating that his document is being stored.

2.3 It must be Easy to Use

According to F. Sáez-Vacas (1990), the user in his learning task goes through a process which can be operatively simplified in three stages: learning, consolidation and mastering.

The learning process has alredy been detailed in the section before, so now we will focus on the consolidation and mastering of the word processor.

The Consolidation stage

According also to F. Sáez-Vacas (1990,p.253) a consolidated management tool is reached by cognitive procedures based on comprehension and repetition.

In the process of consolidation of the learned operations affect the reference facilities offered by the word processor a remarkable way, whether is be through the help facility or through the manuals.

The reference manuals or the books taken from them, offer, as stated Carroll (1985), a lot of information which mask the really important thinks: explanations, side-tracks and drills hide the basic kernel information.

Attemping to use them as a reference during the practical stage, generated in the users of Carroll's experiment great doses of frustration that made the process more complicated and discouraging.

Carroll, with his Minimal Manual, also used in the learning stage, showed that this one was more suitable as a reference for encouraging users than the extensive ones.

Leaving aside, therefore, the reference manuals and supposing that the user has understand how the operations works, the reference needed by the user will be in most cases syntactic, until by using and repeating processes he or she memorizes syntax of the operations that he usually uses: the rest of the operative options, which constitute a bigger set, form part of what is called 'technologic waste'.

WP 5.0 offers, as was seen in section 2.2, a syntax help facility which provides a list of operations in alphabetical order with its syntax. On pressing a key a list with the syntax of all those operations begining with this letter is offered.

The main inconvenience, initialy unavoidable, to be used as a reference, is that the user has to remember the exact operation name to be able to find it in the screens (i.e. if he has forgotten how to save a document and he looks for help for 'File' he doesn't find anything; only if he looks for 'Save' or 'Exit').

Another inconvenience, previously mentioned, is that the user loses sight of his document.

The WS 4.0 help facilities are more limited. Apart from the menu templates, which show which letter the user has to type to select an operation, there is the Help Menu where the user is indicated as to which menu a given operation set is to be found and the syntax of a special group of commands (those so called 'Point commands') by means of a continuous serie of windows, not selectable by a letter, but by displaying one by one until he finds the desire operation.

The problem of using the word processor after the learning phase becomes, according to F. Sáez-Vacas (1990,p.189), in a problem of usage, supported or not, as we have just seen, by the reference facilities which the word processor offers.

One usage facility offered by WS 4.0 in this consolidation process is the posibility of introducing several kinds of 'user profiles' which turn up in the display or non-display of the menus and explanations.

In the lowest help level, the interface becomes purely 'descriptive', because as the menus disappear the user has the feeling of working directly with commands.

Mastering Stage

This stage, which is reached after working many hours with the word processor and the user's interest implies his mastering it, with a full knowledge of all its operations and its syntax as well as in occasions its design failures and tricks or self prescriptions in order to solve certain situations which are learnt by experience and are not revealed in the reference manuals.

The group of users who reach this last stage is almost insignificant and is not the object of our interest, not being representative, in this paper.

2.4 It must be Consistent

An interface can be considered 'consistent' when it keeps a uniformity as to style and minimizes as far as possible the existence of 'modes' which frequently give place to the aparition of 'side effects'.

The uniformity of style in the interface has a lot to do with the kind of user interaction with the interface, in the same way that the interchange with an affable and cordial person is not the same as with an introverted and complexed one whose reactions can never been foreseen.

A 'mode' is the context inside which the system lies. An action having certain meaning in one mode can have another completely different one in another mode, leading the user, as a consequence, to a state of confusion. Following with the same simile, it would be equivalent to the state of mind of the person we are relating to. Depending on if he or she is in a good or a bad mood they can react to something we say in a completely different way.

A 'side effect' is a consequence of the cause of an action whose effect went unnoticed when it was done and is seen afterwards. Referring the person of the simile, we can do or say something that upsets him without noticing, and after a time we realise that he is stand-offish with us.

The uniformity of the interface

The interact way, that is, the way in which the user relates to the word processor is an important factor in the user's approach to the tool with which he has to work.

According to Hammer's classification (1983) the way of interaction with an interface can be 'descriptive', when the user introduce a command which is interpreted by the interface or 'selective' when the user chooses one of the several options showed by the interface.

