Full Principles Set

Derived from the usability literature by Dr. Iain Connell (now of University College London Interaction Centre, or UCLIC), as part of a DPhil research programme at University of York. The contents may be cited as

CONNELL, I.W. (2000). *Full Principles Set*. Set of 30 usability evaluation principles compiled by the author from the HCI literature.
URL as source

This full set of 30 Usability Evaluation Principles (UEPs) is grouped into 7 larger sets, namely

Requirements and Functionality Principles
User - System Principles
User Principles
Comparative Principles
System Performance Principles
Perceptual and Motor Principles
User Support Principles

each containing between 1 and 10 principles. Some principles are further divided into attributes (currently up to 2), each of which can be considered as a sub-principle; where no sub-division has taken place, the attribute name is the same as the principle name.

Contents

Requirements and Functionality Principles

These principles concern the match between what the system does and what its intended users want it to do, and between who those users are and who they are perceived to be by the designers of the system.

Requirements and functional specifications would normally be performed as part of a requirements analysis process.

Requirements Match

Attribute	FUNCTIONAL NEEDS No : 1
Principle	Requirements Match
Set	Requirement and Functionality Principles
Explanation	The set of functions offered by the system (ie. what the system does) should cater for the needs and requirements of the users for whom it is intended.
Example(s)	The range of operations and options provided by a word processor will now need to be extensive, to meet the expectations of users who may have been exposed to other existing and prior versions. Even users new to such a system are likely to be aware of the range of functions which word processors now provide.
Exception(s)	However, even familiar and practised users of popular systems such as word processors will expect to be able to perform simple operations, and the functional set delivered by a new or revised system will have to include these.
Related to or affects	Requirements needs Functional provision Functional organisation
Comments	This is a simply stated principle, yet the matching of system functionality to user requirements is, along with the eliciting of those requirements, an essential pre-requisite to system development.
Source(s)	Williges & Williges 1984, Norman 1988, Denley et al 1993, Sutcliffe 1995, Dix et al 1998, Shneiderman 1998

Attribute	REQUIREMENTS NEEDS No : 2
Principle	Requirements Match
Set	Requirement and Functionality Principles
Explanation	The characteristics and functional requirements of the users for whom a system is intended should have been accurately determined.
Example(s)	The user base for familiar and popular systems such as word processors and spreadsheets is now very large and potentially very diverse. The requirements of such users will thus now be very wide-ranging, and may be reflected in the various additional features and functional groupings offered by such systems.
Exception(s)	However, specialised systems will have a more clearly defined user base, whose requirements will be easier to determine.
Related to or affects	Functional needs Functional provision Functional organisation
Comments	This is a simply stated principle, yet the eliciting of user requirements is, along with the matching of those requirements to functions offered, an essential pre-requisite to system development.
Source(s)	Williges & Williges 1984, Denley et al 1993, Sutcliffe 1995, ISO 9241-10 (1996), Shneiderman 1998

Functional Utility

Attribute	FUNCTIONAL ORGANISATION No : 3
Principle	Functional Utility
Set	Requirement and Functionality Principles
Explanation	The set of functions offered by a system should provide the best means of performing the required operations. The organisation of system functions should match with the expectations and knowledge of the intended users.
Example(s)	The set of menu options offered by a system can be organised in many different ways, but this organisation should be in line with any prior expectations of such systems and should, at the least, match with what is expected of the system by its users.
Exception(s)	It is possible to provide layers or sets of functionality at different levels, to cater for users with differing expectations and experience. Common examples include 'full' and 'short' menu sets, and the extension of existing functionality by additional options.
Related to or affects	Functional provision Functional needs Requirements needs
Comments	The organisation of system functions represents one of the most varied aspects of different system types, and there is little which can be laid down as a guide. However, some standardisation has now become a de facto expectation, such as File .. Edit .. groupings for pull-down menus. It is also possible to explore a selection of competing alternative functional organisations, in the form of paper-based scenarios, before committing resources to prototype versions.
Source(s)	Smith & Mosier 1986, Marshall et al 1987, Nielsen 1993, Sutcliffe 1995, Scapin & Bastien 1997, Jordan 1998, Dix et al 1998, Shneiderman 1998

Attribute	FUNCTIONAL PROVISION No : 4
Principle	Functional Utility
Set	Requirement and Functionality Principles
Explanation	The set of functions offered by a system should provide the best means of performing the required operations. There should be no redundancy or under-provision of system functions, there being exactly those required and no more.
Example(s)	The set of menu options offered by a system can be organised in many different ways, but should, in total, match with the functional requirements of its users.
Exception(s)	It is possible to provide layers or sets of functionality at different levels, to cater for users with differing expectations and experience. Common examples include 'full' and 'short' menu sets, and the extension of existing functionality by additional options.
Related to or affects	Functional organisation Functional needs Requirements needs
Comments	It has been shown that many users make use of only a small proportion of the extensive sets of functions which large systems now provide, a situation which can only partially be explained by the reluctance which even experienced users feel to explore the unfamiliar parts of a system. It is compounded by the resulting reduction in performance which many large and expanded systems now force on their previously adequate host machines.
Source(s)	Murphy & Mitchell 1986, Marshall et al 1987, Brown 1988, Norman 1988, Hix & Hartson 1993, Nielsen 1993, Shneiderman 1998

User - System Principles

These principles concern the ‘flow of interaction’ between user and system, that is, the sequences of choices and actions which the user makes in response to the system, and the types and nature of the messages, displays and other outputs which the system presents to the user.

