ELSEVIER

Interacting with Computers 12 (1999) 81–118

Interacting with Computers

www.elsevier.com/locate/intcom

Editorial

Development milestones towards a tool for working with guidelines

J. Vanderdonckt[1]

Université catholique de Louvain, Institut d'Administration et de Gestion, Place des Doyens, 1-B-1348 Louvain-la-Neuve, Belgium

Abstract

Several tools for working with guidelines already exist, both as commercial products as well as within research and development. As these tools frequently manipulate guidelines during many development steps of a user interface of an interactive application, they can overthrow any approach followed to develop this application. They also raise the fundamental question of to what extent can we trust these tools. To answer this question, we introduce five development milestones through which we must pass to produce a high quality tool for working with guidelines:

1. An initial unstructured but comprehensive set of guidelines is formed by collecting, gathering, merging, compiling guidelines from all available world-wide ergonomic sources.
2. The initial set is sorted and classified within a single organising framework.
3. A methodology, paying particular attention to finding and applying relevant guidelines is developed for grounding interactive applications on the organised set of guidelines.
4. The structured guidelines and the supporting methodology are given computational representations for manipulation by computer-based tools.
5. The methodology developed in (3) is further modified to optimise the effectiveness of computer-assisted user interface design.

In this paper, we define these milestones and their associated goals, specify a general procedure and discuss some problems raised at each milestone. We then deliver an analytic synthesis of various experiences acquired to solve these problems and we discuss the validity of these experiences from the point of view of completeness, consistency and correctness. From these experiences, we finally draw some lessons useful for any future usage and development of a tool for working with guidelines.

Keywords: Computer-aided design of user interfaces; Development methodology; Ergonomic algorithm; Evaluation; Guidelines; Human–computer interaction; Interactive application; Software development methodologies; Software engineering; Software ergonomics; Standard; Style guide; Tools for working with guidelines; Usability; User interface

[1] Editor-in-Chief of this special issue on "Tools for Working with Guidelines"; Tel.: + 32-10/47-85-25; fax: + 32-10/47-83-24; e-mail: vanderdoncktj@acm.org, vanderdonckt@qant.ucl.ac.be; WWW: http://www.qant.ucl.ac.be/membres/jv/jv.html.

PII: S0953-5438(99)00019-3

1. Introduction

Ensuring the ergonomic quality of user interfaces (UIs) has become a major concern when developing interactive applications for over a decade. To characterise this quality, utility and usability are often distinguished. Utility translates the appropriateness of a UI with respect to the functional goals of an interactive application; in this way, it remains user independent. In contrast, usability relates to the appropriateness of this UI with respect to the operational user's goals; it therefore heavily depends on the user. The UI of a word processor could be considered as useful in providing a facility for computer-aided creation of a table of contents of a given document. The usability of this UI will depend on the easiness with which a user will be able to carry out this task, without being blocked by an error, but being supported to fix it.

The suitability of interactive applications which are functionally rich (thus, useful), but operationally poor (thus, unusable) still gives rise to dilemmas:

- intensive, experienced users tend to prefer a useful interactive application, even when it is not usable rather than a usable interactive application which lacks utility;
- casual or novice users tend to reject an interactive application as soon as it is unusable, even if it is very useful; they prefer an application which is a little useful, but usable.

To study, express and ensure the usability of a UI, several disciplines can help every person who is responsible for developing the UI, notably participatory design, cognitive psychology, contextual enquiry, software ergonomics. The latter has already developed methods that have proved their positive impact on the usability of a UI: evaluation methods with or without users [1], heuristic evaluation [2], usability laboratory tests as well as any ergonomic approach grounded on guidelines [3].

We here define a *guideline* by a design and/or evaluation principle to be observed in order to get and/or guarantee the usability of a UI for a given interactive task to be carried out by a given user population in a given context [4]. In the rest of this text, we assume a general method for including guidelines throughout the development life cycle of a UI for a particular interactive application. Guidelines are mostly used during

- the specification phase: a set of guidelines is delimited as requirements for the future UI;
- the design phase: guidelines are exploited in order to decide an appropriate value for each design option by considering the context (which includes the interactive task, the user population and the working environment in which the users are carrying out their task);
- the prototyping phase: guidelines are exploited to obtain as soon as possible a static or working UI prototype that can be showed, tested and evaluated; in particular, techniques for automated UI generation and for computer-aided design of user interfaces (CADUI) [5] can be used;
- the programming phase: guidelines are gathered to guide, orient, decide, ensure a UI development within the developing environment for the targeted computing platform;
- the evaluation phase: the resulting UI is evaluated with respect to guidelines that are often ones that have been selected in previous phases;
- the documentation and certification phase: guidelines which have been manipulated in

previous phases are instructed for documenting an interactive application for communication, reuse, maintenance or commercial promotion purposes; this phase typically consists in exposing the UI towards these guidelines.

