Chapter 1: Introduction

1. Aims

The aim of this thesis is to explore the use of cognitive and other principles in interface evaluation, by investigating the use made of evaluation material which can be shown to derive from such principles. Specifically, to compare an expanded set of principles, whose usage and derivation is made more explicit, with the set of shorter and more general heuristics used in heuristic evaluation (Nielsen 1993, 1994d). The focus of these comparisons was to be the usability problems identified by both novice and experienced evaluators. Analysis will concern both the quantity and type of problems reported.

Thus

1. To investigate whether expanded evaluation principles can be used by both experienced and novice evaluators to identify more usability problems than with Nielsen's heuristics alone.

2. To investigate the claims made for Nielsen's heuristics that in a heuristic evaluation only a small number of evaluators may be required to uncover most of the problems with a given interface.

This Chapter will be concerned mainly with the first of these two themes, namely the comparison of evaluation principles and heuristics. The remainder of the Chapter discusses more general issues concerning usability and usability evaluation. The second theme will be introduced fully in Chapter 4.

2. Rationale

This thesis starts from the view that it is possible to identify a common set of principles which underlie a wide range of interface families, from command-line to web sites, and that such principles stand in contrast to mere design guidelines. The view is that while guidelines tend to be application and interface-specific, principles offer general rules which can be applied to a range of interface styles and design issues. While guidelines merely guide, principles include a rationale with more general application. The view is that such a rationale can most usefully be underpinned by principles derived from cognitive psychology. This thesis makes use of the author's own set of evaluation principles, drawn from a wide range of sources but in particular the application of cognitive psychology to interface design.

Principles will be contrasted with heuristics. Heuristics attempt to encapsulate a wider range of evaluation material into a short set of 'rules of thumb'. Heuristics can therefore be considered as principles shorn of their background rationale. It had appeared to be a relatively simple matter to demonstrate the superiority of principles over heuristics in user

evaluations. The fact that this proved to be quite difficult, and then only for restricted evaluation scenarios, makes up the first of the two themes to be explored in the thesis.

The second theme concerns the predictive power of heuristics. The proponent of one widely used set, Jakob Nielsen (citations below), has claimed that using such a set of heuristics, suitably experienced evaluators can between them identify most of the usability problems with a given interface. The thesis will also explore the validity of this claim in respect of both novice and experienced evaluators. It will investigate the ability of both heuristics and principles to enable not only the identification but also the prediction of usability problems.

3. Guidelines, Principles and Heuristics

3.1 Guidelines

There are a large number of user interface design guidelines, ranging from the very specific and low-level to the more general. Examples of the specific form include metrics for screen display and layout (Davis & Swezey 1983, Van Nes 1986). More general versions include earlier and recent guides to typical GUI and WIMP environments (Cole et al. 1985, Barclay 1986, Mayhew 1992, Bailey 1996). Two very extensive but mainly low-level collections are Smith & Mosier (1986) and Goddard (1992). In-house or product style guides such as Apple (1989), IBM (1992) and Microsoft (1995) have narrowed the focus to specific product domains.

Guidelines have been criticised for being both too general and too specific. They are too general in that it is difficult to translate their content into system-specific design guidance (Mosier & Smith 1986, Smith 1986, Potter et al. 1990), and to decide between or prioritise their advice (Maguire 1982, Mosier & Smith 1986, Preece et al. 1994, Dix et al. 1998). They are too specific when they focus on current technology at the expense of more general guidance (Smith & Mosier 1985, Mosier & Smith 1986). Some guidelines come with a double-bind, their apparent generality masking a restricted domain of application. In such cases designers may be tempted to over-generalise (De Souza et al. 1990) or to apply the guidelines incorrectly (Preece et al. 1994). The sheer size of some collections may also make it difficult to locate specific guidelines within a set (Mosier & Smith 1986).

Remedies for the first deficiency include more explicit statements of the domain of application of particular guidelines (Maguire 1982, Van Welie et al. 1999), and the inclusion of extensive examples of guideline practice (Tetzlaff & Schwartz 1991). Style guides attempt to address both deficiencies. As to the second deficiency, attempts to extend the generality of guidelines have sometimes gone beyond the point of practical application. Examples are Rubenstein & Hersh's (1984, pp219-221) "ten ideas" - no. 1 being "Know thy user. The user is always right" - and Heckel's (1984) collection of 29 'friendly design

elements', the last of which is "You need vision". The third deficiency (the problem of navigating large guideline sets) has been addressed by search and retrieval tools such as hyperSAM (Iannella 1995), for the 944 guidelines in the Smith & Mosier (1986) collection; more recently, guideline management systems such as Sherlock (Grammenos et al. 2000) and GUIDE (Henninger 2000) aim to allow both the retrieval and application of online guidelines.

3.2 Principles

One approach to the deficiencies of guidelines is to back up each application or domain-specific piece of design advice with an underlying rationale from which it can be seen to derive. That is, to combine what Smith (1986) and Dix et al. (1998) called 'design rules' with a more general 'design principle' (Hammond et al. 1987, Dumas & Redish 1993, Preece et al. 1994, Sutcliffe 1995). Thus each design specification, applied to a particular interface domain or component, would be accompanied by its appropriate principle. Each principle would be couched in sufficiently general terms to have application to a wide range of design problems, enabling designers and evaluators to generalise from one problem to another.

This is the approach that will be adopted in this thesis. The three main experiments reported in Chapters 2 to 5 will make use of different forms of principle-based material with the general form just outlined. These principles (or 'principles set') have been drawn from a wide range of sources including those deriving from cognitive psychology. The inspiration for the latter originally came from the chapter titled 'Design guidelines' by Marshall, Nelson & Gardiner (1987) in Gardiner & Christie (1987). The format and content of the principles set will be described below.

The author, then, agrees with Hammond et al. (1987, p40) that the *intent* of guidelines is to "capture some psychological principle which is sufficiently invariant across a range of contexts for the guideline to remain useful". The problem with guidelines is that they rarely manage to do this. If they are to be much more than a basis for checklists (and there are some - Ravden & Johnson (1989) and Clegg et al. (1988) Chapter 10 are examples) then guidelines need to address the "novel and integrative conceptual issues" (Tetzlaff & Schwartz 1991) or "meta-knowledge" (De Souza et al. 1990) out of which specific design guidance might be distilled. They also need to be expressed in sufficiently colloquial form to be of use to designers, and to include any constraints on their application (Thimbleby 1984). In short, guidelines need to support their rules with sufficient 'theoretical evidence' to enable trade-offs between possible designs, particularly in the early stages of the design life cycle (Dix et al. 1998).

Marshall et al. (1987) offered some "initial steps" towards an "opening set" of such guidelines, derived from cognitive psychology principles. These are arranged into a "category framework of relevant concepts" or 14 "sensitive dimensions" (ibid., p222),

namely: Design of procedures and tasks, Analogy and metaphor, Training and practice, Task-user match, Feedback, Selecting terms, wording and objects, Consistency, Screen design, Organisation, Multimodal and multimedia interaction, Navigation, Adaptation, Error management, Locus of control. This thesis will demonstrate that such categorisations can have implications for the usability problem identification process. The focus of this Section is more on the content and format of such guidelines than their organisation.

Most of the Marshall et al. (1987) "sensitive dimensions" (guideline categories) have the same format. Figure 1.1 is an illustration. First comes an overview of the category and the underlying principle(s) from which the guidelines contained in it derive. Next, a concise statement of the principle(s). Then follows the authors' interpretation of the principle(s) in cognitive psychological terms. Next are offered one or more examples of how the principle(s) might apply in practice (usually office-related computerised tasks). Then a statement of the collected guideline(s) in compact form. Finally, some comments are often offered (though not in this case) on how the guideline(s) might be applied in wider or different contexts. The guidelines themselves are listed separately, referenced to the Chapter (of Gardiner & Christie 1987) from which they derive.

Unlike most guidelines, which would at best include only the last two parts of this example, such a format allows the designer or evaluator to extract both an underlying principle - in this case faulty generalisation c.f. its converse, overgeneralisation (the mode capture error) - and to see how that principle might apply in a familiar example. While we might wish the principle(s) terminology to be a little less cryptic - the use of "scoring" and "dominant functional characteristics" for example - we can see how the format can be extended to the other 13 categories (or "dimensions") in the Marshall et al. (1987) collection. Many of these categories feature in the author's own set of 30 principles, which will be described shortly.

In the author's view, the content of the above example from Marshall et al. (1987) is sufficient without most of the seven (of the 156 in Gardiner & Christie 1987) attendant guidelines which these authors separately derived (the above 'sample' is intended to illustrate just two of the seven). However, the author would append to the above pair guidelines 116 and 121 to 122 (Marshall et al. 1987 p267):

-- All procedures and component elements in a dialogue should have consistent properties, names and relationships with other elements; and be used consistently throughout the dialogue.
-- When consistency is not possible throughout the interface be consistent within the immediate domain (for instance, task, application, etc.).
-- If an action or procedure has to be used inconsistently, the user should be warned of the inconsistency of the situations where it holds.

thus making a set of five in this category. It will be clear in the next Section that the author's own Consistency guideline (or 'principle') derives from such a set.