The range of issues include the locational and navigational information which the system provides to the user, the type of feedback which is given in response to user commands, the way in which the system defines and handles errors, the range of choices available to the user at each stage in interaction, and the terminology and language used by the system for its text messages and displays.

Navigational Effort

Attribute	MINIMUM STEPS No : 5
Principle	Navigational effort
Set	User-system principles
Explanation	It should be as easy as possible for the user to move (in steps or stages) between system states and between functional components. There should be minimum number of steps between related components, and un-necessary repetition of step sequences should be avoided. In a multi-state (eg. multi-tasking) system, there should be a means of access to states which are hidden or occluded
Example(s)	A 'step' can be represented in a number of ways, depending on the interface type. In a WIMP interface, steps might be successive mouse clicks required to open successive dialogue boxes or windows. In a menu interface, each step might be successive sub-menus in a menu hierarchy.
Exception(s)	In a tutorial or 'training wheel' (rehearsal sequence, offered only on request), the number of steps might be deliberately extended, for clarity or explanation. Reduced-step sequences might also be offered for experienced users, such as keyboard or toolbar shortcuts to familiar selection sequences, or user-defined macros.
Related to or affects	Minimum retraction Step modification
Comments	One of the most common complaints from users concerns the amount of un-necessary work which repetitive or over-simplistic systems require of them in order to perform simple operations. The now common use of within-screen dialogue boxes and windows means that it is no longer necessary or expected that a new screen-full be presented for each and every selection or response in an interaction sequence; increasingly, combinations of selections are being offered within a single box or window, often with multiple 'sub-modes' (eg. the 'card-index' metaphor) from which sets of options are offered. Many experienced users will expect and require that they be permitted to define their own shortcuts to frequently used sequences, even to the extent of re-configuring the whole menu layout or toolbar collection.
Source(s)	Galitz 1985, Denley et al 1993, Sutcliffe 1995, ISO 9241-10 (1996), Scapin & Bastien 1997, Shneiderman 1998

Attribute	MINIMUM RETRACTION No : 6
Principle	Navigational effort
Set	User-system principles
Explanation	It should be as easy as possible for the user to move (in steps or stages) between system states and between functional components. There should no un-necessary retraction of steps already made.
Example(s)	A 'step' can be represented in a number of ways, depending on the interface type. In a WIMP interface, steps might be successive mouse clicks required to open successive dialogue boxes or windows. In a menu interface, each step might be successive sub-menus in a menu hierarchy. It should be possible to return to a previous step without having to retract all the intervening steps already made (though this should be still offered as the default), for example in an error situation or when the most recent step does not lead to the desired situation.
Exception(s)	In an error situation, where potentially serious consequences might ensue from continuance with the initiated sequence (eg. 'file not saved : do so before closing ?'), it might not be wise to offer a return to a previous step from which the safety option ('save file', here) is not available.
Related to or affects	Minimum retraction Step modification
Comments	One of the most common complaints from users concerns the amount of un-necessary work which repetitive or over-simplistic systems require of them in order to perform simple operations. In particular, in a hypertext or web-browser system, the navigational effort required to return to some known and previously found state should not be the same as that used to become 'lost' in the first place : one solution is the use of history lists or other shortcuts to backtracking.
Source(s)	Galitz 1985, Brown 1988, Hix & Hartson 1993,

Memory Load

Attribute	MEMORY LOAD No : 7
Principle	Memory load
Set	User-system principles
Explanation	Complex input formats should be avoided where possible. If they are necessary, indication should be available concerning the format required, and defaults should be provided.
Example(s)	Command line syntax generally, and Internet addresses with a common prefix, are examples of complex input formats which could be avoided or shortcutted. In the former case, the rise of graphical interfaces and menu systems has largely bypassed the need for non-specialist users to have to remember long and difficult syntax sequences; the latter represent a more recent opportunity for default formats.
Exception(s)	Skilled users of powerful operating systems, and programmers in even simple languages, will continue to develop considerable skill in dealing with complex syntaxes.
Related to or affects	
Comments	The difference between graphical, menu-based and command-line interfaces can be summed up in the different demands these interface styles make of their users for remembering the type and range of inputs which they require.
Source(s)	Foley & Van Dam 1982, Williges & Williges 1984, Nielsen 1993, Sutcliffe 1995

Error Management

Attribute	ERROR MANAGEMENT No : 8
Principle	Error management
Set	User-system principles
Explanation	Prevention of erroneous user actions (before the action) is preferable to identification (after the action). User actions with potentially serious consequences should be completely prevented, or warning given before final initiation. In both cases, the consequences of the error should be indicated, along with any alternative action(s). User actions with less serious or trivial consequences should be immediately retractable; this includes the provision of a general 'undo' facility. Compound or complex inputs should be retractable and modifiable before initiation. The tone of error messages and warnings should be affirmative and positive rather than negative.
Example(s)	While it is not possible to prevent all unwanted user actions (indeed, the definition of what constitutes an unwanted action is an open-ended problem), it is possible to anticipate what they might be. Where such actions cannot be prevented, for example where a complex user input is required, incorrect or potentially serious actions should always be identified and appropriate indication given of the consequences. Serious actions (eg. which would lead to data loss) should be completely prevented, or warning given, along with a halt to further input and the opportunity to retract, before the action is finally initiated. Examples include 'Unsaved data : are you sure ?' (on Exit) and 'Save before closing ?' dialogue boxes. Less serious actions or trivial errors should be immediately retractable but not necessarily prevented. Examples include deletion of typing errors, the undoing of unwanted actions in a drawing system, etc. Compound or complex inputs, for example a collection of settings made in a dialogue box, should be both retractable in full (via a 'cancel' option) and fully modifiable, via resetting or re-writing of any inputs, before final acceptance.
Exception(s)	The only exception to the requirement to deal with all user errors is in the definition of some errors as trivial or inconsequential. Like the definition of errors themselves, this is not a simple problem; where it is not possible to come to a decision, the minimum requirement will be a warning of the consequences.
Related to or affects	Minimum retraction Accuracy of content
Comments	The problem of defining and predicting errors is a major aspect of interface design. While errors are useful, in that they point up features of the interface yet to be improved, it will never be possible to remove all potential for user errors; it is, however, possible to predict most that will occur, even if not all of them can or should be dealt with.
Source(s)	Foley & Van Dam 1982, Galitz 1985, Brown 1988, Thimbleby 1991, Hix & Hartson 1993, Nielsen 1993, Sutcliffe 1995, ISO 9241-10 (1996), Scapin & Bastien 1997, Marshall et al 1987, Dix et al 1998, Jordan 1998, Shneiderman 1998