We are going to see immediately how the WP 5.0 and WS 4.0 are a mixture of both ways of interaction depending on the context.

In WP 5.0 from the Edition Screen all the operations are selected by typing a command; it is therefore 'descripitive'.

The line with the avalaible options which turns out a menu typical in the purely selective interfaces does not exist on the screen.

Nevertheless, when typing a command, the interface interacts with the user in several ways: a first group of commands makes the interface selective, but not in a homogeneous way, because some commands turn out a menu which occupy all the screen, for which the user loses sight of his document and others turn out an option line beneath the text.

A second group of commands turns out a dialogue with the user, requiring information from him.

A third group of commands turn out a change of context and there is even a fourth group of commands which just turns out the activation of the associated operation.

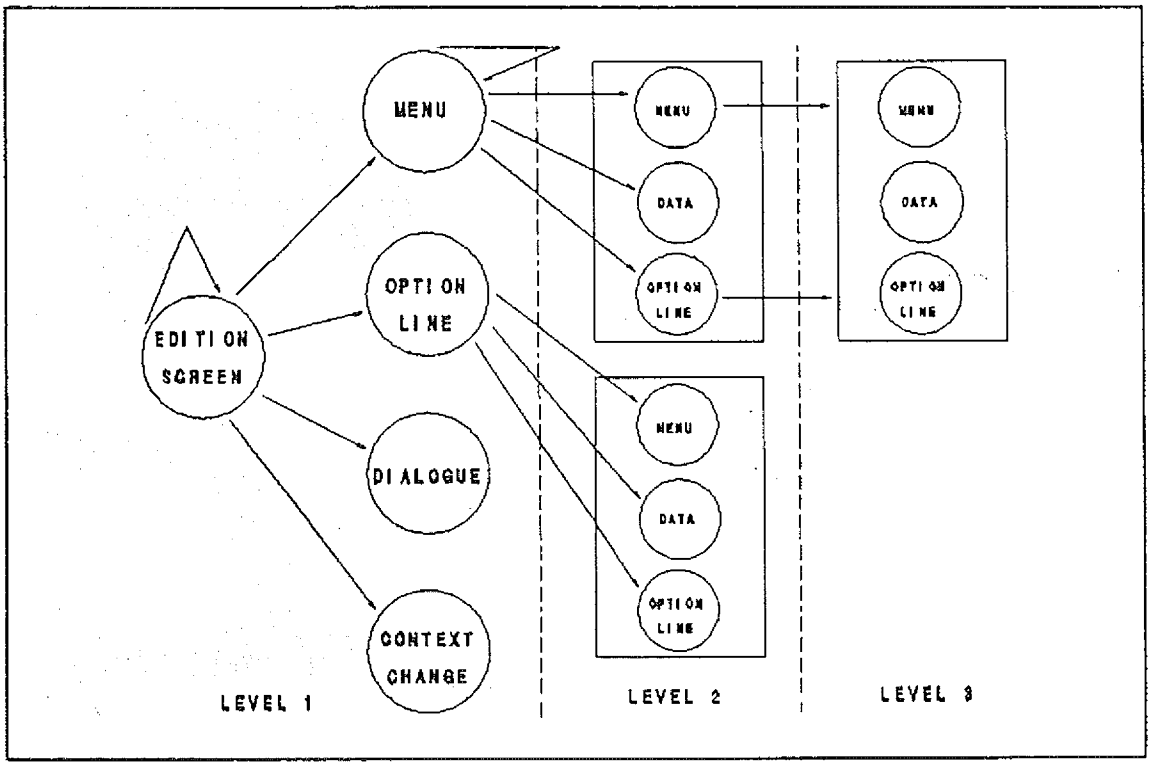

ILLUSTRATION 1: Diagram of stages associated with the interaction with WP 5.0

Nevertheless, the non uniformity of the interaction mode doesn't end in this first level which consists of the user's interaction from the Edition Screen.

Neither does uniformity exists in the second level of interaction, when the user relates to the menus and the option lines.

From one menu there are options which turn out: another menu (submenu) which replaces the previous menu on the screen, the positioning of the cursor on the selected option requiring him to change the data behind it, a line of options beneath the menu or just the operation selection.

The same variants also exist from the submenus too.