Aside from these situations, guidelines also claimed a new territory, that of teaching UI design based on guidelines throughout the life cycle. This editorial introduces a special issue devoted to tools for working with guidelines and underlying activities: searching for guidelines, collecting, working, writing, propagating, using, applying, verifying and, more recently, teaching and learning. These activities are progressively better understood and structured into methods; they are today manipulated by people who often cumulate several roles: project leader, methodologist, designer, analyst, developer, evaluator, human factors specialist, psychologist, teacher, computer scientist.

To support these people in their activities, some research and development has been commercialised since 1989 (with NaviText SAM® [6]), resulting in a wide range of tools for working with guidelines. From this commercialisation, a profound change emerged in the current practice of using guidelines as observed in several questions mainly raised at three levels.

1. At the level of the source of guidelines:
   - To what extend can we trust guidelines based on an ergonomic approach?
   - Where and how guidelines can be applied in existing methodologies (i.e. the above phases) for developing interactive applications?
   - What are the exact roles played by these people who manipulate guidelines during development phases and how do they interact within the existing structures of a development team?
2. At the level of computer tools, where the above questions must be reframed to address software issues:
   - To what extent can we trust guidelines incorporated into software tools? More particularly, if the confidence level of a guideline is considered acceptable at the beginning, what is the resulting confidence level after transformation?
   - Where guidelines can be located in existing methodologies for developing interactive applications? How are these locations redefined after their usage in a software tool? To what extend could we delegate a portion of the activities to these software tools and how?
   - What are the exact roles played by people using software tools for working with guidelines and how do they participate within a development team? More critically, if we consider a person who recently learned software ergonomics and using such tools, is this person able to assume the same roles of a person who acquired extensive experience in human factors, but not using such tools? For instance, could a human factors expert be replaced by an inexperienced designer assisted by software tools? Could an expert also be assisted by tools?
3. At the level of the development of software tools:
   - How can we manage to develop an efficient and valid software tool?

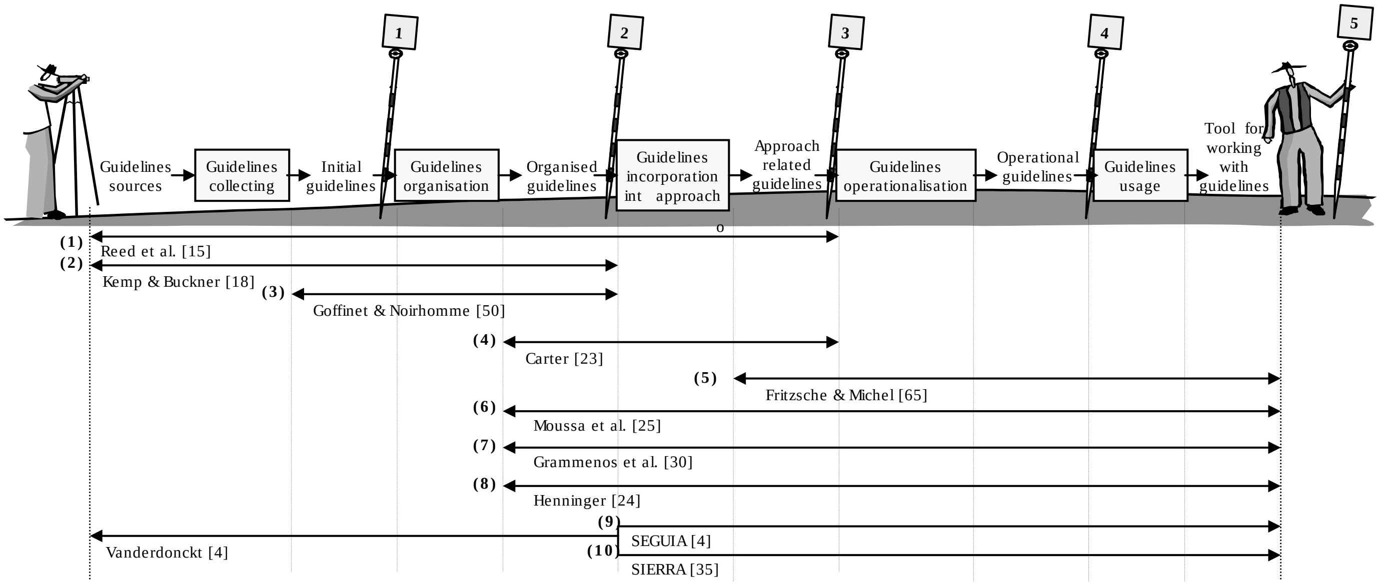


Fig. 1. Development milestones towards a tool for working with guidelines.

- On which guidelines can we ground this tool and how can they be incorporated into software?
- What are both negative and positive consequences of this transformation at the different steps of this transformation?