Feedback

Attribute	FEEDBACK No : 9
Principle	Feedback
Set	User-system principles
Explanation	The status of the system (ie. what it is doing) should be visible to the user at all times. Immediate confirmation of user-initiated processes should be given, and all system processes should indicate that they are continuing. For processes of any length (>10 seconds), indication of elapsed duration or completion time should be given. All user inputs (ranging from keyboard to tracker ball) should be immediately confirmed. All continuous user input should be matched by appropriate feedback.
Example(s)	At no time should the user be left not knowing what the system is doing. In the case of user-initiated actions such as data transfer, copying, or transformation, indication should be given that such operations are underway. In the case of system-initiated operations such as batching, indication should also be given prior to commencement. The indications are that 5 seconds is the maximum time which users are prepared to wait before requiring confirmation that processing is taking place. After 10 seconds, additional information as to likely duration will be required. This can be in the form of an 'amount complete' indicator, perhaps including an estimate of actual duration; but confirmation that processing is proceeding at a detectable rate is more important; it is the knowledge that delays can be expected that is more valuable than the actual length of those delays. Immediate confirmation that user input is being accepted is equally important (one of the indications of system 'crash' is total lack of response to keyboard presses or mouse movements). The willingness of users to put up with delayed response to simple inputs such as keyboard action is now likely to be very low. Continuous input should be matched in real time by system confirmation : for example, cursor movement in a drawing package must immediately follow mouse action.
Exception(s)	When system response can be predicted, the ability to 'stack up' user inputs, without the necessity of waiting for each one to complete before initiating the next, is valuable. However, this is likely to be limited to simple inputs such as keyboard presses.
Related to or affects	Location and navigation Responsiveness
Comments	It is surprising that many well-known systems still refuse to inform their users that they are performing even simple operations, or, indeed, that they are doing anything at all. The 'watch' and 'egg-timer' icons may be over-used, but these are better than nothing (still the default response in many cases). User tolerance of slow or unresponsive systems is likely to decrease rather than increase.
Source(s)	Foley & Van Dam 1982, Williges & Williges 1984, Galitz 1985, Murphy & Mitchell 1986, Marshall et al 1987, Brown 1988, Norman 1988, Thimbleby 1991, Denley et al 1993, Hix & Hartson 1993, Nielsen 1993, Sutcliffe 1995, Zetie 1995, ISO 9241-10 (1996), Scapin & Bastien 1997, Cox & Walker 1998, Dix et al 1998, Jordan 1998, Shneiderman 1998, and others.

Location and Navigation

Attribute	LOCATIONAL INFORMATION No : 10
Principle	Location and navigation
Set	User-system principles
Explanation	It is important that the user knows 'where' in the system they are, that is, what step in an interaction sequence they have reached and what they can do from it. Thus every system state (eg. screen, window, dialogue box) should be labelled or titled. The relationship of every system state to other states should also be indicated. Thus as well as labelling, each state should indicate the range of user options which it permits. These should always include a return to the previous state; it should not be possible to enter a state from which there is no exit. Where states represent different functional modes, those modes should be clearly distinguished.
Example(s)	Labelling of every system state is important, not just for browsing systems. Each and every dialogue box, selection window, information box, etc., should have an indication of what its contents are and how they can be used. Examples include page format settings, copy options, etc. When one state allows access to another, it should be possible to return to the prior state without affecting any settings already made; there should always be an option to cancel or retract any journeys into successive states. A common convention is to indicate states from which further selections can be made and which do not yet commit the user to an action by an ellipsis (...) after the command from which they are opened.
Exception(s)	
Related to or affects	Error management Minimum retraction Minimum steps
Comments	The advent of hypertext and browser systems has emphasised the need for locational clues. While it is recognised that a single state (eg. dialogue box) can be reached in many different ways, the requirement remains to indicate what each state does. This is particularly true in a browser system, where a potentially large variety of states, with no necessary relationship between them, can be reached. The labelling of previous states will be essential in any backward movements.
Source(s)	Marshall et al 1987, Denley et al 1993, Hix & Hartson 1993, Nielsen 1993, Sutcliffe 1995, Shneiderman 1998