On selecting an option from the option line, the user could get: a menu, another option line, the requesting of a data or simply the activation of the operation associated with the selected option.

Other option menus or option lines turn out a specific screen of edition such are headings and foot notes, graphics editing screens, commentaries, tabulators rule, etc.

The diagram of stages the user can go through is better shown in Ilustration 1.

There are even more levels that a user can go deeper into on trying to select an operation and which are not shown on Illustration 1, not being in most of the operations. For example, to specify the paper the user is going to use, the user, after arriving at the second level through two menus, has to go through a third menu to specify the paper size and through a fourth one to specify the quality. That is if he or she uses a predefined paper type, because otherwise, he or she will have to introduce the size of the paper onto a third menu and from the fourth menu he will even go through a fifth one to select the quality before finishing the specification of his paper.

OPERATION	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
Margins	menu	menu	data		
Headings	menu	menu	op.line	op.line	screen
Bold	op.line				
Move block	op.line	op.line			
See codes	change context				

TABLE 1: Some WP 5.0 operations

Another example of a never-ending tree of menus, option lines, data, explanatory and specific screens is the 'Printer Select' operation, the description of which goes beyond the limits of this paper.

Some examples of operations and the sequence of stages the user has to go through are those on Table 1.

The user's mode of interaction with WS 4.0 is, as we will now see more uniform.

The interface can be considered of the 'descriptive' type because all the operations are selected by introducing a combination of keys.
On typing a command, the user finds: the associated operation activation/desactivation (they are the so-called 'Switchings', the operation associated with the command or that a dialogue with the user requiring additional information is iniciated.

If WS 4.0 is configurated with the highest help level, the user will have the feeling of working with a 'selective' interface, because at the top of the screen a template showing the avalaible commands at each moment appears.

The diagram of stages associated with the interaction with WS 4.0 is showed in Illustration 2.

Nevertheless, this larger uniformity of the interface has a price: the existence of 'modes'.

The existence of modes and side effects

A 'mode' is the context in which the system is found at a given moment. An action which has a certain meaning in one mode could have a completely different meaning in another one leading the user to a state of confusion.

In WS 4.0, above all if one is working with the help level in which the templates are shown, it is important that the user knows at each moment where he is.

The WS 4.0 user interface is organized in operation groups which are selected from several menus which are reached by typing a command. Once inside the menu the required operation is selected by typing a letter key. If one is working without the templates, as is the case of a master user (who we are not concerned with for the moment), the modes are masked because the user will type all the keys consecutively, but if he works with the templates, from the Main Menu he will be able to reach another menu on typing the corresponding combination of keys. The new menu will replace the old one on the screen and the user will have changed mode, because though his screen would look the same, a menu with the text document underneath he will not be allowed to complete the same actions as from the Main Menu. Particulary, if he tries to type text from a different menu than that the main one, the first character he types will be