Attempting to answer these questions transforms the research agenda for computer-based guidelines tools. Although we cannot answer them now (and may never be able to in full), they do expose the critical steps in moving from manual to computer-supported use of guidelines. For this purpose, we introduce five development milestones through which such a tool should go (Fig. 1). A milestone is placed after each transformation step:

1. Completion of the collection, gathering, merging, and compilation of guidelines coming from world-wide available ergonomic sources to form an initial unstructured but comprehensive set of guidelines (milestone #1).
2. Completion of sorting, classifying and integrating these initial guidelines within an organisational structure (milestone #2).
3. The incorporation of organised guidelines into a methodology that supports precise location and circumscription of a subset of relevant guidelines and their points of application, that will result in an interactive application grounded on sound ergonomic advice (milestone #3).
4. Automation and digital representation of the guidelines and their use within a computer-support methodology (milestone #4).
5. Refinement of the methodology in (3) to optimise the benefits of computer-tool support (milestone #5).

Fig. 1 shows the relation of papers in this special issue to the five milestones introduced in this editorial. The papers for lines 1, 2, 3, 4 and 9 will form the first part, with papers for lines 5–8 will form the second part.

On reaching the last milestone, we would have completed the transformation of the current incomplete, unsystematic and unreliable use of guidelines into a comprehensive and systematic approach with known levels of reliability in use. To reach this point, we need to better understand each of the steps between the five milestones. In this editorial, we consider four key aspects of each step.

- *Definition and goals:* we summarise the transformation between milestones in terms of its main goals when transforming input resources into output results.
- *Procedure:* we indicate the concrete actions that must be carried out to achieve the goals associated with the step.
- *Problems and analysis:* we report known problems raised by applying this procedure; we then report different solutions and perspectives that significantly contributed to available knowledge for accomplishing the transformation; we evaluate them and identify complementarity and inter-relations. An interdisciplinary synthesis is then proposed by:
  - first, detailing the results of our own experience in developing two such software tools, i.e. SEGUIA and SIERRA (line 9 in Fig. 1);
  - next, highlighting other perspectives provided by contributors to this special issue

and who have been retained for the variety of their work, their originality and their soundness (lines 1–8 in Fig. 1);
  - lastly, summarising these perspectives via an analytic comparison.

- *Validation:* we perform some step validation to check its results and their conformity with the previous transformation steps. Three properties will be assessed: completeness, consistency, and correctness of the results with respect to the input resources. If they seem complete, consistent and correct to solve a specific usability issue, how are the properties preserved or not by the transformation step?

Completeness should let us check that all step results still hold the desired properties. For instance, are guidelines resulting from one step still necessary and/or sufficient to solve a specific problem? *Consistency* should let us verify whether the step results are internally contradictory or contradictory with results provided by previous steps. *Correctness* should let us verify whether the step results are not subject to intrinsic errors and bias.

These completeness, consistency and correctness will generally be verified a posteriori. Where problems are apparent, we will propose some ways to ensure an internal validity (i.e. do the results achieve the goals of the transformation step) and also external validity (i.e. do the step results produce acceptable inputs for subsequent transformations?). For instance, are the guidelines resulting from each step still applicable outside the context in which they have been selected? Could they be generalised?

2. Guidelines collecting

2.1. Definition and goals of guidelines collecting

Everyone who want to apply knowledge recorded as guidelines must start by collecting guidelines. Their goal is to gather a useful subset of guidelines suitable for guiding a UI development.

2.2. Procedure of guidelines collecting

A wide range of literature should be systematically investigated to find sources of guidelines. This investigation consequently results in a partial selection of ergonomic sources according to certain criteria (for example, with respect to the exposed problem, to a particular design option to be decided, to a target computing platform). Next, a useful subset of guidelines is extracted to match the collector’s goals and to satisfy some other criteria (for example, a restriction to experimentally verified guidelines).

2.3. Problems and analysis of guidelines collecting

An enormous quantity of researchers’ papers, internal reports, manuals has been published on guidelines. These are unfortunately widespread in both time and space, as well as in both format and access. This access to guidelines remains mostly difficult, even for skilled, experienced persons.

2.3.1. Our experience in guidelines collecting

2.3.1.1. Method. Since guidelines contained in ergonomic sources come in different guises (for example, design principles, heuristics, golden rules, maxims, commands, state rules, guidelines, recommendations), to differentiate them seems opportune. Five types of ergonomic sources can be identified [7–10].