Attribute	LOCATIONAL MODES No : 11
Principle	Location and navigation
Set	User-system principles
Explanation	Where states (eg. windows, screens) represent different functional modes, those modes should be clearly distinguished. Where different states (eg. windows) can be opened concurrently, the 'top' or currently active state should be clearly indicated. It should also be possible to determine which states are currently open, and to switch between states.
Example(s)	An extension to the indication of states within single systems is the possibility of running more than one system concurrently. When different states represent different systems, eg. with multi-tasking, it is possible to have several major tasks or systems running at the same time. In a windowing system, different systems may be held in overlapping or occluding windows, possibly with more than one window per system. Alternatively, each system (or system window) may take up the whole screen space. In either case, it is important to be able to switch between systems (or window states), and the currently active ('top') system should be clearly apparent.
Exception(s)	It may be difficult to define a 'functionally distinct' mode, particularly when the same state has more than one function according to when opened. Systems with single workspaces, such as word processors and spreadsheets, are unlikely to have clearly separate modes. Excessive use of functional modes within single systems is to be avoided, but switching between separate applications is welcome.
Related to or affects	Location and navigation Minimum steps
Comments	The ability to switch between concurrently active applications, for example from a spreadsheet to a graphics tool and back, without having to close one system and open the other, is one of the most useful attributes of multi-tasking environments.
Source(s)	Thimbleby 1991, Hix & Hartson 1993, Nielsen 1993, Zetie 1995, Cox & Walker 1998

Choice Availability

Attribute	CHOICE AVAILABILITY No : 12
Principle	Choice availability
Set	User-system principles
Explanation	At ever system state, the range of user options should represent those which are appropriate from that state. Thus neither too few or too many choices should be available from any one state, a balance being maintained between the number of steps required for particular operations and the number of options available at each step. The range of choices at any step should not appear overwhelming or impossible to encompass. Each option available at a state should be functionally distinct from the other options at that state.
Example(s)	Menu-based systems are an obvious example (but see below); the organisation of such systems represent the way in which their functionality may be divided into coherent groups. The organisation of other systems is more open, but some organisational framework will still have to be established.
Exception(s)	
Related to or affects	Functional organisation
Comments	Interaction with a system via an interface is largely a matter of successive choices between concurrently available options. Menu interfaces are just one version of this, where the variety and number of choices is limited to a few at a time, following some organisational framework (usually hierarchical). Other interface styles present their choices in different fashions, but the principle of successive selection between alternatives remains. This discussion is therefore not limited to menu-based systems, putting them in a wider context. As to the number of options which are appropriate at each step, a likely maximum is between 7 and 9.
Source(s)	Williges & Williges 1984, Murphy & Mitchell 1986, Marshall et al 1987, Norman 1988, Zetie 1995, Jordan 1998

User Match

Attribute	TERMINOLOGY AND LANGUAGE STYLE No : 13
Principle	User match
Set	User-system principles
Explanation	Terminology and language style should match with the experience and background knowledge of the intended users. The size, format and complexity of each piece of text should be minimally sufficient to convey its intended meaning.
Example(s)	In a public access system such as an ATM (automatic teller machine), the language used should address the minimum expectations of a potentially very wide user group. For more specialised systems, terminology can have a narrower focus, but should not descend to mere jargon. If in doubt, the more general of two alternatives should be used.
Exception(s)	Even with a public access system the use of some application-specific terms is unavoidable. In such cases additional information on terminology should be provided. In specialised systems a knowledge of acronyms and other terminology should not be assumed.
Related to or affects	
Comments	Inappropriate terminology is a good indicator of a lack of concern for users of a system; it is, however, very easy to fix (and, by the same token, to implement), and can be addressed without affecting other more important aspects of system design.
Source(s)	Marshall et al 1987, Thimbleby 1991, Hix & Hartson 1993, Nielsen 1993, Scapin & Bastien 1997, Shneiderman 1998

Attribute	VISUAL METAPHOR No : 14
Principle	User match
Set	User-system principles
Explanation	The system should encourage users to create for themselves a coherent conceptual model of its functions and organisational structure. This can rely on metaphors from the user's environment or task domain, using visual and other representations of real-world objects and operations. The use of visual metaphor is particularly appropriate in this context.
Example(s)	In graphical interfaces, visual icons (small graphical representations) have been shown to be useful in encapsulating both objects, eg. 'printer', 'folder' and operations, eg. 'install', find', as well as signifying functional applications or tools. Other visual metaphors include the signification of active processes or process initiation, eg. 'process underway' (by an 'egg timer' icon, spinning wheel, etc.); folder opening' (animated window enlargement), 'menu item activated' (flashing menu text); hierarchical folder and/or file structure arrangements.
Exception(s)	Numerical data and text will resist representation in other forms. Though it is useful to combine particular kinds of message output with a visual icon and/or auditory signal (an 'earcon'), for example to signify a warning or error situation, the content of the message itself will still need to be in text form. It would be dangerous in such cases to assume that the metaphor could carry the message on its own without the text. Mixed metaphors (the use of different analogies to convey the same idea) should be avoided.
Related to or affects	Visio-perceptual load Audio-perceptual load Appropriateness of content Feedback
Comments	The graphical user interface itself can be considered as an elaborate visual metaphor, in which iconic representation and direct manipulation (representations and manipulation of real-world objects and processes in visual or other form) play an important role. However, it is important not to over-do the use of icons and visual representations; their use should be considered only where they can be said to encapsulate an aspect of the user's experience in a way which could not be easily done by other means. In particular, the possibility of incorrect user analogy (an aspect of user-system model mismatch) is an important consideration.
Source(s)	Marshall et al 1987, Brown 1988, Norman 1988, Hix & Hartson 1993, Nielsen 1993, Sutcliffe 1995, Shneiderman 1998

User Principles

These principles concern the degree to which the system caters for the user's preferences and the need to adapt the system in line with those preferences. It also concerns the way in which the system can cater for more than one style or type of user input.

The content (as opposed to the style) of the system output is also dealt with here, along with the need for emphasis of certain parts of that output.