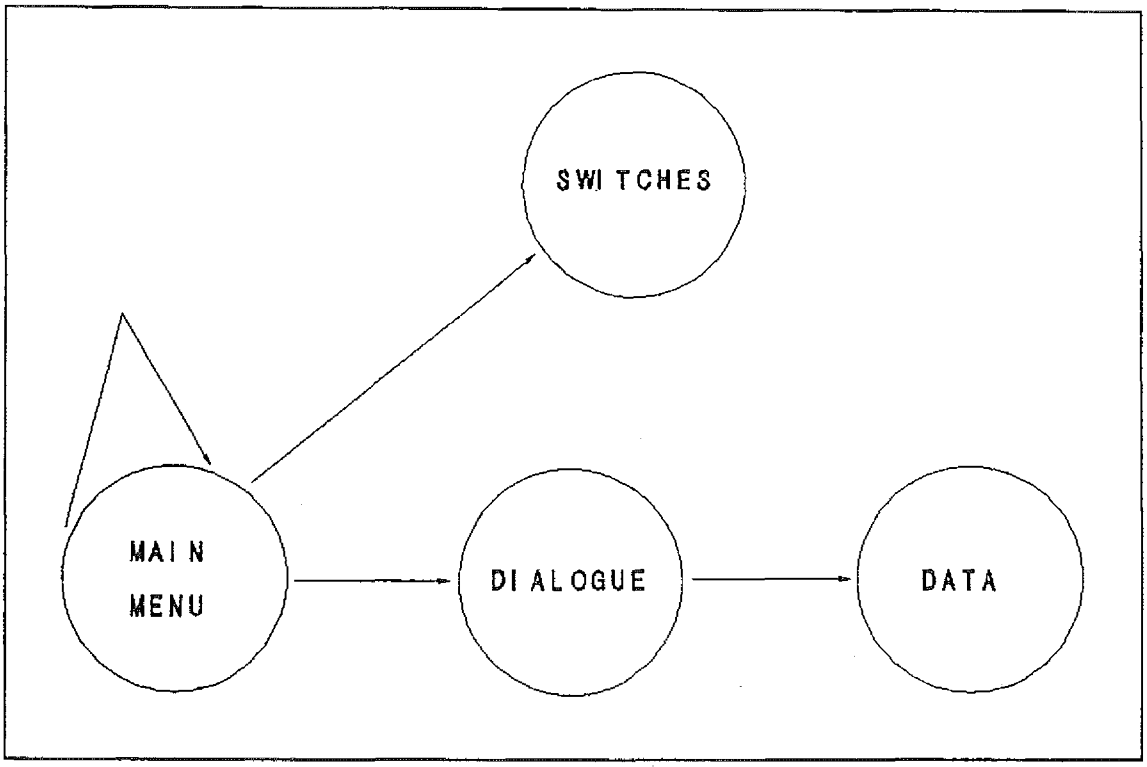

ILLUSTRATION 2: Diagram of stages associated with WS 4.0

interpreted by the interface as a command if it coincides with the letter associated with some command menu, if not, it will turn out on the Main Menu.

This problem can occur frequently, above all with beginners, because, as Carroll states, the user frequently trying to attain an operation (i.e. center a line), can lose sight of his final goal while trying to reach partial goals (looking for the menu where the command is, finding the letter associated) and he may decide to stop at any given moment of the process, getting caught up on a mode, that looking the same as the initial situation from where the user started, can give him the impression of having reached it.

This 'going back', in WP 5.0, would oblige the user to undo his good work and climb all the levels he has gone through, however once his text is displayed on the screen he will have reach the situation from which he started. Another different problem is that of the 'side efects' that imply this going back, frequently uncontrolled.

The existence of modes in WP 5.0 are revealed in special operations such as 'Codes window' and the 'Delete' operation.

Without the Codes Window on the screen the user can delete any character on his text. If underneath the cursor there is a code and 'Delete' is selected, WP 5.0 will ask the user if he really wants to delete this code, but if he has on the screen the Codes Window it will not ask him anything and it will just delete it. This is, therefore, a mode, because the same action (deleting a character) has different consequences depending on the stage in which the system is at. In this case it is disastrous for the user, as he can delete a code without noticing, while trying to delete a character of his text. This also implies a side effect, because on the code's disappearance the word processor will perform in a different way than the user expected.

We conclude from these examples that in most cases the existence of modes in the interface induces side effects when the user doesn't notice in which situation the system is and proceeds erroneously under the impression he is doing things right.

3. Disasters prevention mechanisms

According to our criterion the profile elaborated by Hammer does not include all the elements of interface conviviality. In particular he does not consider certain security mechanisms absolutely essential for the user.

By disaster prevention mechanisms we mean those facilities supplied by the interface to protect the user from losing his job, or part of it, due to his own mistakes or failures of the system. Nothing is more anticonvivial than wasting hours of working time in front of the computer.

Generally speaking these mechanisms can be classified in three groups:

a) Recuperation mechanisms
b) Mechanisms to face system failures
c) Confirmation mechanisms.

3.1 Recuperation mechanisms

WP 5.0 provides a function called 'Restore erased text' which stores up to three erased groups of characters.

WS 4.0 has not a similar mechanism , so whatever the user erases in his text is lost forever.

3.2 Mechanisms to face system failures

WP 5.0 supplies the possibility of automatically storing, every so often and at user's will, the document being edited, in a temporary file.

In this way, if a system failure does occur, the user loses only the thing he was typing up to the moment in which the system failed.

The security files (of which there are two, one for each document that can be edited in WP 5.0) remain in the disk til the user logs out of WP 5.0 correctly.

It also provides the possibility of keeping the previous version (with .BK! extension) each time the file is edited and saved.