7. Consistency

Consistency is fundamental to effective interface design. It contributes to usability in a number of ways. For example, it facilitates learning, lessens the number of errors made and helps the user to develop an accurate system model. It is rarely possible to be completely consistent within and between the applications of a system. So the designer will often have to determine the priorities of the system and make tradeoffs accordingly. For example, should the designer be rigidly consistent if the result is a slow and inappropriate procedure when and an inconsistent procedure would achieve the goal more efficiently? The decision has to be made based on the function of the system and users for whom it is intended.

The guidelines in this category emphasise the importance of consistency in certain domains, and identify areas where tradeoffs could be made: for example, it is suggested that consistency within an application should take priority over consistency between applications.

Sample

Principle(s)

-- If the scoring between a system's dominant functional characteristics and those of other systems is low, then the probability of the user making the correct assumptions about the dominant functional characteristics of each of the major features and operations of the system will be reduced.
-- If the probability of the user making the correct assumptions about the dominant functional characteristics of each of a system's major features and operations is low, then the probability of capture will be increased.

Interpretation

If there are large inconsistencies between the different modes, application programmes or versions of a system, then the user's ability to generalize from one to another is reduced. If users assume that system aspects are similar, they will be prone to 'capture', that is, errors in which they overgeneralize and assume that actions taken in one system mode will have the same effect in other, related, system modes.

Example(s)

If most of the systems in the office market employ the 'desktop metaphor' and an apparently similar interface is introduced in another product or context, but this interface contains different types of command and consequences of those commands, and different relationships between elements in the interface, then users will find it particularly difficult to use and learn. (Possibly because they have different stored knowledge for what desktop systems are all about and the new interface breaks all of the rules stored with such knowledge.)

Guideline(s)

-- If the intended interface is functionally different from the other extant interfaces make it look different.
-- Within the same system, ensure that different modes, application packages or versions either share common operational attributes or that they look sufficiently different from one another that users will not be tempted to overgeneralize.

Figure 1.1. The "sensitive dimension" of Consistency (Marshall et al. 1987 pp238-239).

3.3 The Author's Principles Set

See Appendixes B to D for transcripts of the full and abridged versions used in Experiments 1 to 3.

The author's own 'principles set' derives its inspiration from Marshall et al. (1987), but uses a slightly different format. This is an adapted version of that created by Smith & Mosier (1986, originally 1984) in their widely quoted but little seen[1] guidelines collection.

The full format[2] of the Smith & Mosier (1986) guidelines is illustrated by the example in Figure 1.2. It is similar to that used by Marshall et al. (1987), but includes exceptions and

1 The author has not seen a paper copy either.
2 Not all of the 944 guidelines offer every part of this format.

2 DATA DISPLAY
2.5 Format
2.5/1 Consistent Format

Statement
Adopt a consistent organization for the location of various display features from one display to another.

Comment
The objective is to develop display formats that are consistent with accepted usage and existing user habits. Consistent display formats will help establish and preserve user orientation. There is no fixed display format that is optimum for all data handling applications, since applications will vary in their requirements. However, once a suitable format has been devised, it should be maintained as a pattern to ensure consistent design of other displays.

Exception
It might be desirable to change display formats in some distinctive way to help a user distinguish one task or activity from another, but the displays of any particular type should still be formatted consistently among themselves.

Example
One location might be used consistently for a display title, another area might be reserved for data output by the computer, and other areas dedicated to display of control options, instructions, error messages, and user command entry.

Reference
Brown et al. 1983 § 1.1 1.8.5
Engel Granda 1975 § 2.2.5 2.3 2.3.3
MIL-STD-1472C 1983 § 5.15.3.2.1 5.15.3.3.4
Foley Van Dam 1982
Stewart 1980

See also: 4.0/6 Consistent Display Format

Figure 1.2. Guideline 2.5/1, Consistent Format, of Smith & Mosier (1986) (in hyperSam - Iannella 1995; also reproduced in Smith 1986).

source citations. It also includes a 'See also' (reference to related guidelines). The Smith & Mosier guidelines were prepared by the MITRE Corporation and sponsored by the United States Air Force. They comprise a total of 944 guidelines, covering six main categories, namely Data Entry, Data Display, Sequence Control, User Guidance, Data Transmission and Data Protection (from hyperSAM - Iannella 1995). Though written for 1980s generation interfaces (thus mainly command-line and menu-based), the scope of the collection encompasses a range of dialogue types including natural language and 'graphic interaction' (icons, direct manipulation, and 'control options'). The accompanying material is explicit concerning the need for the tailoring and prioritising of the content, and in stating that guidelines cannot take the place of experience. It also makes it clear that the interpretation of any guideline involves consideration of exceptions and trade-offs. The Smith & Mosier format is close to what Sutcliffe (1995, pp8-9) called an 'engineering principle' - one which draws on an 'applied science' such as cognitive psychology but which includes 'scoping rules' and 'caveats' to generalisation.

The author's full principles set currently totals 30 sub-principles (dubbed 'attributes'), organised into seven main categories or sets (originally based on earlier work on a usability audit). The contents are shown in Figure 1.3. Each set has one or more principle, but not all principles have more than one attribute (in such cases the attribute name is the same as the

Requirements and Functionality Principles
- Principle: Requirements Match
  - Attribute: Functional Needs [1]
  - Attribute: Requirements Needs [2]
- Principle: Functional Utility
  - Attribute: Functional Organisation [3]
  - Attribute: Functional Provision [4]

User - System Principles
- Principle: Navigational Effort
  - Attribute: Minimum Steps [5]
  - Attribute: Minimum Retraction [6]
- Principle: Memory Load [7]
- Principle: Error Management [8]
- Principle: Feedback [9]
- Principle: Location and Navigation
  - Attribute: Locational Information [10]
  - Attribute: Locational Modes [11]
- Principle: Choice Availability [12]
- Principle: User Match
  - Attribute: Terminology and Language Style [13]
  - Attribute: Visual Metaphor [14]

User Principles
- Principle: Modifiability
  - Attribute: Functional Modification [15]
  - Attribute: Step Modification [16]
- Principle: Flexibility
  - Attribute: Multiple Initiation [17]
  - Attribute: Multiple Inputs [18]
- Principle: Accuracy of Content [19]
- Principle: Salience [20]

Comparative Principles
- Principle: Consistency [21]

System Performance Principles
- Principle: Manipulability [22]
- Principle: Responsiveness [23]

Perceptual and Motor Principles
- Principle: Visio-Perceptual Load [24]
- Principle: Audio-Perceptual Load [25]
- Principle: Motor Load [26]
- Principle: Perceptual Clarity [27]
- Principle: Perceptual Contrast [28]

User Support Principles
- Principle: General Help [29]
- Principle : Context-Sensitive Help [30]

Figure 1.3. The contents of the author's principles set. Numbers refer to sub-principles or 'attributes'.

principle name). Thus principle Requirements Match has two attributes (sub-principles), namely Functional Needs (no. 1) and Requirements Needs (no. 2), while principle Error Management (no. 8) as yet has no attributes.

The scope of the full set is similar to but wider than that developed by Bastien and Scapin, discussed in Section 4.3. The structure (originally based on earlier work on a usability audit) has and will continue to evolve as new content is incorporated. It is only one of many possible organisations, of which Bastien's & Scapin's is an alternative.

The format of each attribute is illustrated in Figure 1.4, Consistency (no. 21). (See Appendix D for the full set.) Explanations are couched as far as possible in domain-independent terms (e.g. "components", "states"), but without resorting to the full Marshall et al. (1987) treatment: thus jargon is usually relegated to the Comments, while Example(s) flesh out the

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 (e.g. 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 and 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 (i.e. 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 & 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.

Figure 1.4. Attribute (principle) no. 21, Consistency, of the author's principles set.

terminology. Exception(s) are as complete as considered necessary. 'Related to or affects' refers to other attributes. Cited sources are described below and included in the References to the thesis.

The three main experiments reported in Chapters 2 to 5 make use of different versions and/or subsets of the full principles set. (See Appendixes B to D for transcripts.) Experiment 1 (Chapter 2) uses a three-page version (as then developed) covering 26 principles but the same seven categories as shown in Figure 1.3. In this version each

principle comprised just one paragraph, generally a statement of the principle followed by examples. By Experiment 2 (Chapter 3) this had been expanded into the full version, covering 23 principles and 30 attributes, described above (some of the original 26 principles having been merged and others split apart). In Experiment 2 the size of this full version proved unwieldy for evaluators to navigate (like some other guideline collections), so an additional summary, featuring only the Explanation part of each attribute, was provided. It may not come as a surprise to the reader to learn that most subjects found this to be difficult, even with prior training. For this reason, in Experiment 3 (Chapter 5) subjects were provided with just two attributes (namely Consistency and Error Management) which were considered relevant to the tasks to be performed in that experiment.