Modifiability

Attribute	FUNCTIONAL MODIFICATION No : 15
Principle	Modifiability
Set	User principles
Explanation	The system should allow users to modify or adapt some aspects of its functional scope and organisation to fit with their level of experience or preferences. This might be in line with other, similar, applications and systems (or previous versions of the same system).
Example(s)	Page setup parameters, printing options, file saving frequencies; pull-down menu organisation and contents.
Exception(s)	The extent of the modifiability which is permitted will depend on the degree to which such modifications interact with other parts of the system, that is, the extent to which they are functionally distinct. Another issue is whether such changes extend across the whole of a system or are specific to a particular document, worksheet, etc.: for example, page layouts will usually be document-specific, while file saving specifications are more likely to be system-wide.
Related to or affects	Step modification Minimum steps
Comments	While modifiability is a positive aspect of a system's usability, it needs to be carefully controlled and directed : if all aspects of every interface were modifiable at each user's whim, one system would be as good as any other. Most systems which offer this facility maintain their overall look and feel while allowing considerable freedom for particular classes of modifications (eg. established workspaces around which many functional parameters can be adjusted). For example, some long-established systems (eg. Word) allow their complete menu sets to be re-configured, while retaining a library of commands. The ability to make wholesale changes to what otherwise would represent standard menu configurations, shortcuts, etc. is desirable in principle, but may in practice make for little consistency across systems.
Source(s)	Galitz 1985, Murphy & Mitchell 1986, Marshall et al 1987, Denley et al 1993, Hix & Hartson 1993, Sutcliffe 1995, ISO 9241-10 (1996), Scapin & Bastien 1997, Cox & Walker 1998, Dix et al 1998

Attribute	STEP MODIFICATION No : 16
Principle	Modifiability
Set	User principles
Explanation	The system should allow users to modify or adapt some aspects of its functional scope and organisation to fit with their level of experience or preferences. This includes shortcuts for frequently used operations or sequences. Default or given shortcuts may themselves be modifiable, as may the default set itself.
Example(s)	Keyboard equivalents ('accelerators') for established operations which require more complex or lengthy sequences of actions (eg. using combinations of mouse and menu use); user-definable macros for combinations of sequences; icons ('toolbars') for established operations. Standard or default accelerators, including for example the familiar Copy, Cut, Paste sets (Command-C, Command-X, Command-V), may also be modifiable, as may default sets of toolbar icons.
Exception(s)	While modifiability is a positive aspect of a system's usability, it needs to be carefully controlled and directed. Most systems which allow modifiable shortcuts will maintain a default set of commands, while allowing users to adapt or create their own sets out of this and the remaining command set. Some command sets may remain un-modifiable, for example those which would conflict with identical commands used in related applications.
Related to or affects	Functional modification Minimum steps Consistency
Comments	
Source(s)	Smith & Mosier 1986, Marshall et al 1987, Brown 1988, Hix & Hartson 1993, Nielsen 1993, Sutcliffe 1995, Cox & Walker 1998, Dix et al 1998

Flexibility

Attribute	MULTIPLE INITIATION No : 17
Principle	Flexibility
Set	User principles
Explanation	While single operations will normally be performed in one fashion, in larger systems it will often be possible to complete a particular operation from more than place in different sequences. Thus the initiation of operations may be from various steps in more than one sequence of steps.
Example(s)	The dialogue box for 'page setup' may be available from a variety of other dialogue boxes (and from other applications). Thus there may be several possible step sequences which describe a single operation such as 'set page parameters'.
Exception(s)	Many specific operations will only be performable in one fashion, with its own distinctive sequence of steps.
Related to or affects	Step modification Choice availability Consistency
Comments	The general principle of consistency (here applied to operation sequences) may be broken in favour of flexibility, where it is advantageous to provide more than one route to a particular operation, or where some operations fit naturally into (or can be usefully done in context of) more than sequence.
Source(s)	Galitz 1985, Thimbleby 1991, ISO 9241-10 (1996)

Attribute	MULTIPLE INPUTS No : 18
Principle	Flexibility
Set	User principles
Explanation	In mixed-input systems (eg. allowing combinations of mouse and keyboard), it may be possible to perform most operations using more than one input mode.
Example(s)	Some long-established systems (eg. Word, Excel) allow keyboard selection of dialogue box options in preference to the mouse. A standard default is the use of the return key for the highlighted option in a dialogue box.
Exception(s)	Keyboard alternatives can only be used where these do not conflict with default keyboard inputs.
Related to or affects	Multiple initiation Choice availability Consistency
Comments	
Source(s)	Denley et al 1993, Story 1998

Accuracy of Content

Attribute	ACCURACY OF CONTENT No : 19
Principle	Accuracy of content
Set	User principles
Explanation	Each piece of information conveyed by the system should be accurate, unambiguous and explicit.
Example(s)	Text messages, headings and labels; graphical material such as graphs, illustrations, tables; instructions; error messages.
Exception(s)	The descriptive content of some systems may need to be such that the same material is repeated in more than one place, or amplified in some ways.
Related to or affects	Terminology and language style Error management
Comments	Clearly, the need for accuracy of content is higher with information systems than with most others; however, all descriptive material should be as concise and explicit as possible. The tone of error messages, in particular, should be positive and reinforcing rather than negative and critical.
Source(s)	Marshall et al 1987, Thimbleby 1991, Hix & Hartson 1993, Nielsen 1993, Cox & Walker 1998

Salience

Attribute	SALIENCE No : 20
Principle	Salience
Set	User principles
Explanation	Some system components may be of particular salience, either exceptionally (unlike the remainder of the system, occurring rarely) or prominently (in terms of its importance for system operation, or having particular content). Such components should be given appropriate emphasis.
Example(s)	Warning of exceptional or critical error conditions; initial instructions which affect later use; indication of infrequent or unpredictable occurrences.
Exception(s)	
Related to or affects	Error management Feedback
Comments	
Source(s)	Murphy & Mitchell 1987, Brown 1988, Denley et al 1993, Hix & Hartson 1993, Nielsen 1993, Sutcliffe 1995, Shneiderman 1998

Comparative Principles

The only principle contained here is that of consistency, both between and within system components.