WS 4.0 does not keep temporary security files, so the user has to concern himself periodically with saving his document in prevention of system failures, this not being very probable in the case of a novel user.

The .BK! file is not supplied as an option but it is created every time a new file is saved.

3.3 Confirmation mechanisms

When we say confirmation mechanisms we mean those which ask the user to confirm his action. Obviously not all commands need to be confirmed, only those that could cause loss of information.

Among these functions we can include those related to erasing files, deleting large sets of characters, blocks to be exact, and archiving and exit operations.

WP 5.0 considers this aspect carefully. As to the of erasing files from the directory, WP 5.0 always requires confirmation through a question which includes the name of the file which is to be erased with "Yes" or "No" as posible answers and with "No" as a default option. The only affirming key is the "Y" or "y" key, the rest of the keyboard automatically means "No", which is an important protection against mistakes or carelessnesses on touching the key.

WP 5.0 provides the possibility of selecting a set of files before performing an operation with them. If this operation is 'Erase' a double confirmation will be required with the same format mentioned above.

WS 4.0 does not require confirmation from the user in any of its erase functions or those that have to do with files, or those involving delete blocks so that a mistake in the management of these functions will end up losing the information definitively (it can be only recovered from the .BAK file if it is alredy in it).

An inexperienced user could easily walk away from the word processor when he has finished his job, forgetting to save his work or even being ignorant of the fact that he has to do so.

The difference between WP 5.0 and WS 4,0 in relation to the way of performing this operation is that in WP 5.0 the user gives a name to the file after editing it while WS 4.0

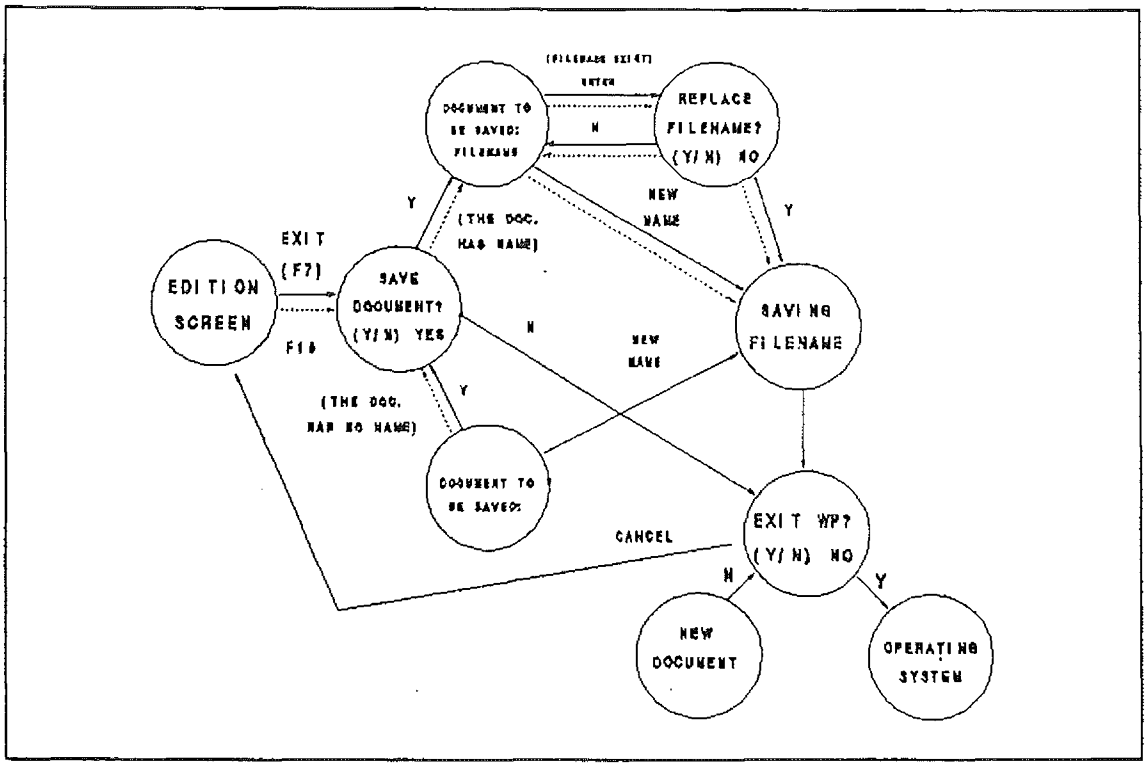

ILLUSTRATION 3: Diagram of stages of WP 5.0 Operation Save

requires the name of the file to get into the editing mode.