The contents of the principles set draw on many sources including Marshall et al. (1987). For the most part, these represent principles collections (whether or not actually *called* "principles") which can be said to be domain- or application-independent and which offer some underlying rationale. Avowedly domain-specific principles have been employed where some general application can be deduced. For the rest, use has been made of the guidelines literature, including Smith & Mosier (1986) and others. The main sources cited are listed (in chronological order) in Figure 1.5. Other than Marshall et al. (1987), Murphy & Mitchell (1986) is most explicit in its drawing on cognitive psychology. Nielsen (1993), ISO 9241-10 (1996), Scapin & Bastien (1997) and Shneiderman (1998) will be discussed below.

It will be apparent that this version of a principles set is by no means complete (for example, it lacks 'user control'[3]). It is also evident that much more could be included by way of illustration of the examples offered. It is acknowledged that future set(s) will have to be tailored towards particular domains, for example multimedia (for which Johnson & Nemetz 1998 offer some different principles) and internet browsers (which might draw on Ratner et al. 1996 and Tauscher & Greenberg 1997). The diversity of the alternatives listed in Figure 1.5 suggests that no single all-inclusive set is likely to emerge. However, it appeared to be relatively simple to demonstrate that a larger set (such as the author's) whose scope incorporated that of a shorter set (such as Nielsen's heuristics, described below) would be capable of addressing a wider range of issues than the shorter set. Experiments 1 to 3 address this proposition.

[3] This and other principles have been in, and out, of earlier versions: their absence indicates that (for the moment) the author believes them to be accounted for elsewhere in the set.

Foley & Van Dam 1982 pp217-242 (seven "design principles" for "interactive computer graphics": give feedback, help the user learn the system, allow backup and accommodate errors, control response time, design for consistency, structure the display, minimize memorisation).
Williges & Williges 1984 (six "fundamental design principles" for "interactive computer systems" - compatibility, consistency, brevity, flexibility, immediate feedback - plus extensive low-level guidelines).
Galitz 1985 (11 brief "desirable qualities of a system", followed by a mixed collection of general and interface component-specific guidelines, including: design tradeoffs, flexibility, information load, response time).
Murphy & Mitchell 1986 (18 "cognitive attributes" for "highly automated control systems" and "real-time decision-making environments", organised into three groups: knowledge structures in memory (schemas), active processing, problem solving).
Marshall et al. 1987 (14 "sensitive dimensions", listed above; the first source for the author's principles set).
Brown 1988 (12 "general concepts" underlying the interface component-specific guidelines which follow, including: allocation of functions, consistency, physical analogies, stimulus-response compatibility, providing multiple paths).
Norman 1988 pp8-18 (the "fundamental principles for designing for people" - providing good conceptual models and visibility - plus affordance, mapping, feedback).
Thimbleby 1991 (24 "design choices and principles" for "interactive systems", including: consistency, display inertia, modes, minimising the user's memory load).
Denley et al. 1993 (12 classes of "generic principles" for integrated broadband communication systems, including: compatibility, coherence, simplicity, salience, reversibility, controllability, flexibility, feedback, support orientation).
Hix & Hartson 1993 pp27-55 (26 interface "guidelines to ensure usability", including: keep the locus of control with the user, use modes cautiously, make user actions easily reversible).
Nielsen 1993 pp115-163 (the 10 "usability heuristics", described in full below, which are compared in this thesis with the author's principles set).
Sutcliffe 1995 (seven "basic principles" for HCI design - consistency, compatibility, predictability, adaptability, economy and error prevention, user control, structure, plus three "basic concerns of design quality" and eight "principles of dialogue design").
Zetie 1995 ("fundamental principles" for GUI design - visibility, feedback, mappings, cues and affordance, the burden of interpretation, input and output gaps, closure, degrees of freedom, modes - plus 10 cognitive principles; also multi-document interface (windowing) principles).
ISO 9241-10 (1996) (seven high-level "dialogue principles": suitability for the task, self-descriptiveness, controllability, conformity with user expectations, error tolerance, suitability for individualisation, suitability for learning). Part 10 of a 17-part standard for office work with visual display terminals.
Scapin & Bastien 1997 (eight "ergonomic criteria for evaluating the ergonomic quality of interactive systems" - guidance, workload, explicit control, adaptability, error management, consistency, significance of codes, compatibility - and 13 sub-criteria).
Cox & Walker 1998 pp174-187 (20 "general principles" for user interface design, including: consistency, closure, easy reversal).
Dix et al. 1998 pp162-175 (three "principles to support usability": learnability, flexibility and robustness, with 14 sub-principles; plus a later chapter on user support (help) systems).
Jordan 1998 pp25-38 (10 "principles of usable design" - consistency, compatibility, consideration of user resources, feedback, error prevention and recovery, user control, visual clarity, prioritisation of functionality and information, appropriate transfer of technology, explicitness).
Shneiderman 1998 pp67-90 (three overall principles - recognise the diversity, use the eight golden rules of interface design (outlined below), prevent errors - and sub-principles; plus later guidelines for data display and data entry).

Figure 1.5. Main sources cited in the author's principles set.

3.4 Heuristics

Interface design heuristics attempt to encapsulate the principles which underpin guidelines or other material into shorter sets of 'rules of thumb'. They are meant to accompany, rather than replace, that guidance material, with which, it is assumed, designers are already familiar. In that sense, heuristics can be considered as principles shorn of their background rationale. Heuristics thus rely to a greater or lesser extent on the ability of designers to relate their contents to the larger material on which they are based.

Many of the guidelines collections cited in Figure 1.5 include such a summary, which could therefore be extracted as heuristics. The most well-known is Shneiderman's (1998,

originally 1987) "eight golden rules" for interface design (listed as "Principle 2" in the 1998 version), namely: Strive for consistency, Enable frequent users to use shortcuts, Offer informative feedback, Design dialogs to yield closure, Offer error prevention and simple error handling, Permit easy reversal of actions, Support internal locus of control, Reduce short-term memory load. Even style guides such as Microsoft's (1995) Windows®, IBM's (1992) CUA and Apple's (1989) Hypercard® guidelines include, respectively, a "guidelines summary", "goals and design principles", and "ten general principles" which could serve the same purpose. However, Shneiderman (1987) appeared to have intended his "golden rules" to stand relatively alone, since only short sections of that book are devoted to further guideline exposition. Shneiderman (1998, p74) later describes the rules as "derived heuristically from experience".

This origins of Nielsen's heuristics, in a paper by Rolf Molich and Nielsen (1990)[4], appear to support the view that they are derived mainly from experience. The then nine heuristics are described as "a short checklist of usability considerations in a good dialogue" (Molich & Nielsen 1990 p339), and, though "corresponding to similar principles described by others [such as the Apple 1987 style guide]", the checklist "reflects our [Molich's & Nielsen's] personal experience" (bid., p339). The heuristics are first described in short paragraphs of no more than 50 words, comparable in format to Shneiderman's ten. Only later, in Nielsen's (1993) book, are they fleshed out with examples and their origins made more explicit. By 1994 (in Nielsen 1994d and several other papers) they and the heuristic evaluation method had become fully formed.

It is not clear from Nielsen's descriptions whether the heuristics (as used in heuristic evaluation) are 'rules' or 'principles':

> "For the discount [usability engineering] method I advocate cutting the complexity [of guidelines containing thousands of rules] by two orders of magnitude, to just 10 *rules*, relying on a small set of broader *heuristics* such as the basic usability *principles* listed in Table 2 [of Nielsen 1993] and discussed in Chapter 5 [of Nielsen 1993], Usability Heuristics". (Nielsen 1993 p19, thesis author's italics.)

> "During the [heuristic] evaluation session, the evaluator goes through the interface several times and inspects the various dialogue elements and compares them with a list of recognised usability *principles*. These *heuristics* are general *rules* that seem to describe common properties of usable interfaces ..." (Nielsen 1994d p28, thesis author's italics.)

Such conflation lends support to the view that the heuristics are, after all, meant to stand on their own, as 'rules', even 'checklists', without supporting principles or other background material. The role of this other material, and the extent to which evaluator expertise should compensate for its absence, remains unclear.

However, the status of heuristics as evaluative criteria *is* clear, and that is the form in which they (meaning both Nielsen's heuristics and the author's principles set) will be used in this

4 Nielsen's (1989) paper on 'discount usability engineering' cites what appears to be a review version of Molich & Nielsen (1990).

thesis. In the event, Experiments 1 to 3 (Chapters 2 to 5) differed in both the size and scope of the materials compared and the degree of training offered in their use. In that sense, these experiments *did* vary the amount of 'background material' which accompanied the 'heuristics'. However, the method of use was held the same within each experiment, thus representing comparisons of *materials* rather than *methods*. The intention of these experiments was not to compare alternative evaluation methods (though such studies will be summarised below). The aim was to attempt some closure on just one part of a single approach, namely the size and scope of the 'heuristics' which might be used in a heuristically-based evaluation.