Consistency

Attribute	CONSISTENCY No : 21
Principle	Consistency
Set	Comparative principles
Explanation	The steps required to complete any one operation should be consistent. Movement between components should also operate consistently, such that the user should be able to predict what the result of a particular movement will be. The layout any one component or state should not differ according to the type of operation being performed. Thus the range of options available from any one state should not change, nor should the relationship between different components and sub-components. Terminology and language style should remain consistent across components and states, as should the format of informational content of the same type. All messages and feedback should be consistent in style and format. All salient (exceptional or important) content should be consistently emphasised.
Example(s)	The user actions necessary to perform a particular operation (eg. save file with a new name, delete file) should not differ according to the stage reached in interaction, or under arbitrary conditions. Once the user has learned how to move between components, such as between worksheet to help information, the nature of that movement should not change without reason. Though it may be necessary to enable or disable certain options within a component under particular conditions, the layout of each component should not change. The relationship between components (ie. the functional organisation of the system) should not change either. Navigation within online help should be consistent with that used elsewhere. Unless it is necessary for reasons of emphasis (salience), text layout and formatting should be consistent across components, and consistency of language style should be maintained.
Exception(s)	In larger systems it may be possible to initiate operations from more than one state or component. It may also be possible to navigate around the system in more than one way. It may be necessary to enable or disable certain options under certain circumstances, and while the broad layout of individual components should not change, additional sets of options, or access to particular components, may become available. Text layout or format might also be varied for emphasis (salience), as might language style. Where there are different functional modes, the layout and appearance of each mode might also be different.
Related to or affects	Multiple initiation Salience Step modification Choice availability Error management Terminology and language style
Comments	Consistency is the most commonly found, and easiest to agree upon, of all usability criteria. Unfortunately it is also one of the most difficult to define precisely. The above represents an attempt at describing the broad limits of the problem, with qualifications where appropriate. While there are sometimes good reasons to break this principle, for example for emphasis, in general consistency is to be aimed for, at least within modes.
Source(s)	Foley & Van Dam 1982, Williges & Williges 1984, Galitz 1985, Murphy & Mitchell 1986, Marshall et al 1987, Brown 1988, Thimbleby 1991, Denley et al 1993, Hix & Hartson 1993, Nielsen 1993, Sutcliffe 1995, Zetie 1995, ISO 9241-10 (1996), Scapin & Bastien 1997, Cox & Walker 1998, Dix et al 1998, Jordan 1998, Shneiderman 1998, and many others.

System Performance Principles

These principles concern the degree to which the system inhibits or imposes restrictions on the user's ability to physically manipulate its components. As well as responsive to inputs and manipulation of screen objects, it includes the ability to switch between active processes and components, and any delays between input and initiation of processes.

Manipulability

Attribute	MANIPULABILITY No : 22
Principle	Manipulability
Set	System performance principles
Explanation	The user should have the maximum freedom to switch between components and to arrange interface elements. In a multi-tasking or multi-state system, there should be means of access to states which are hidden or occluded, and means of switching between active states. In a graphical system, containing objects such as windows should not be immovable, or prevent interaction with other objects, without good reason.
Example(s)	Most WIMP systems now allow users to move and place on screen windows, dialogue boxes and other objects, unless (for example when it is essential to deal with a dialogue box before continuing) there are good reasons for not doing so. Many systems, however, continue to feature modal components (those which prevent interaction with the rest of the system, including other applications) where there is no obvious reason to do so. Many WIMP environments also allow switching between concurrently open applications, often using keyboard shortcuts.
Exception(s)	Components which confine interaction to themselves are called modal; there are some good reasons for having modal dialogues, and it is certainly easier to use them (as designers) than attempt to predict all possible ways in which non-modal switching might be done, but there are only limited situations in which they are essential. Similarly, it is sometimes necessary to prevent a window (etc.) from being moved, but in general there are few reasons to do so.
Related to or affects	Location and navigation Minimum steps
Comments	
Source(s)	Zetie 1995, Dix et al 1998

Responsiveness

Attribute	RESPONSIVENESS No : 23
Principle	Responsiveness
Set	System performance principles
Explanation	System components which are physically moveable by the user should present no resistance. There should be minimum delay in the initiation (as opposed to processing time) of system processes.
Example(s)	Windows, icons and other graphical objects should be draggable, expandable (etc.) in real time; scrollable lists or files should present no delay in response. As well as the immediate confirmation of user inputs and the need to indicate that processing is taking place, there should be no perceivable delay between acceptance of inputs which initiate a system process and the initiation of that process.
Exception(s)	Some users, particularly novices, have difficulty, for example in controlling cursor movement with the mouse, with very fast or responsive systems. Users who are used to slower systems may also have difficulty in adjusting to faster versions, for example having learned to re-direct gaze during short delays. (The skilled adaptation to predictably slower systems is an under-researched area). The problem of defining precisely where a user input ends and a system response begins is not trivial, even apart from the considerable effort needed to 'wrap' an 'active process' indicator round each and every response sequence.
Related to or affects	Feedback Manipulability
Comments	It is important to distinguish between system response times and system processing times; while both are little under the control of the interface processes, the latter may be more amenable to coding intervention than the former. While it should be less surprising that unresponsive systems are still to be found than that the designers of merely slow systems do not bother to indicate the fact, this and lack of reliable indication of processing activity remain among the most persistent aspects of poor usability.
Source(s)	Foley & Van Dam 1982, Galitz 1985, Brown 1988, Nielsen 1993, Zetie 1995, Scapin & Bastien 1997, Cox & Walker 1998, Jordan 1998, Shneiderman 1998

Perceptual and Motor Principles

These principles concern the visual and auditory load which is presented to the user by the system, the motor load (number of physical actions) which the system puts on the user, plus the clarity and contrast of and between screen images.