1. *Design rules:* they comprise a set of functional and/or operational specifications that specify the design of a particular UI. These specifications are presented in a form that requires no further interpretation, either from designers or from developers. They are typically presented as physical rules, screen formats and window templates. Their straightforward format allows an immediate exploitation.
2. *Guidelines sets:* they comprise several prescriptions written for a wide range of UIs. Each prescription is presented as a statement, sometimes along with examples, with or without clarifying explanations and comments. Each prescription generally results from a human consensus between guidelines users. This consensus is less relevant once a prescription is experimentally tested and verified. These prescriptions can become the subject of paper published in dedicated conference proceedings (for example, [11]), a scientific journal. Alternatively, they can be embodied in a larger document typically called a *guidelines set* or *ergonomic guide*. Papers contain prescriptions on a specific topic (for example, the user preference for an interaction technique for acquiring a graphical choice, the colour usage in a search task), whereas *ergonomic guides* try to be more exhaustive (for example, they cover a wider spectrum of possible interactions between users and UIs).
3. *Standards:* they comprise a set of functional and/or operational specifications intended to standardise UI design. Standards are promulgated by national or international organisations for standardisation. They can be military, governmental, civil or industrial. These organisations are numerous: Association Française pour la NORmalisation (AFNOR) in France, British Standardisation Institute (BSI) in UK, Comité Belge pour la Normalisation (CBN) in Belgium, Human Factors and Ergonomics Society (HFES) in the USA, American Department of Defence (DoD), International Standard Organisation (ISO) located in Europe. Standards should aim at generalisation. One of their advantages is their ability to generalise over computing platforms and development environments which each have their own idiosyncrasies. They help to define a minimal threshold for usability below which a UI cannot drop.
4. *Style guides:* their definitions are diverse. A style guide is a rule compendium, with recommendations for designing a UI satisfying a given specification. The style guide is known as a popular tool for defining UI presentation and dialogue, for ensuring its conformity with the corporate specifications and for reaching design consistency across a same platform or a same products line [12]. Such definitions emphasise the consistency aspect of style guides, but fail to ground the style specification (is it technical, organisational, or computer-based?). A style guide forms a high-level design document to consistently describe design options of a common UI, rules, guidelines and conventions related to a specific interactive application or a family of such applications. It should serve as a starting base, as a contract between designers and developers [13].

Although this definition appears to be more interesting, it fails to mention the target application family that style guides tend to address. A *style guide* in this sense comprises a set of guidelines and/or functional or non-functional specifications aiming at consistency for a family of distinct UIs. This family can be based on an operating system (such as Windows'95), on a software editor (such as Borland's products), by a particular physical environment (such as NeXTStep), by a domain of human activity (such as telecommunications) or by a corporate (in-house style guide). Large corporate environments have committed themselves to the deployment of their own style guide, called *custom style guide* in contrast to *general style guides*.

5. *Ergonomic algorithms:* these aim to systematise one UI design aspect by building it in the same way, according to a same base of design rules. They much more frequently appear as a software component that implements an algorithm rather than a paper procedure.

To report on these five types of ergonomic sources, the existing literature was browsed as exhaustively as possible in considering sources published before June 1994 and in reporting them with the following format:

- *Title:* this is the full title of the ergonomic source.
- *Type of ergonomic source:* one of: design rules, sets of guidelines, standards, style guides, ergonomic algorithms.
- *Bibliographic references:* this is the single or multiple bibliographic references as several references may form the source or when the same source went through several revisions.
- *Contents description:* this is a summary of the source structure and contents.
- *Organisation:* this is the name of the organisation which sponsored the source and its diffusion; it could be civil, official, governmental, military, private or public.
- *Computing platform:* this is the list of computing platforms (for example, Amiga, Macintosh, MS-DOS, Windows V3.11, Windows'95, OS/2, Open Look, Open Windows) for which the source is intended; the source is said *indefinite* when no computing platforms are specified or *multiple* when several computing platforms are concerned.
- *Discipline:* this is the main human discipline from which the source originates.
- *Amount of guidelines:* this is the total count of guidelines catalogued in the source, in so far as separate guidelines can be identified.
- *Amount of pages:* the page count of the source.
- *Amount of bibliographic references:* this is the total amount of references found in the source bibliography which are (in)directly related to guidelines; this information provides a first impression of the extent to which the source is relying on earlier results.
- *Amount of citations:* this is the total amount of quoted bibliographic references.
- *Experimental citations:* this summarily characterises the total amount if which the source quotes experimental results according to a “none/some/several/many” scale or according to a metric whenever possible.
- *Guidelines examples:* this provides one or several representative examples of guidelines as discovered in the source.

- *Example types:* an example is defined as *partial* if it only depicts one guideline at a time or as *total* if several guidelines are exemplified together; the amount of example is evaluated according to a "none/some/several/many" scale; the example style is qualified as *good* and/or *bad* as examples of good or bad guideline applications are provided.
- *User interface types:* this specifies all UI types covered by the source: character-based user interface (CUI), graphic user interface (GUI), vocal user interface (VUI), tactile, stylus, virtual.
- *Source discussion:* this evaluate the source impact and utility.