3.5 Nielsen's Heuristics

Experiments 1 to 3 will compare the author's principles set with Nielsen's (and Molich's) ten usability heuristics, as described in Nielsen (1993) and Nielsen (1994d). The primary source is the set of nine heuristics which appeared in Molich & Nielsen (1990), plus the tenth, namely 'Help and Documentation' (Nielsen 1993 p20), added in 1991 (Nielsen 1994d p29). This original set (slightly shortened but not substantially changed in the 1993 and 1994 versions) is reproduced in Figure 1.6. Appendix A shows the numbered transcript used in the experiments.

The Molich & Nielsen (1990) version of the heuristics (plus Help and Documentation) was chosen because it is the one most often cited or used in studies by its originators (e.g. Nielsen 1989, Nielsen 1990c, Nielsen 1990e, Nielsen & Molich 1990, Nielsen 1992, Nielsen et al. 1992, Nielsen & Phillips 1993, Molich 1994) and, directly or indirectly, by other researchers, e.g. Jeffries et al. (1991), Bailey et al. (1992), Desurvire & Thomas (1993), Virzi et al. (1993), Cuomo & Bowen (1994), Dutt et al. (1994), Iannella (1994), Karat (1994), John & Marks (1997).

Nielsen's heuristics were used for two reasons. First, heuristic evaluation has been a focus of much research (just some of which is cited above), including Nielsen's own. The second reason is the claims made for the heuristics by their proponents. The weaker claims include the following.

> "These [basic usability] principles [...] can be used to explain a very large proportion of the problems one observes in user interface designs." (Nielsen 1989 p397, Nielsen 1993 p19)
>
> and
>
> "These nine principles correspond more or less to principles which are generally recognised in the user interface community ..." (Nielsen & Molich 1990 p250).

Simple and Natural Dialogue
Dialogues should not contain irrelevant or rarely needed information. Every extraneous unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. All information should appear in a natural and logical order.

Speak the User's Language
The dialogue should be expressed clearly in words, phrases, and concepts familiar to the user rather than in system-oriented terms.

Minimize the User's Memory Load
The user's short-term memory is limited. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. Complicated instructions should be simplified.

Be Consistent
Users should not have to wonder whether different words, situations or actions mean the same thing. A particular system action - when appropriate - should always be achievable by one particular user action. Consistency also means coordination between subsystems and between major independent systems with common user populations.

Provide Feedback
The system should always keep the user informed about what is going on by providing him or her with appropriate feedback within reasonable time.

Provide Clearly Marked Exits
A system should never capture users in situations that have no visible escape. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue.

Provide Shortcuts
The features that make a system easy to learn - such as verbous [verbose] dialogues and few entry fields on each display - are often cumbersome to the experienced user. Clever shortcuts - unseen by the novice user - may often be included in a system such that the system caters to both inexperienced and experienced users.

Provide Good Error Messages
Good error messages are defensive, precise and constructive. Defensive error messages blame the problem on system deficiencies and never criticise the user. Precise error messages provide the user with exact information about the cause of the problem. Constructive error messages provide meaningful suggestions to the user about what to do next.

Error Prevention
Even better than good error messages is a careful design that prevents a problem from occurring in the first place.

Help and Documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Figure 1.6. Nielsen's and Molich's ten heuristics, as appeared in Molich & Nielsen (1990) and listed (with the addition of 'Help and Documentation') in Nielsen (1993) and Nielsen (1994d).

Stronger claims include the assertions that

> "Almost all usability problems fit well into one of the categories." (Molich & Nielsen 1990 p339)

and

> "These nine usability principles should be followed by all user interfaces." (Nielsen 1990e p111).

The authors of these statements elsewhere qualify them. The original nine (or ten) heuristics are merely "similar to" or "correspond to" other usability principles or guidelines (Molich & Nielsen 1990, Nielsen 1990e, Nielsen & Molich 1990, Nielsen 1993, 1994d). Additional "relevant" problems, not elicited by the heuristics themselves, may "obviously" be considered during an evaluation (Nielsen 1993 p158, Nielsen 1994d p28). Further

category- or domain- specific heuristics might also be developed (Nielsen 1994d p29). And there may of course be better or more inclusive versions. The few attempts to develop and test such versions will be described below.

4. Comparison of Guidelines, Principles and Heuristics

To the author's knowledge, the only two attempts to compare principles or guidelines with Nielsen's heuristics are in Jeffries et al. (1991) and Cuomo & Bowen (1994). These were both between-method studies whose wider context will be summarised in Section 5.4.1 (along with other such studies which have included heuristic evaluation). Nielsen himself (in Nielsen 1994a) has proposed a new set of nine heuristics, but only one validation attempt (Muller et al. 1995) is known. Again to the author's knowledge, the only attempt to compare or validate alternative principles sets is the series of experiments by Bastien and Scapin (most recently reported in Bastien et al. 1999).

4.1 Guidelines versus Heuristics

Jeffries et al. (1991), Miller & Jeffries (1992) compared heuristic evaluation with guidelines as part of a study of four methods (the other two being cognitive walkthrough and usability testing). The heuristics used were probably the original Molich & Nielsen (1990) set; the 62 (unspecified) guidelines were a Hewlett-Packard collection. Though the guidelines found most recurring and general problems, heuristic evaluation found more (serious and non-serious) problems, and at the lowest cost, than any of the four methods. However, it is not specified how the guidelines were used (nor, indeed, now the heuristic evaluation was carried out, and if this was done differently), so it is difficult to draw firm conclusions from this study regarding guidelines versus heuristics.

In a later study focused on direct manipulation, Cuomo & Bowen (1992, 1994) compared heuristic evaluation, guideline evaluation and cognitive walkthrough against the results of a usability test. The heuristics are again unspecified, but probably the Molich & Nielsen (1990) set. The guidelines were the Smith & Mosier (1986) collection described above. An analysis based on Norman's (1986) user activity model (Norman's 'theory of action') found that the guidelines had the best spread of unique problem types across the stages of the model (and most types overall). These findings imply that guidelines allow a wider range of problem identification than heuristics. However, in this study the two heuristic evaluators were allowed to proceed in any manner they wished, whereas the single guideline evaluator worked more systematically. It is not clear, therefore, if the results were confounded by differences in the methods of use, nor what would have been the effect of introducing additional evaluators.

4.2 Comparing Heuristics

Nielsen (1994a) created a new "candidate set" of nine heuristics by matching each of the 101 heuristics from seven published sets including Molich & Nielsen (1990) against the 249 usability problems collected from 11 earlier (unspecified) studies by the same author. A factor analysis revealed a list of seven most important ('summarising') factors. Two additional heuristics with the "widest coverage" made up the nine. These plus Help and Documentation are listed in Figure 1.7.

Visibility of system status
Match between system and the real world
User control and freedom
Consistency and standards
Error prevention
Recognition rather than recall
Flexibility and efficiency of use
Aesthetic and minimalist design
Help users recognise, diagnose, and recover from errors
Help and Documentation

Figure 1.7. Nielsen's (1994a) "candidate set" of nine new heuristics, plus Help and Documentation, listed in Nielsen (1994d) p30.

To the author's knowledge this alternative set has not been used in a comparative study. However, examination of the contents (in Nielsen 1994d) reveals what appears to be a complete match with the original list in Figure 1.6. Of the first three, 'Visibility of system status' is very close to 'Provide feedback', 'Match between system and the real world' is similar to 'Speak the user's language', and 'User control and freedom' may be considered to mirror 'Provide clearly marked exits'. Thus a validation of the complete set would seem to be comparing like with like. However, some, notably 'Recognition rather than recall', are couched in more principle-oriented terms, implying that small-scale comparisons might be profitable.

As for additional or domain-specific heuristics, Nielsen (1990d) has identified further 'navigational dimensions' and 'elements' for the evaluation of hypertext. Elsewhere, Karat et al. (1992) used a set based on Nielsen (1993) (minus 'error messages' and 'error prevention', plus four additional heuristics) in their comparison of (team and individual) heuristic-based walkthroughs and user testing. The experienced evaluators who use the heuristics reported that they were of limited added value in identifying usability problems. Muller et al. (1995) have proposed an additional four heuristics to encompass what they call the 'process-oriented' (user's work) aspects of usability. Their set of 14 heuristics for 'participatory heuristic evaluation' would include Nielsen's 'new' (1994a) set, as listed above (Figure 1.7), plus their "validated" extensions.

4.3 Comparing Principles

During the 1990s, Bastien and Scapin (principally and in English, Bastien & Scapin 1992, Bastien & Scapin 1995, Bastien et al. 1996, Scapin & Bastien 1997, Bastien et al. 1999) have developed and validated a set of 'ergonomic criteria for evaluating the ergonomic

quality of interactive systems'. Their eight ergonomic criteria and sub-criteria (making 18 'elementary criteria') are based on the set originally devised by Scapin (1990). This was drawn from a range of 'existing human factors guides', namely an earlier collection by Scapin (1987), Smith & Mosier (1986), Williges & Williges (1984), *Ergonomics Abstracts*, and "previous unpublished work" (Scapin 1990 p208 and p209). See Figure 1.8.