Visio-perceptual Load

Attribute	VISIO-PERCEPUTAL LOAD No : 24
Principle	Visio-perceptual load
Set	Perceptual & motor principles
Explanation	The visual load presented by any system component (in a multi-state system, a combination of components) should not appear excessive. This includes clutter, alignment, grouping, colour, etc.
Example(s)	Containing objects such as windows, dialogue boxes, etc., should not appear cluttered. Objects such as file icons, data input fields, buttons, with related functions should be grouped and aligned together, and this should be maintained. The number of colours used should be kept to a minimum. (Where colour is used to indicate (code for) specific meanings, this should not be the only means of doing so). Animation (moving or flashing of graphical items) should be kept to a minimum and be used for a specific purpose. The excessive use of multiple text fonts should be discouraged. In a multi-state system without manipulable (moveable, occludable) states, the number of states which are concurrently visible should be kept to a minimum.
Exception(s)	
Related to or affects	Perceptual clarity Perceptual contrast Salience
Comments	It is acknowledged that response to visual appearance is likely to be among the most subjective features of user acceptance, and that it is very difficult to give practical definitions of clutter and grouping (though there have been attempts to do so). The increasing use of graphical features to enhance appearance (eg. in web sites) is likely to make these problems (if they be so) more prevalent in the future; this author remains of the view that they are a mixed blessing and should be used with caution.
Source(s)	Williges & Williges 1984, Murphy & Mitchell 1986, Marshall et al 1987, Hix & Hartson 1993, Nielsen 1993, Scapin & Bastien 1997, Jordan 1998, Shneiderman 1998

Audio-perceptual Load

Attribute	AUDIO-PERCEPTUAL LOAD No : 25
Principle	Audio-perceptual load
Set	Perceptual & motor principles
Explanation	The auditory load presented by the system should not be excessive. Audio output should be used sparingly, such as to indicate salience, and volume levels should be adjustable.
Example(s)	Error tones; warning tones; indicators of changes in system state (eg. mail message arrival, process completion). Particular signals could be used to indicate particular types of situation.
Exception(s)	
Related to or affects	Visio-perceptual load Salience
Comments	Other than obvious uses of audible tracks, such as music teaching, speech analysis, and the aural adaptation of visual interfaces for blind or partially-sighted users, this author's view is that audible clutter and superficial sound effects are as mixed a blessing as are excessive visual bombardment. In single-user environments the additions of audible signals can be at the user's discretion, but in shared work environments they can be a positive distraction.
Source(s)	Marshall et al 1987, Hix & Hartson 1993

Motor Load

Attribute	MOTOR LOAD No : 26
Principle	Motor load
Set	Perceptual & motor principles
Explanation	The number of physical actions required of the user should be kept to a minimum. Thus apart from the requirement to keep the number of steps in any sequence to a minimum, the motor action required to accomplish a sequence should also be minimised.
Example(s)	One approach is to allow multiple input modes, such as mouse and keyboard, to be performed together (one per hand); another is to allow choice of either mode, without the need for switching. Some systems with pull-down menus allow menu access without mouse movement, keyboard equivalents for menu items, etc. The many attempts at redesigning the QWERTY keyboard (which was originally intended to slow down typists on mechanical typewriters) attest to the need for faster and more efficient input modes.
Exception(s)	Where a particular action is an established default (such as the Return key for the default option in dialogue boxes), it may be necessary to deliberately slow down practised responses whose consequences might be serious. For example, if a 'Yes' response would delete data, it would be inappropriate to use this as the default in favour of the 'No' response.
Related to or affects	Minimum steps Multiple inputs Error management
Comments	The repetitive nature of much data input, and the speed at which it can be performed, are an unfortunate side-effect of the general drive towards minimising motor actions; a technique for avoiding RSI is to vary and break up the pattern of input sequences. However, it has been shown that redesign of the steps required for certain sequences can considerably reduce the time taken for keystroke and other actions.
Source(s)	Brown 1988, Denley et al 1993, Jordan 1998

Perceptual Clarity

Attribute	PERCEPTUAL CLARITY No : 27
Principle	Perceptual clarity
Set	Perceptual & Motor principles
Explanation	All graphical objects should be both discernible and distinguishable from other objects. All text should be readable (via font size, type and line separation).
Example(s)	Icons, buttons, input fields.
Exception(s)	
Related to or affects	Visio-perceptual load Perceptual contrast
Comments	Like visual clutter, readability and clarity are likely to be a factor of user preference (not to mention monitor quality). However, very small fonts and icons should not be used without good reason, and the appearance of icons (etc.) with different meanings should be as different as possible.
Source(s)	Williges & Williges 1984, Scapin & Bastien 1997, Jordan 1998