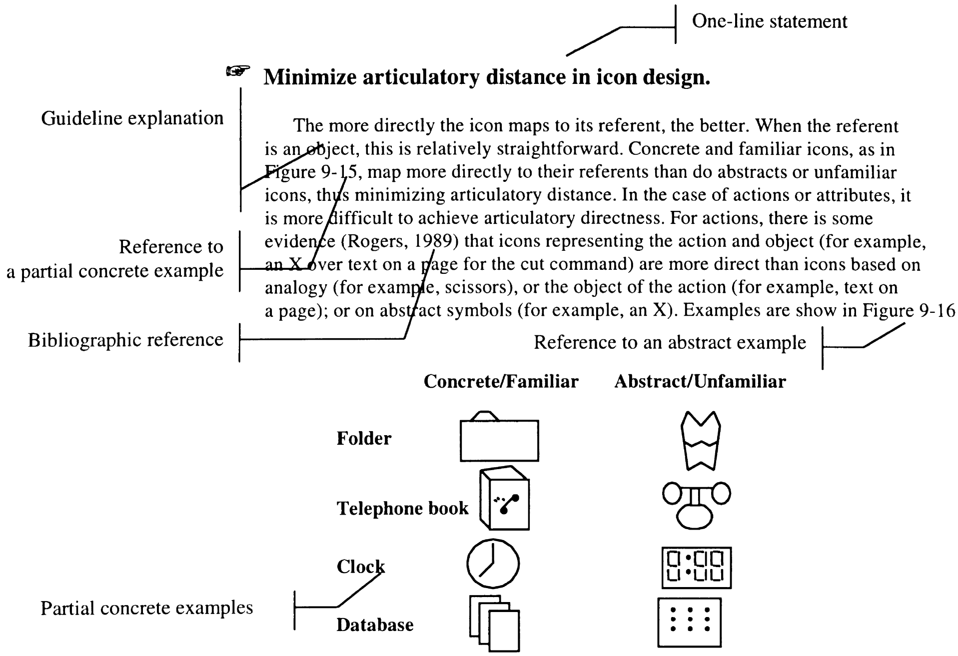
By using this identity card, we hope to highlight salient characteristics of ergonomic sources for collecting guidelines such as thrust in the origin, the presence of examples, explicit reference to experiences, scope, utility.

2.3.1.2. Results. Nine-hundred and thirty-eight ergonomic sources were flipped through and described according to the above format.[2] Mayhew's [13] guide is shown as an example below

- *Title:* Mayhew's guide.
- *Type of ergonomic source:* set of guidelines.
- *Bibliographic reference:* Ref. [13].
- *Contents description:* this guide is divided into 12 chapters according to a partition of interaction styles taken globally (Table 3).
- *Organisation:* personal.
- *Computing platform:* undefined, but examples are heavily tinted by the Macintosh platform.
- *Discipline:* human–computer interaction with references to other disciplines.
- *Amount of guidelines:* 293.
- *Amount of pages:* 619.
- *Amount of bibliographic references:* 316, of which 122 are experimental.
- *Amount of citations:* each chapter is always separated into two parts. The first part recalls major experimental results whereas the second delivers guidelines that can be deduced from these results. Citations are numerous. This dichotomy isolates the 'why' (the experimental results) from the 'what' (the guidelines), but removes any direct link between a guideline and its experimental citations. The following table shows the proportion between references bringing experimental results and references limited to principles. It reveals the sectors where empirical knowledge is extensive (for instance, menu selection, screen format design) or almost non-existent (for instance, selection of interaction style, design models, multi-windowing in chapter 13).

[2] The list of these ergonomic sources are available at http://belchi.qant.ucl.ac.be/guidelines/ErgoSources.doc (Microsoft Word 97 format) or ErgoSources.pdf (Adobe Acrobat format).

- *Guideline example:*



- *Example types:* numerous abstract examples which are partial and total, some partial concrete examples.
- *User interface types:* textual, graphical, vocal, virtual and tactile.
- *Discussion:* this guideline picks back most of traditional guidelines for modern UIs as well as for older textual UIs. It naturally starts with user considerations and cognitive profile. It then emphasises seven interaction styles (or dialog styles) ending up with a summary to select among them. The main interest of this guide consists in its deliberate aim to summarise for each chapter both experimental results and guidelines which can be derived following a similar structure. The reader who is interested by a quick application of these guidelines can skip the first part. Although this guide is obviously directed towards direct manipulation and multi-windowing, examples are as numerous as self-explanatory guidelines. Some small flowcharts mark out the application of guidelines.

2.3.1.3. Discussion It appears from the observed characteristics for each type of ergonomic source (Table 1) that these five types should be carefully distinguished.

The selection of ergonomic sources could rely on information provided in Table 1. In particular, applying the most general ergonomic source introduces few assumptions about the user's context, task type, or hardware and software computing platform. Sets of