1. Guidance
    1.1. Prompting *
    1.2. Grouping and distinguishing items
        1.2.1. Grouping and distinguishing items by location *
        1.2.2. Grouping and distinguishing items by format *
    1.3 Immediate feed-back *
    1.4 Legibility *
2. Workload
    2.1. Brevity
        2.1.1. Conciseness *
        2.1.2. Minimal actions *
    2.2. Information density *
3. Explicit control
    3.1. Explicit user actions *
    3.2. User control *
4. Adaptability
    4.1. Flexibility *
    4.2. User's experience *
5. Error management
    5.1 Error protection *
    5.2. Quality of error messages *
    5.3. Error correction *
6. Consistency *
7. Significance of codes *
8. Compatibility *

Figure 1.8. The eight 'ergonomic criteria' developed by Bastien and Scapin. The 18 'elementary criteria', marked *, are those which cannot be further sub-divided (Scapin & Bastien 1997 p222).

The 18 elementary criteria (in a version very similar to Figure 1.8) have been validated by having specialists and non-specialists match the criteria against a set of 36 usability problems previously found with a database application (Bastien & Scapin 1992). The results showed that half of the criteria (among them 'Feedback', 'Error protection', 'Grouping and distinguishing by format') were relatively robust and might remain unchanged, while the other half would need improvement. A "modified" set (though with the identical structure) was later used in an evaluation exercise (Bastien & Scapin 1995). In that study, two groups of usability specialists evaluated a different database system using either the criteria or control materials. The difference between usability problem counts taken when using "solely their expertise" and when later using the control materials (Control group) or the criteria (Criteria group) was significantly higher for the Criteria group than the Control group. The Criteria group had both a higher proportion of the 503 "known" problems and a higher proportion of problems in common. In a later study (Bastien et al. 1996, Scapin et al. 1999), the same elementary criteria were compared with Part 10 of the 17-part international standard for Ergonomic Requirements for Office Work with VDTs (ISO 9241), titled 'Dialogue Principles' (ISO 9241-10 (1996)). No difference was found between a Control group and the ISO group in the number or proportion of usability problems found with the database

system. However, both the number and proportion of problems found by the Criteria group were significantly higher than those of the Control group.

The Bastien & Scapin criteria are the only validated set of 'principles' known to the author. A set of scope comparable to (and in part based on) that of Bastien & Scapin is that of Jean Vanderdonckt (outlined in Bodart & Vanderdonckt 1993, 1995). However, to the author's knowledge this set has not been validated or compared with other sets. The author has located only a summary contents (via ftp from Jean Vanderdonckt).

Much of what Bastien & Scapin have attempted has been replicated in this thesis. The principles format adopted by the author is very similar to theirs (first illustrated in Bastien & Scapin 1995), though arrived at independently (and based on the same source, namely Smith & Mosier 1986). Experiment 1 of this thesis used both experienced and novice subjects. Experiments 1 to 3 included control groups not provided with evaluation materials. The Bastien et al. (1996) study is very similar to Experiment 1 in its comparison of two forms of evaluation materials plus a control (and, as we shall see, in its results). In the sense that the ISO 9241 Part 10 'Dialogue Principles' are more like general (if longer than typical) heuristics than a 'full' principles set[5], the 1996 study is yet more similar to Experiment 1. Like Experiment 2, this study gave subjects prior exposure to the evaluation materials (though unlike Experiment 2, a training session was not used). The proportion of problems found by individual evaluators will also be a major consideration of this thesis (Chapter 4 onwards).

However, there are differences between Bastien & Scapin and this thesis in not only the scope of the principles sets but also the procedures used. The author's set consists of 30 sub-principles (attributes) (Figure 1.3) compared with Bastien & Scapin's 18 elementary criteria (Figure 1.8). While there are clear matches (for example between Minimum Steps [5] and Minimal actions [2.1.2], and between Feedback [9] and Immediate feed-back [1.3]), there are more attributes which do not feature in the criteria (for example Responsiveness [23] and Perceptual Clarity [27]) than vice versa. The range of sources used by the author (Figure 1.5) is also wider than those cited by Bastien & Scapin (both including Smith & Mosier 1986 and Williges & Williges 1984). Experiments 1 to 3 subjects were free to identify usability problems not directly derived from the evaluation materials (in Experiment 2 this was explicitly stipulated), whereas in Bastien & Scapin's (1995, 1996) studies more attempt was made to confine subjects to the materials. Experiments 1 to 3 are between-subjects studies, whereas in Bastien & Scapin (1995) a partial within-subjects design was used (perhaps biasing the resulting problem reporting - of a video replay of each subject's own interactions - towards previously identified problems).

---

[5] Whether ISO 9241 Part 10 is a principles set or a heuristics set is probably not an issue worth pursuing. It is certainly very general, and avowedly domain-independent. Its application in any instance is stated to depend on "the characteristics of the intended user of the system, the tasks, the environment and the specific dialogue technique used" (ISO 9241-10 (1996) p2). Dix et al. (1998, p190) characterise standards as "high in authority and limited in application".

More importantly, this thesis does not compare principles sets, nor does it set out to validate the author's own set. Even if some superiority (such as problem count) could be found for one large and wide-ranging set over another, it would be very difficult to determine what it was about the winning set which made it superior, and there would be no guarantee of repeating the trick with a different system. For that reason, this thesis attempts only the 'soft option' of comparing principles with heuristics. However, even this reduced objective suffers from the same limitation, namely that without clear indications of the strengths and weaknesses of both materials, any success for one over the other will be difficult to interpret. While there are strong indications of a common set of core issues (e.g. consistency, feedback, error management), exactly what makes principles better than heuristics (if they are) will need more work than is attempted here. This important issue will be returned to in Chapter 8.

5. Usability and Usability Evaluation Methods

The remainder of this Chapter will be taken up with a necessarily brief discussion of usability and usability evaluation methods, including heuristic evaluation. This aim is to place the principles versus heuristics theme in a wider context, and lead up to some more specific remarks regarding comparisons of evaluation methods. The Chapter concludes with a discussion of the role of cognitive psychology in usability research.

5.1 Usability

As late as 1992, Adler & Winograd wrote that "... the typical human factors effort is given low priority among a design team's objectives. Usability issues are often left to the latest possible date, by which time modifications are too expensive to make." (Adler & Winograd 1992 p4). In 1994, the mean staff utilisation for usability laboratories in 13 companies was 17 per company, with IBM and Microsoft having the largest lab count (Nielsen 1994f). By 1997, however, Microsoft had 76 usability engineers and 18 labs (Williams & McClintock 1997), and when this author left in early 1998 more were being built. In the same year, John Carroll observed that "In industry, HCI practitioners have become well integrated in system development. HCI specialists have moved into a great variety of roles beyond human factors assurance." (Carroll 1997 p506). Even by 1994, according to Rubin (1994), the demand for usable products was outpacing the supply of trained usability professionals. In 2000, this demand produces around 10 jobs per week world-wide[6] for usability engineers and related personnel (HCI RN Jobs Index).

This "explosion of usability" (Rubin 1994 page xvii) has been accompanied by a dramatic growth in research and other publications. A search of the Social Science Citation Index from 1981 to 1990 inclusive (via the BIDS[7] ISI Data Service) on keyword 'usability' (in title,

6 In theory. In practice, most are based in the US.
7 BIDS has since changed to Web of Science (WoS).

authors & journal) returned just 19 entries. The same search for years 1991 to 1994 inclusive returned 91 citations, and for 1995 to July 2000 inclusive there were 246. Books published between 1993 and 2000 include practical guides (Dumas & Redish 1993, Rubin 1994), a 'usability in practice' collection (Wiklund 1994), handbooks (e.g. George 1995) and collections on the politics of usability (Bias & Mayhew 1994, Trenner & Bawa 1998). Usability may even find its way into public access systems (Rowley & Slack 1998).

However, the fact that PCs are now 'usable' does not mean that we use them. The bad news, according to Tom Landauer (Landauer 1995, Nickerson & Landauer 1997) is that the effect of "computer aids" on work efficiency and productivity has been very slight. Though computers have taken over the number-crunching and repetitive processes that humans once did, this first phase is now running out of steam, and in the second phase, in which computers augment rather than take over our everyday tasks, we have failed (Landauer 1995 pp6-7). According to Donald Norman (1998), the solution is to make computers do less, or rather, to match the many separate tasks that we perform by producing not general-purpose devices but separate "information appliances" specialised for particular activities. According to Landauer (1995) and Nickerson & Landauer (1997), the solution is for new applications to be made not just 'usable' but 'useful' and 'socially valuable'; that is, to make computers do more, or rather, do more "sufficiently useful things" (Landauer 1995 p7).