The only way to exit from WP 5.0 (except for switching off the computer), is by means of 'Exit function' (which syntax is F7).

The possible situations are showed in Illustrations 3 and 4.

As we can see WP 5.0 stresses that the user should confirm whether he wants to overwrite the file, but it does not control whether the user wants to exit the document in which changes have been made, except that the default option of the question asking for filing the document is 'Yes'.

But, in WS 4.0, the opposite occurs, emphasis being stressed on not exiting by mistake without saving the changes made in his document. In this case it asks the user for confirmation. On the other hand, it does not require confirmation when the user selects any of the three available functions for saving a document.

The WS 4.0 criterion seems more sensible since it is more critical that the user should not save the changes made in his document than that he should overwrite changes that

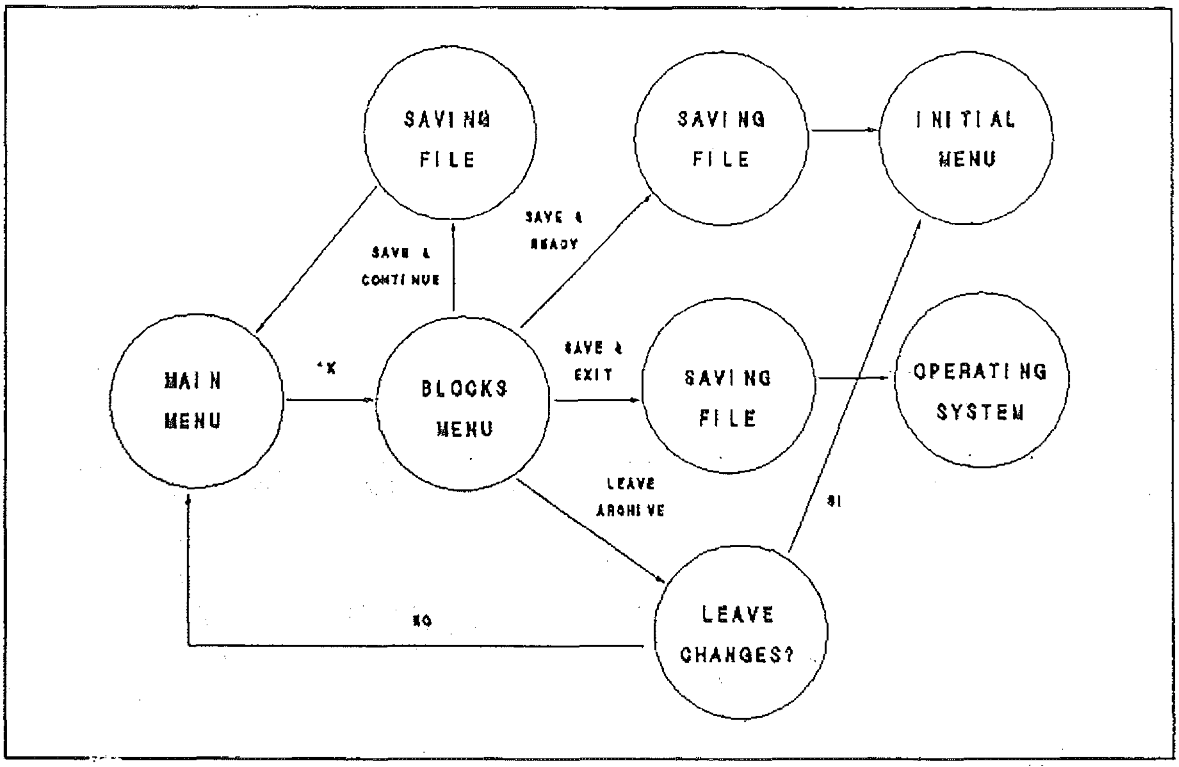

ILLUSTRATION 4: Diagram of stages of WS 4.0 Operation Save

he does not wish to keep on the file, as he can always recover these from the .BAK file.