Table 1
Characteristics of the five ergonomic source types

Ergonomic source/ characteristic	Design rules	Sets of guidelines	Standards	Style guides	Ergonomic algorithms
Form	Procedure manuals, technical reports, internal memorandums	Scientific journal papers, conference proceedings, research reports, handbooks	Official documents	Book, internal document for a company	Functional specifications coming from detailed analysis, software, particular techniques, routines
Origin	Company, development team	Experimental laboratories, research centres, research teams	Governmental organisations, public organisations, national or international standardisation offices	Hardware manufacturer, software vendor, private company	Research centres, research teams, development teams
Availability	Often private	Public, the most frequently, according to the origin	Public	Public if it is a hardware manufacturer of a software vendor; often private if it is a private company	Generally public, in spite of their potential ownership and trademark (rare)
Applicability	Specific to a family of interactive applications	General, but function of the experimental context	General, probably the widest	Hardware specific if from a manufacturer; Product specific if from a software vendor; Company specific if from a private company	Development environment specific (including computing platform, tools, procedures) to the company, yet often generalisable
Personalisation	Moderate (some guidelines could be personalised to satisfy particular requirements	High	High	Moderate	Depends on algorithm flexibility and the parameters (typically weak)
Support for personalisation	Reference to where the guideline comes from	Examples, exceptions, comments	Non-existent	Examples, predefined widgets	Depends on supported parameters (typically, weak)
Imposition	Managerial, organisational	By consensus	Contractual, legislative	By the manufacturer; By the software vendor; By the company itself (in this case, by the top management)	By computer technology and tools
Deviation	By designers with agreement of hierarchy, project leader, human factors expert	Allowed by people who proposed the guidelines	Formal agreement to accept an exception	Freely agreed for a manufacturer or a software vendor, rarely free for a company	Non existent, unless by unlocking the algorithm code from the software
Advantages	Provide clear entries, practical design means, increase productivity	Provide a more flexible assistance, but less efficient	Pay attention on key dimensions often used as commercial arguments of conformity, though	Increase look & feel consistency if respected	Speed up development time, insure consistency, guarantee automatic respect of guidelines incorporated in the algorithm

guidelines reduce this generality to the circumstances of the experimental context of their verification (when this is specified). Style guides deliberately aim at some specific hardware and/or software-computing platform. Tight specifications arise with design rules and can be implemented in a computer algorithm. We must consequently be conscious of these differences when selecting sources and their guidelines.

2.3.2. Other experiences in guidelines collecting

The need for a rigorous collecting was noted in [8]. Also, selection criteria should be clearly defined. If someone interested in guidelines could or does not want to proceed with guidelines collecting using the procedure described above, they can turn to sources claiming some form of comprehensiveness, e.g.: sets of guidelines, standards and style guides. Interestingly, attempts for guidelines collecting into sets of guidelines remain somewhat backward and sporadic: apart from the editor's [10], the Smith and Mosier document [14], published in 1986, remains the last known attempt to collect guidelines with all the drawbacks that this brings: obsolescence, incompleteness, interpretation trouble, unusefulness,…

In this special issue, Reed et al. note that this situation has improved with standards [15]:

- Although oldest standards emanate from official organisations, such as MIL-STD-1472D from DoD [16], they remain mostly private initiatives, dedicated to character-based UIs and to physical hardware aspects.
- Some more recent standards, such as NUREG-0400 from the American Regulatory Commission for Atomic Energy (NUREG) [17], include more information about graphical UIs, modern UIs or some cognitive aspects in interactive tasks; they nevertheless stay a private initiative.
- Contrarily, modern standards, emanating from international standard organisations, thus independent, distinguish themselves by the way they collect guidelines: they capitalise guidelines from all people and organisations who are able to afford a representativeness and a participation. Of course, this avoids those who neither have this opportunity nor the resource to attend periodic meetings throughout the world.

Reed et al. provide a substantial overview of standards from which guidelines can be collected (Appendices A–C). In Section 3.3 of this editorial, their selection logic is detailed: they constrain themselves to all formally presented guidelines. The same problem led Hardman and Sharatt [19] to consider every ergonomic source containing hypermedia related guidelines as a candidate source; they separated the contents into principles and guidelines, and then classified them into hypermedia functional areas.

2.4. Validation of guidelines collecting

As noted above, three factors are to consider: completeness, consistency and correctness.

2.4.1. Completeness of guidelines collecting

Although guidelines have been collected by merging ergonomic sources [10], either by consensus [15] or through formalisation [18], collections will remain perpetually incomplete with respect of UI usability for the following reasons [8,20,21].

- Unraised questions: e.g. what is the usability of novel interaction techniques uncovered in existing guidelines?
- Raised but unsolved questions: e.g. what is the most usable metaphor for a given context?
- Raised but only partially solved questions: e.g. what are the possible metaphors for a given context?

It is demonstrated [22] and reinforced throughout this special issue [23–25] that relying on guidelines could produce a more usable UI than just having a first sketch with no consideration of guidelines. In addition to the internal incompleteness (we can always lack a particular ergonomic source to answer a special question), we are confronted with the external incompleteness (we need more than ergonomic sources to guarantee UI usability). It is often recommended to use ergonomic sources in combination with scenario- or use case-based design/evaluation user tests. Ergonomic sources are therefore necessary, but not sufficient.

2.4.2. Consistency of guidelines collecting

Collecting guidelines inevitably induces conflicts between various sources. For the same usability question, ergonomic sources can lead to inconsistent results. For example, Schumacher tried to deduce design rules from three style guides [26]: the Apple Human Interface Guidelines [27], the Windows V3.11 style guide [28] and the OSF/Motif style guide [29]. Solving such conflicts requires adoption and awareness of an explicit consensus as reported in [15] when position diverge. The key point is to moderate each source using a range of criteria and to balance retention or rejection with sound, rational and explicit criteria. Guidelines conflict is most severe when they contradict themselves [20,21,30]; in identical conditions, they recommend opposite directions. Schumacher is disappointed that his three studied sources allow at least eight different ways to select a widget (for instance, a radio button, a selection list) for a single choice without refining the circumstances in which it is selected [23]. For this design problem, the three sources are inconsistent.