What seems to have happened is that usability (the match between what a system does and how well it can be used) has taken over from utility (the match between what a system does and what its users might want to do with it). This is in spite of utility being incorporated, implicitly or explicitly, into many definitions of usability (Shackel 1984, Gould & Lewis 1985, Nielsen 1990b, Bevan et al. 1991, Nielsen 1993). If usability is a measure of the achievement of specified goals in a specified context (ISO 9241-11 (1998)), then it is 'goal achievement' rather than 'context identification' which has become the focus. The "early and continual focus on users" seems to have been superseded by the third of Gould & Lewis's four design 'principles', namely "early and continual user testing" (Gould 1988). Usability has become task- (or perhaps test-) centred rather than utility-centred.

> "... the usefulness and usability of software have usually not been subjected to the regular and systematic testing of actual utility that underlies improvements in most other technologies. [...] Too often insufficient attention has been paid to measuring how well and in what ways systems do and do not actually help people get work done." (Nickerson & Landauer 1997 p14.)

This thesis will encompass aspects of both usability and utility. The versions of the author's principles set used in Experiments 1 and 2 incorporate 'requirements and functionality' principles not present in many sets including that of Bastien & Scapin. The approach adopted in these two experiments was to have subjects freely explore the whole of the software to be evaluated, rather than perform specific tasks. As we shall see, this enabled reporting of usability (or rather utility) problems related to wider issues. These results will be contrasted with those of Experiment 3, where closely confined tasks (and subsets of the

evaluation materials) were used. The balance between free exploration and set tasks is an issue for this thesis and will be explored in later Chapters.

5.2 Usability Evaluation Methods

Figure 1.9 shows a 2 x 2 taxonomy of usability evaluation methods (UEMs), adapted from Whitefield et al. (1991). This classification scheme is simpler than most others, but involves minimal overlap between categories. The discussion which follows draws on both Whitefield et al. (1991) and Macleod's (1992) elaboration.

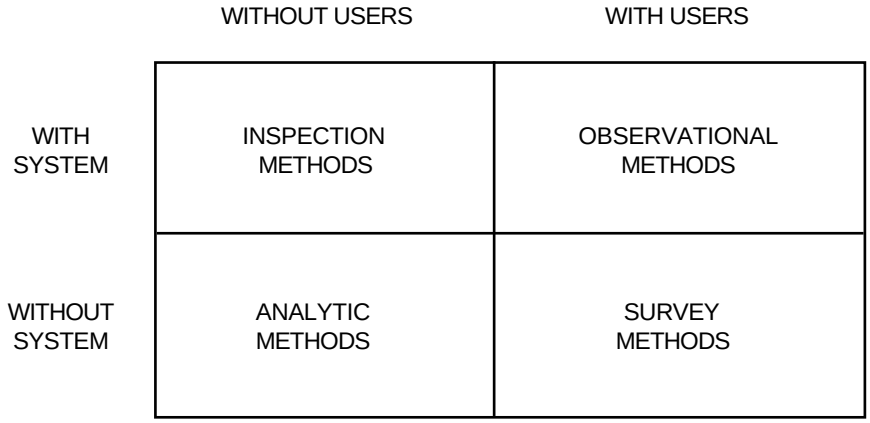

Figure 1.9. A 2x2 taxonomy of usability evaluation methods, adapted from Whitefield et al. (1991). Groupings are "broadly indicative", only. In Whitefield et al. the term 'system' is used to refer collectively to the user and computer; 'inspection methods' and 'survey methods' were dubbed 'specialist reports' and 'user reports', respectively.

On this "broadly indicative" classification, both inspection methods and observational methods are performed using the system under evaluation, whereas analytic and survey methods take place in the absence of the system. The difference between inspection and observation is that inspections are carried out without the presence of users (normally by specialists or designers), whereas observations include end users.

Inspections [with system, without users] are performed on a system, prototype (including mockup) or specification by usability specialists or other personnel, who judge the system against some criteria (Mack & Nielsen 1994, Nielsen 1994e). Judgements and predictions are made about the use of the system by its end-users. In 'guideline reviews' (Mack & Nielsen 1994) and heuristic evaluation (described in the next Section) the evaluative criteria are guidelines, principles or heuristics.

Other inspection methods include cognitive walkthrough, pluralistic walkthrough and error analysis. In cognitive walkthrough (Polson et al. 1992), evaluators work through the interaction process with the user and the user's task in mind. At every step, the interface is assessed in terms of its ability to support the user's goals (using a goal structure), and to guide user actions towards task completion. Pluralistic walkthrough (Bias 1991) involves teams of evaluators (involving users, developers and human factors professionals) together reviewing design screens, each in turn acting as users. Error analysis (e.g. Christie et al.

1995) involves assessment of the range of possible errors which might be associated with each step of in interaction sequence.

Studies which have compared heuristic evaluation with other methods will be summarised below. Chapter 6 will make use of data from an earlier error analysis performed by the author.

Observational methods [with system, with users] are those in which data is collected during user-system interaction. Methods range from automatic monitoring and observation under laboratory conditions ('user testing') to ethnographic approaches. Performance data (either system or user, or both) may be collected. Users may or may not be asked to 'think aloud', that is, vocalise their reactions; other comments may also be elicited. The degree of observer involvement ranges from nil in automatic logging to high in ethnographic approaches. Settings for observational studies include both real and simulated work environments and laboratory set-ups.

Observational methods include 'usability engineering' towards measurable criteria (Tyldesley 1988, Whiteside et al. 1988, Wixon & Wilson 1997), cooperative evaluation (Wright & Monk 1991, Monk et al. 1993), and ethnographic approaches. The latter include contextual design (Wixon et al. 1990) and contextual enquiry (Holtzblatt & Jones 1993), plus participatory design (Schuler & Namioka 1993).

Chapter 4 features an outline of cooperative evaluation and discusses one of the studies on which it is based. Ethnographic approaches (as well as focus groups and interviewing) are germane to the above discussion of utility and context of use, and will be returned to in Chapter 8.

Analytic Methods [without system, without users] are those in which experts or specialists derive predictions about system use from formal or semi-formal models or specifications of the system, its interface, or its users. Targets of such predictions include user errors, task time, user goals and task models.

Analytic methods include GOMS (Card et al. 1983) and the Keystroke Level Model (KLM) (Card et al. 1980) analysis, Cognitive Complexity Theory (CCT) (Kieras & Polson 1985), and Task Action Grammar (TAG) (Payne & Green 1986).

Survey methods [without system, with users] are those in which data is collected from users concerning system use, (mainly) away from the system itself. Data can include users' attitudes, opinions and problems concerning both the system and its wider context.

Survey methods include questionnaires, interviews and focus groups.

The two main themes of this thesis are concerned with the distinction between inspection methods and observational methods, as depicted above. The first theme (Chapters 2 to 5)

explores the use of one form of inspection method, namely guideline review, to identify usability (and utility) problems. The reviews used in this thesis make use of principles and heuristics, and most of the procedures prescribed for heuristic evaluation. The second theme of the thesis (Chapter 4 onwards) concerns the number of evaluators required to identify the majority of usability problems. In Chapter 5 onwards, this will be further explored in relation to problems occurring under one form of observational method, namely 'think-aloud' user testing.

This contrast between predicted (as opposed to merely reported) and observed problems will be characterised in terms of Gray & Salzman's (1998) broad distinction between analytic and empirical methods. **Analytic** methods attempt to make predictions about user-system interaction from intrinsic system features (including mockups or prototypes), while **empirical** methods measure some aspect(s) of interaction while a (full or partial) system is in use. On this classification, both inspection methods and (for example) GOMS are analytic while observational methods are empirical. In this thesis, the term 'analytic' will be used in this wider sense, that is, to encompass the predictive nature of inspection methods such as principle- or heuristic- based evaluation.

5.3 Heuristic Evaluation

Though in the author's view it is not essential that the declared methodology of heuristic evaluation (or any other method) be followed in order to compare evaluation materials, it will be useful to set out what is claimed for it. In addition, some of Nielsen's recommendations for heuristic evaluation, particularly those concerning evaluator expertise, are relevant to the second theme of the thesis.

In heuristic evaluation (principally Nielsen & Molich 1990, Nielsen 1993, Nielsen 1994d) an evaluator systematically inspects an interface to assess compliance with a set of heuristics such as those proposed by Molich & Nielsen (1990, described above), plus the evaluator's "general knowledge of usability principles" (Nielsen 1994d p61). It is recommended that the evaluator goes through the interface twice, focusing first on its "flow" (Nielsen 1993 p158) and then on its "individual dialogue elements" (Nielsen 1994d p61). Evaluators will usually be usability specialists or, better, "double specialists" (Nielsen 1992), but heuristic evaluation can be performed by "people with little or no usability expertise" (Nielsen 1993 p162). However, it is not recommended to use end users as evaluators, though they can and should be used in user tests (Nielsen 1994d).