Perceptual Contrast

Attribute	PERCEPTUAL CONTRAST No : 28
Principle	Perceptual contrast
Set	Perceptual & motor principles
Explanation	The contrast between visual objects, including text, and their background should be sufficient to discriminate them, but should not be excessive. Positive polarity (dark on light) is preferable to negative (light on dark), and common colour clashes (red-green, blue-black, blue-red, blue-yellow) should be avoided. Saturated (bright) colours should be avoided unless used to indicate exceptional salience.
Example(s)	Icons, background colours, illustrations, captions, main text.
Exception(s)	
Related to or affects	Perceptual clarity Visio-perceptual load
Comments	There is more agreement for figure-ground and colour clashes than with other vision-related aspects of user acceptance, but room for personal preferences remains. For example, positive polarity is only a majority preference.
Source(s)	Williges & Williges 1984, Scapin & Bastien 1997, Jordan 1998

User Support Principles

These principles concern the nature and extent of online assistance which is available to the user, both as general, searchable help and as context-sensitive help.

General Help

Attribute	GENERAL HELP No : 29
Principle	General help
Set	User support principles
Explanation	Online help should be provided and should be accessible from any state or component. It should be possible to search the material in more than one fashion. The amount of material should not be excessive; a minimum focus is the steps required to achieve operations. Illustrations and examples from the system should be used wherever possible. The means of navigation through help material should be consistent with that used elsewhere.
Example(s)	Help systems have improved dramatically in recent years, with hypertext browsers, backtracking, related information, index searches, etc. However, the content of the material still tends to be excessive and text-heavy, with little balance between the trivial (eg. 'how to' step guides) and the complex (eg. commonly requested information buried amongst other material). Even 'how do I' guides tend to focus on elementary features, and a 'tip wizard' soon becomes intrusive and is left off or ignored. One of the problems with extensive and complex systems is the large number of option combinations which are available from any one state, and the consequent difficulty in predicting what the result of any given set of combinations will be. A little-used feature is 'see what it does', a facility to test out a set of choices without having to commit the system to permanent changes : for example, the ability to try out a new page layout while keeping the original intact. A feature of recent systems is the 'step wizard', whereby the user is guided step by step through a sequence, the result of each set of choices being shown, with full retraction available.
Exception(s)	Some systems, particularly smaller ones such as functional add-ons, are sufficiently self-explanatory to warrant little by way of help material. However, even these would benefit from some sets of examples, and the usual solution - a 'read me' text file which may not have been installed - is usually not sufficient.
Related to or affects	Context-sensitive help Accuracy of content Choice availability Terminology and language style Consistency Requirements needs
Comments	It is well known that users tend not to use online help until obliged to, even when it is available throughout. Ironically, help is one of the most commonly cited of desirable interface features. Ease of finding relevant information is likely to be one of the major incentives to use of help material; however, finding a balance between clarity and sufficiency of information is not trivial, especially given the difficulty in predicting the background knowledge and experience which a range of users will have. One approach is to offer access to introductory help material at system startup, with an option to bypass it subsequently . However, while the initial availability of help is likely to encourage its use, it is the system, not the help material, which new users want to get to grips with; when the help material is larger than the system itself, and when it is pitched at the wrong level, the initial experience is likely to discourage further use. While extensive and detailed assistance should be available, and it is true that skilled use does not come without extensive practice and a willingness to explore, it is probably the case that the majority of users only ever discover a small proportion of any given system, and could be demonstrated to be using it at far less than optimal efficiency.
Source(s)	Hix & Hartson 1993, Nielsen 1993, Sutcliffe 1995, ISO 9241-10 (1996), Dix et al 1998, Shneiderman 1998

Context - Sensitive Help

Attribute	CONTEXT-SENSITIVE HELP No : 30
Principle	Context-sensitive help
Set	User support principles
Explanation	In addition to searchable online help, material relating to the context of use should be available from every state or system component. This is particularly necessary for error states, where the material should amplify and not supplant the information already given. Context-sensitive help should not merely replicate or allow retrieval of the existing general help material relating to the attempted operation.
Example(s)	Additional and amplifying information on the likely causes of an error and any remedial action(s); in a dialogue box, what the options are for and how to use them; information on menu items, toolbar icons, etc. A common approach to the latter is a 'balloon' or pop-up help text, appearing when the cursor is held over the item; another method for menu items would be to press a special key while holding down the mouse on the item. Pop-up panels describing terminology are commonly used in hypertext systems, etc. Dialogue box help buttons are not universally available in current systems, except for error and warning boxes; it is recommended that they be extended to all components, and that the information that they access be specific to that component (and not merely a replication of general help material).
Exception(s)	Where the appropriate portions of general help material are sufficiently brief and descriptive, and the operations on which help is sought relatively simple, general help material may suffice as context-sensitive information. However, this is unlikely to be the case in many instances, and the usual approach - access to the same large piece of general help text for a number of specific error states and dialogue boxes - is insufficient (and is likely to discourage further exploration of the help material).
Related to or affects	General help Error management Requirements needs Manipulability Accuracy of content Terminology and language style
Comments	For the reasons indicated above, most 'context-sensitive' help is not sensitive to context. If much general help material were not over-inclusive and impenetrable enough, this is compounded by having the 'help' or ? button merely open a large and vaguely-related portion of that material and expecting the user to work out (a) what part of it is relevant and (b) how it relates to the problem in hand. While it is acknowledged that it is very difficult to predict the user's intentions for each and every use of each and every part of a large system, and thus to tailor context-sensitive help to those usages, it is possible, it is believed, to do better than is commonly the case.
Source(s)	Marshall et al 1987, Brown 1988, Thimbleby 1991, Nielsen 1993, Dix et al 1998