2.4.3. Correctness of guidelines collecting

A few ergonomic sources contain incorrect [21] or contentious guidelines [31]. Guaranteeing their detection and correction is difficult since their intrinsic flaws cannot be always detected [18], and they remain open to misleading expressions. Their correctness also depends on the basis of the guideline set [21].

Other sources are grounded on experimental results obtained in situ or in a usability laboratory [32,33]; some other enshrine commonly adopted views [7], some of them became axiomatic [34]; some other come from personal experience that no experimentation can (dis)confirm.

On one hand, some ergonomic sources are incorrect for specific UIs: general guidelines derived from fundamental research are suspect because their particular application involves human imponderables that can render them incorrect. On the other hand some ergonomic sources are also incorrect for general UIs: specific guidelines, derived from applied research contain particular conditions impeding any generalisation.

Table 2
Structural division of the guidelines corpus

Division	Division title	Amount of guidelines
1	Information input	528
2	Information display	970
3	Dialogue	162
4	Graphics	393
5	Interaction media	123
6	Interaction styles	609
7	Guidance	370
8	Messages	125
9	On line help	155
10	Documentation	171
11	Evaluation	8
12	Methodology	86
Total		3700

Selection and extraction are also error prone, particularly when the sheer volume of ergonomic sources requires condensation [18]. The main guards against incorrectness are the quest for globality, for exhaustiveness, for representativeness and a restriction to experimental sources.

3. Guidelines organisation

3.1. Definition and goals of guidelines organisation

An initial set of guidelines will be copious and will range over many levels of rigour and credibility. Both can be ameliorated by a good organisational structure. Providing this is the main goal of guidelines organisation.

3.2. Procedure of guidelines organisation

To attain this step's goal, initial guidelines are subject to a careful study, to a translation of their initial expression into a canonical one based on a uniform scheme of classification that gives rise to an appropriate structure for the guideline user's goal.

3.3. Problems and analysis of guidelines organisation

Carrying out this procedure exacerbates problems already met in step 1 and introduces more once any re-expression and organisation are applied. These are by nature rigid.

3.3.1. Our experience in guidelines organisation

3.3.1.1. Method. Translation of initial guidelines into their final expression requires a translation method. We retained the translation of an initial guideline from step 1. The

Table 3
Contents description

Chapter	Title of chapter	Amount of guidelines	Amount of experimental references/amount of bibliographic references
1	Introduction	/	1/5
2	The user profile	/	7/23
3	Conceptual models	18	5/17
4	Dialog styles: menus	19	20/31
5	Dialog styles: fill-in forms	48	7/11
6	Dialog styles: question and answer	8	0/1
7	Dialog styles: command languages	17	8/19
8	Dialog styles: function keys	15	4/9
9	Dialog styles: direct manipulation	13	10/25
10	Dialog styles: natural language	8	5/17
11	Dialog styles: summary	3	0/4
12	Input and output devices	27	25/60
13	Organisation of functionality	8	3/5
14	Screen layout and design	53	14/38
15	Response time	7	1/6
16	Error handling	41	1/5
17	User documentation	35	9/29
18	Development methodology	0	2/11

underlying *general guideline model*, refined by trial and error [35], has the following attributes:

- a guideline identifier;
- a brief and representative title;
- a complete statement written in a natural language;
- a list of bibliographic references quoting this guideline;
- the linguistic level to which the guideline is applied: one of: goal, pragmatic, semantic, syntactic, lexical, alphabetic and physical;
- the design ergonomic criteria respected by the guideline;
- the evaluation ergonomic criteria guaranteed by the guideline;
- the utility and usability factors satisfied by the guideline;
- a mathematical formula expressed in terms of the first order predicate logic;
- a rationale justifying the guideline;
- one or many negative examples depicting UIs which violate the guideline;
- one or many positive examples depicting UIs which respect the guideline;
- one or many exception cases, with one or many positive or negative examples each;
- the set of relationships established with other guidelines according to a link typology;
- the interaction style(s) for which the guideline is valid;
- the interaction media related to the guideline.