Since different evaluators will identify different usability problems, and a single evaluator will miss most problems, it is recommended that the findings from about 3 to 5 evaluators be aggregated together, the exact number depending on a cost-benefit analysis (Nielsen & Molich 1990, Nielsen 1993, Nielsen 1994d). However, each evaluator should inspect the interface alone, and should not be allowed to communicate until after all sessions are

complete (perhaps using a debriefing session). Results can be recorded either as written reports by each evaluator, or by having evaluators "vocalise" to an observer-recorder as they proceed. If an observer is used, he or she can assist the evaluator in case of problems, for example if the evaluator has limited domain experience (Nielsen 1993). However, in such a case the responsibility for analysing the interface still rests with the evaluator: the observer "does not need to interpret the user's actions", and the evaluator should not be given help until clearly in trouble and has commented on the problem (Nielsen 1994d pp157-158).

One drawback with heuristic evaluation is that it tends to find more minor usability problems than major problems (Nielsen 1994d). For this reason, some means of prioritising is necessary. One solution is to have evaluators assign severity ratings. According to Nielsen, the severity of a usability problem is based on a combination of three factors, namely frequency [of occurrence], impact [on the user's subsequent use of the system], and persistence [of the problem subsequent to users first encountering it], plus a fourth, namely "market impact" (Nielsen 1994d p47). Severity should be assigned by each evaluator independently after the evaluation sessions, e.g. by questionnaire. The reliability of individual severity judgements is very low, so mean ratings from three or four evaluators should be used (Nielsen 1994d). Problems judged to be more severe tend to be found by a higher proportion of evaluators than those than given low severity (paraphrasing Nielsen 1994d p49).

Heuristic evaluation is explicitly intended as a "discount usability engineering" method (Nielsen 1989, Nielsen 1990a, Nielsen 1993, Nielsen 1994c). Discount usability engineering is based on three techniques, namely scenarios, simplified thinking aloud and heuristic evaluation, plus "early focus on users" (Nielsen 1993 p17). Scenarios reduce both functionality and interface features to a minimum. Simplified thinking aloud involves having one test user at a time perform set tasks while 'thinking out loud'. Heuristic evaluation can be used on prototypes (including paper mockups), thus being suitable for use early in the usability engineering lifecycle. The discount method is not intended to be the "best" method or to give "perfect" results, but it is claimed to identify most of the usability problems which can be found by more expensive methods at a reduced cost (Nielsen 1989, Nielsen 1993, Nielsen 1994c). Since heuristic evaluation and user testing tend to identify different sets of problems, it is recommended that testing be used to 'pick up' any problems not found by heuristic evaluation (Nielsen 1994d).

Experiments 1 of 3 of this thesis made use of most of the above recommendations for heuristic evaluation. Both novices (mainly psychology undergraduates) and experienced subjects (HCI researchers) acted as independent evaluators. The software and tasks were chosen to be either within the subjects' experience and acquaintance (public access system, word processor, spreadsheet) or directly relevant to their (alleged) interests

(psychology teaching package). In Experiment 1 the software evaluated was depicted as an "early prototype". In each experiment subjects were instructed to first run through the whole interface and then to proceed in more detail. Subjects were to 'think aloud', using the evaluation material as a guide, but also to identify any issues of their own (unlike in Bastien & Scapin's studies). In Experiment 2, subjects were specifically asked to consider remedies and other recommendations related to the problems identified. In Experiments 1 and 3 the experimenter acted as observer and recorder, assisting and prompting only if necessary and taking great care not to direct or influence the subject's comments. (In Experiment 2, subjects entered their comments and reactions online, into a prepared spreadsheet.)

Exceptions to the above recommendations were that reduced-scope scenarios ('closed' tasks) were used only in Experiment 3. (The issue of task scope will be taken up in later Chapters.) Severity ratings were assigned by each subject at the end of each session, in an individual 'debriefing' with the experimenter. In assigning severity, subjects were asked to consider only the impact of each problem (from 'trivial' to 'serious'). Two forms of problem frequency, namely frequency of prediction and (in Experiment 3) frequency of occurrence, were derived, but both were computed from the subject data. Persistence was not assessed (nor 'market impact'). The issue of severity judgement will also be taken up in later Chapters.

While it is clear that evaluator expertise plays a strong role in heuristic evaluation, it is not clear to what extent heuristics rely on background experience over substantive content (see Section 3.4). For this reason, Experiments 1 and 2 made use of a control condition in which subjects were not provided with either heuristics or principles. Experiment 1 also compared the ability of novices and experienced subjects to identify usability problems. It was hoped that the novices would provide a 'base line' against which evaluator performance might be measured.

The "discount" nature of heuristic evaluation, specifically the claim that only 3 to 5 evaluators can together find most usability problems with an interface, is the second major theme of the thesis. This and the closely related issue of problem identification will be taken up in full in Chapter 4 and explored in the succeeding Chapters.

5.4 Comparative Usability Evaluation

5.4.1 Summary of Related Research

There have been a large number of studies comparing different usability evaluation methods (UEMs). The following summary focuses on those which have compared heuristic evaluation with other methods, those which have contrasted inspection methods (including heuristic evaluation or guideline review) with user testing, and those which have compared predicted (analytic) and observed (empirical) usability problems. Some studies fall into more than one category.

Studies which have compared heuristic evaluation with other methods are in general agreement that heuristic evaluation is good at finding a wide spread of general usability problems (Virzi et al. 1993, Cuomo & Bowen 1994) at comparatively low cost (Jeffries et al. 1991, Nielsen & Phillips 1993). Heuristic evaluation and a 'think-aloud' test were roughly equivalent in finding core problems (Virzi et al. 1993), and a walkthrough without heuristics found more minor problems than heuristic evaluation (Kelley & Allender 1995). However, other methods such as cognitive walkthrough may be necessary in order to focus on specific, task-related problems or re-design issues (Cuomo & Bowen 1994, Desurvire 1994, Dutt et al. 1994).

There is also general agreement that inspection methods (including heuristic evaluation) and user testing identify usability issues of different sorts and scope (Bailey et al. 1992, Karat 1994, Desurvire 1994, Karat 1997). According to Karat (1994, p221), inspection methods are appropriate for identifying "numerous low-level design trade-offs", while user testing is suited for "high-level design guidance" [and] "full coverage of [the] interface". The two approaches may also be best suited to different stages of the development cycle, with inspections most effective in the earlier stages (Jeffries & Desurvire 1992, Karat et al. 1992, Desurvire 1994, Karat 1997). The approaches are thus considered to be complementary rather than overlapping in yielding different results (Jeffries & Desurvire 1992, Karat et al. 1992, Karat 1994). However, the precise nature of this difference needs to be better understood (Karat 1997). As far as heuristic evaluation is concerned, Jeffries & Desurvire (1992, p41) caution that the method's advantages ("it's fast, it's cheap, it finds lots of problems) should be weighed against its disadvantages ("it requires multiple evaluators, it works best with experts, it finds a distressing number of minor problems") and the advantages of user testing ("it overwhelmingly finds severe problems, it finds problems that impact real users").

There is less agreement on the proportion of empirical problems that can be successfully predicted by analytic methods. Desurvire et al. (1992) found that (at "only" 44%), experts using heuristic evaluation correctly predicted more test problems than cognitive walkthrough, while 55% of non-experts' predictions "could not occur". Sears (1997) also found that heuristic evaluation correctly predicted a consistently higher proportion (mean 47%) of observed problems than cognitive walkthrough (24%), but that heuristic evaluation identified many more minor problems and problems not observed[8]. Cuomo & Bowen (1994) found that (at 58%) cognitive walkthrough predicted more problem types than either heuristic evaluation (46%) or the Smith & Mosier (1986) guidelines (22%). In this study the guidelines and heuristics also elicited more problem types which were not experienced or detectable in the usability test. In a case study of cognitive walkthrough and five other

[8] Sears (1997) showed that a hybrid of the two methods (and a third from Karat et al. 1992), dubbed "heuristic walkthrough", performed about the same as heuristic evaluation on minor problems, but out-performed both methods on serious problems.

methods including heuristic evaluation, John & Mashyna (1997) found that just 5% of observed problems had been precisely predicted and another 5% vaguely predicted, with 27% false alarms, by one evaluator using cognitive walkthrough. Of the total predicted problems from this study, John & Marks (1997) reported that 52% were not observed, and the changes made as a result of those which were observed resulted in just 23% fewer problems (and 11% new problems). In an earlier study, Bailey et al. (1992) estimated the ratio of "potential" to "real" problems found by heuristic evaluation at 10:1.

5.4.2 Some Conclusions from Related Research

What can be concluded with reasonable certainty from the above summary is that inspection methods elicit a more divergent range of usability problems than user testing, and that the two approaches may be considered complementary. Whether or not walkthroughs can replace empirical tests, heuristic evaluation and other guideline-based techniques may best be used in the early stages of the development cycle. However, predictions thus made are unlikely to match well with those later observed, with as many as half turning out to be 'false alarms'. The predictive power of heuristic evaluation seems to be higher than that of cognitive walkthrough, though only two of three studies agree on this. There is more agreement that cognitive walkthrough and user testing are more alike in focusing on defined tasks and scenarios. In any case, it appears that the effectiveness of inspection methods in diagnosing and addressing real problems remains to be established.