WS 4.0 is inconvenient in that it provides four different functions for saving documents, this increases the variety perceived by the user who will have to discern which of the four is the best for him at each given moment. In WP 5.0, as shown in Illustration 3, these functions are substituted for only two functions which includes a dialogue with the user to inquire whether, after saving, he wishes to continue editing, he wishes his document to dissappear from the screen, or whether he wants to exit to the system.

4. Migration from one word processor to another

While speaking of the 'naturality' of an office tool as a word processor it was said that the transition from one word processor to another was less traumatic for the user than the transition from a traditional work implement, like the typewriter, to a word processor. There is yet a lot to be say for this 'less'!.

The user who is obliged to make a transition from one word processor to another, whether it is from WP 5.0 to WS 4.0 or viceversa or to whatever processor, has a semantic baggage that the user who changes his work environment for a word processor has not.

The word processor terminology and jargon is, more or less, common to all, and little or nothing can be done without some basic knowledge of them.

Notions like justifications, indentations, blocks, automatic returns, unbreakable spaces, automatic references, etc, do not belong to the common vocabulary of people who have not used a word processor.

But, the syntactic change that is required to pass from WP 5.0 to WS 4.0 (or other releases of WP and WS) or viceversa forces the user to get rid of his syntactic mental network and to replace by a new one, process which is as much costly as much consolidated is the old processor handling, to extend of interfering cognitively (Sáez-Vacas,1990,p.252).

WP 5.0 as we have already seen runs by means of selectionnable commands with the function keys, whether alone or whether combined with the Ctrl, Shift and Alt keys, which gives rise to the appearance of menus or option lines whose options are selected by the number or the key which it is associated with. WS 4.0 works by means of commands, supported or not by help menus, built by means of Ctrl key combination with one or two letters and through the so called 'point commands' which are typed into the text, preceded by a point, and whose syntax are very strict.

5.Conclusions

We have analysed the two most wide-spread word processors in the world market from the point of view of conviviality. Although the brevity of the paper has not let us give a completely detailed account of both products conviviality features, which do not necessarily mean the superiority of one over the other, we think that their inadequacy with regard to their human interface has been made perfectly clear.

Both tools are very powerful and give rise to an incredible amount of possibilities but their functional complexity is not accompanied by a directly proportional usage facility. We can compare this experience with those involving other technological products, proving, in our opinion an unsuspected and general lack of sensibility (which lately has begun to correct itself) of the technological industry regarding this fundamental factor for the user. In other words, the industry has preferred, as always, sophistication rather than conviviality.

To avoid these problems, which create the logical rejection of the user, the industry should design its products fitting them out with a much more natural, easy to learn and to use, and more consistent interface, at least for a basic kernel of functions, and without losing potencial in the rest of their possibilities, as these correspond to later and more advanced phases in the users training period, and consequently only concern a minority of them.

Latest releases of these word processors (i.e. WP 5.1 or WS 6.0) let the user choose a windowed menus and mouse oriented interface, but this does not resolve the problems of design analyzed in this paper.

We think we can do something to improve the training processes of the current tools without waiting for the moment the industry will develop its products in a more convivial form. For this reason the analysis presented in this paper supplies a lot of information on how to go about selecting a basic kernel of functions, how to create minimal manuals and how to design teaching methods which help to avoid the difficulties that tool designers incompetence in the area of conviviality have created.

References

Carroll,J.M., The adventure of getting to know a computer, **IEEE Computer**, 15 (11), 1982.

Carroll, J.M., Minimalist design for active users, **Human Computer Interaction -Interact'84**, North Holland, Amsterdam, 1985.

Hammer, M. et al., What makes a good user interface?, **AFIPS Office Automation Conference Digest**, 1983.

Illich, I., **Tools for Conviviality**, Harper & Row, N.Y., 1973.

Mack, R.L., C.H. Lewis, J.M. Carroll, Learning to use a word processor: Problems and prospects, **ACM Transactions on Office Information Systems**, 1 (3), 1983.

Sáez-Vacas, F., **Complex Office Automation**, Fundesco, Madrid, 1990 (in Spanish).

Sáez-Vacas, F., A cybernetic theory about computer interfaces and human factors within a framework of technological innovation, paper submitted to this **Conference**, Tampere, 1991.