While the above view of inspections vis-a-vis user testing is broadly in line with Nielsen's appraisal (see Section 5.3), any over-prediction inherent in heuristic evaluation would detract from its role as a 'discount' technique. Even if most of the 'false alarms' are taken up with minor and tangential issues, the claim for a discounted approach is that it enables a high proportion of real problems (such as those found in user testing) to be accounted for by a small number of suitably experienced evaluators. The second main theme of the thesis examines the validity of this claim for heuristic evaluation. As has been said, this author believes that it is not necessary to follow the dictates of any one method in order to compare evaluation materials. Part of the difficulty in disentangling the results of UEM studies is that (with the notable exception of cognitive walkthrough) the methods are rarely described sufficiently well, and offer so much potential for combination and variation, that findings can appear contradictory. Though many aspects of heuristic evaluation have been accounted for, the central procedural issues - *precisely* how the heuristics are to be used to elicit usability problems, how problems are recorded and counted as separate issues - remain open to interpretation.

The role of evaluator expertise in heuristic evaluation, in particular Nielsen's (1992) use of "double specialists" (those with both domain and evaluation expertise), appears not to have been confirmed. Of the above studies, only two (Desurvire et al. 1991, Desurvire et al. 1992) set out to compare the effects of experience level on problem identification. As has

been pointed out, Experiment 1 of this thesis uses both experienced evaluators (HCI researchers) and novices (mainly psychology undergraduates). However, Experiments 2 and 3 used only (different) novices from the same pool. Evaluator expertise is an issue for this thesis and will be returned to in Chapter 8.

Gray & Salzman (1998) have criticised the comparative UEM literature in general, and five studies (Jeffries et al. 1991, Karat et al. 1992, Nielsen 1992, Desurvire et al. 1992, Nielsen & Phillips 1993) in particular, for lacking both internal and external validity. According to these authors, the five studies lack internal validity in relying on small sample sizes and one-off measurements ('statistical conclusions validity'), and, in particular, by not obtaining independent assessments of problem categories and severity ratings. They lack external validity by being insufficiently explicit about what method variants or software characteristics were used ('causal construct validity'), and by attempting to generalise beyond the scope of particular results or conclusions. Gray & Salzman specifically criticise the use of problem count as a measure of the effectiveness of a UEM, recommending that researchers limit both their expectations and their claims for UEM studies. One way would be to focus on the predictive power of analytic methods by distinguishing between correct predictions ('hits'), over-predictions ('false alarms' or false positives), misses, and 'correct rejections'. Another way would be to distinguish clearly between problem tokens (separate instances of usability issues) and problem categories. A third would be to employ multiple (convergent) means of usability assessment.

The approach taken in this thesis addresses many of Gray & Salzman's criticisms and recommendations. Subject sample sizes vary from 7 to 9 per cell, with full reporting of statistical analyses. Care has been taken to be explicit about procedures used and the scope of evaluation materials. Experiments 2 and 3 offer (albeit limited) inter-rater reliability data for their problem count totals (but not severity ratings), allied to a model of the problem reduction process to be introduced in Chapter 4. Experiment 3 uses Gray & Salzman's distinction between hits, false positives and misses to differentiate between measures of predictive power. Chapters 4 and 6 demonstrate the effect of problem categorisation on problem count. Chapter 7 presents some (mainly qualitative) evidence for the use of principles in usability assessment.

This author agrees with Gray & Salzman that problem count is an impoverished measure of the effectiveness of a UEM, let alone competing guidelines and heuristics. Given that it *is* the main dependent variable in these experiments, the author believes that the establishment of some internal validity for this measure is more important than whether or not the procedures used qualify as heuristic evaluation. However, in that these procedures *do* closely resemble Nielsen's prescription (Section 5.3), it is believed that conclusions concerning that particular method can be drawn with more confidence from the experiments to be reported than from studies whose interpretation of heuristic evaluation is less explicit.

5.5 Cognitive Psychology and Usability

As stated earlier, this thesis starts from the view that it is possible to identify a common set of principles which underlie a range of interface families, and that the rationale behind these principles can most usefully be underpinned by cognitive psychology. The author's principles set is drawn from a wide range of sources (listed in Figure 1.5), but in particular the collection of guidelines in Marshall et al. (1987). The origins of this latter set lie in turn in the many principles from cognitive psychology described in Gardiner & Christie (1987). The author's view is that by structuring each principle or sub-principle in the manner illustrated in Figure 1.4, it is possible to avoid the pitfalls of many guideline collections. It is also possible to make explicit the rationale for each principle by including the source(s) from which it is derived. In the case of Marshall et al. (1987), those sources include the psychology of thinking and mental models, of memory, of skill acquisition and of language (in Gardiner & Christie 1987). Marshall et al. put it thus:

> "What is different about the guidelines presented here is that they are built directly, in a structured manner, from the theoretical bases which had been largely untapped until now. Very few of the design guidelines which are currently available have been derived directly from research on high-level cognitive processes, along the lines described here. In previous chapters [of Gardiner & Christie 1987], psychological theory was distilled into broad 'principles' or summaries. These principles are now taken through a process of simplification and turned into guidelines which are as jargon-free and explicit as possible. [...] The aim is to demonstrate an approach which may prove increasingly useful, as our understanding of human cognitive psychology continues to broaden and deepen, rather than to provide a tool which is fully tested and ready to go." (Marshall et al. 1987 p221).

Though the project began by these authors remains incomplete (not all of the 14 "sensitive dimensions" are developed as fully as the example in Figure 1.1), and (as the authors say) the guidelines may require updating in the light of new research, the ideal - of generating design and evaluation principles from empirical findings - remains valid. The issue, then, is whether HCI research in general, and cognitive psychology in particular, is up to the job.

Some practitioners do not think so. Views range from the 'strong' position - that there is *nothing* in theories of human behaviour with sufficient generality and detail at the cognitive level which can serve as a basis for HCI (Landauer 1991, thesis author's italics), to the 'weak' - that the role of psychology in HCI has proved difficult to articulate in sufficient detail and to verify in practice (Carroll 1991). Yet the discipline of cognitive ergonomics (cognitive engineering in the US) exists to articulate just such a role for psychology:

> Cognitive ergonomics is concerned with the mental aspects of the [human-computer] interaction and so with developing specifications of the knowledge required by the human to interact with the computer to perform work effectively. [...] To support the specification of knowledge required by humans to use computers, the discipline of cognitive ergonomics itself needs to acquire and apply principles concerning that knowledge. (Long 1989, p5).

So what has gone wrong? Can cognitive ergonomics have had so little impact that the principles and models in such books as Long & Whitefield (1989) and Wærn (1989) have

nothing to offer to our understanding of the "mental aspects of the interaction"? Can cognitive psychology have *nothing* to contribute to usability?

Barnard (1987) accounted for the 'weak' view in terms of mappings between cognitive theory and real-world user-system interactions. On this view, there is no necessary match between theories developed in the laboratory and real user performance or the complex choices involved in system design, without showing that the former apply unequivocally and uniquely to specific aspects of the latter. But while it is not possible to solve each and every design issue by formal experiment, even complex models such as GOMS (Card et al. 1983) and Norman's theory of action (Norman 1986) have the capacity to illuminate specific aspects of user-system interaction. This author's interpretation of Barnard's view is that in 1987 such capacity was not (yet) sufficiently apparent. Yet all of the analytic methods listed in Section 5.2 (GOMS, KLM, CCT and TAG) predate 1987. Since then there have been extensions to GOMS such as NGOMSL (Kieras 1997), elaborations of cognitive walkthrough (e.g. Sears 1997, Sears & Hess 1999), and empirical tests of action theory (e.g. Lim et al 1996).

It appears, then, that at least *some* explanatory power for models based on principles concerning "the knowledge required by humans to use computers" has been and is being established. It does not, therefore, seem too much to expect that the cognitive principles in such collections as Marshall et al. (1987) may prove to have similar potential when expressed in guidelines of the form illustrated earlier in the Chapter. That, in essence, is the first theme of this thesis.

Summary of Chapter 1

1. This Chapter introduced the two main themes of the thesis and attempted to place them in the context of research comparing both usability guidelines and usability evaluation methods.

2. The author's principles set was introduced and contrasted with Nielsen's heuristics.

3. Heuristic evaluation was summarised and set in the context of the theme of principles versus heuristics.

4. Summaries were offered of both usability evaluation methods and the role of cognitive psychology in usability.

5. Several issues were flagged and positioned for later Chapters. They include: comparison of principles and heuristics rather than of principles sets; usability problem identification and 'discounting'; free exploration vs. set tasks; usability testing vs. inspection; usability problem severity judgement; usability problem identification and reduction; the role of evaluator expertise in heuristic evaluation; ethnographic approaches and usability vs. utility.