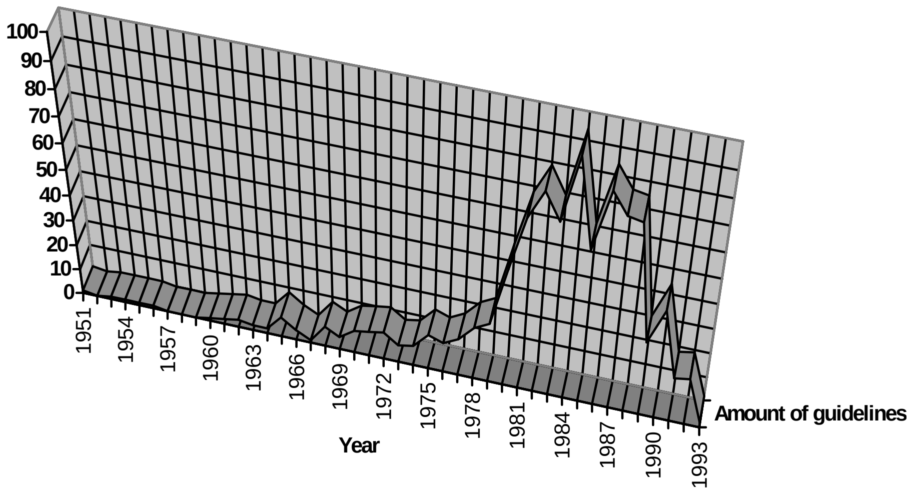


Fig. 2. Guidelines distribution by their publication year.

The choice, the definition of these attributes and their respective values are detailed in Refs. [4,35]. In parallel with this model, we decided to insert each guideline into a hierarchy of taxonomic sections where each section is characterised by an identifier, a name, a definition and a goal.

3.3.1.2. Results. From the 938 ergonomic sources collected in Section 2, 3700 guidelines were extracted and expressed in terms of the above general model as far as possible and organised in divisions as depicted in Table 2.

Each division is hierarchically decomposed into taxonomic sections, which were merged from headings in Smith and Mosier [14], Mayhew [13] and Horton [36]. Several presentation design options, that did not appear in the guideline sources, were deliberately added: e.g. selection of widgets for information input ([10], Section 1.4, pp. 40–57) and for information output (Section 2.2, pp. 110–119). Emphasis was placed on window identification (Section 2.2.6, pp. 115–119) and the placement of widgets into a container (Section 2.3–2.5, pp. 119–152).

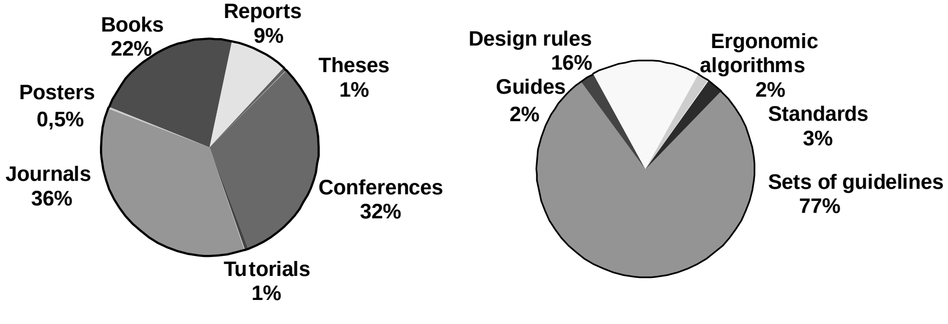


Fig. 3. Guidelines distribution by: (a) nature; (b) type of source.

During guidelines collecting and organisation, several statistics were computed:[3]

- the guidelines distribution by their publication year (Fig. 2);
- the guidelines distribution by nature (Fig. 3a);
- the guidelines distribution by type of source (Fig. 3b);
- the guidelines distribution by source discipline (Fig. 4);
- the guidelines distribution by source origin (Fig. 5).

3.3.1.3. Discussion. The guidelines organisation according to the above method shed some light on a series of overlapping problems that are now summarised.

Initial guideline sets are extremely large [20,37]: for instance, 944 guidelines in Ref. [14], 1812 in Ref. [38], 3700 in Ref. [10]. Guideline translation will proceed smoothly when each translation adds a unique rule. Translation becomes more complicated when existing rules are retranslated from a new source. This situation can arise when guidelines from different sources translate to the same advice or when an identical guideline is repeated in several sources. If the set of guidelines becomes unmanageable at this point, little benefit can be gained in future steps.

Initial guidelines rapidly grow over time [3,8,37], as our experience and research improve [39]. However, an examination of the guidelines distribution by publication year (Fig. 2) reveals a quick start culminating in an apparent slow extinction (our collecting ended up in June 1994). We believe there are mainly two reasons for this trend:

1. Guidelines do not evolve as quickly as computer technology: for instance, new multimodal UIs are already implemented and used before there is any good understanding of determinants of their usability.
2. Many elementary and fundamental guidelines have already been discovered, the guidelines that remain to discover are proving to be more elusive, particularly as experiments become more complex and expensive as they simultaneously manipulate several usability factors.

Initial guidelines are scattered throughout sources that are sometimes hardly accessible [8,40]. One initial isolated guideline may come from a specialised journal; and other complementary one is found elsewhere. According to Fig. 3, one records that three quarters of guidelines are coming from sets of guidelines, mostly conference papers. A person in a commercial environment does not generally have the time nor capability to explore such a distribution in a limited amount of time.

Initial guidelines are biased towards particular disciplines: the corollary of the previous problem is thus the multidisciplinary challenge of guidelines (Figs. 2 and 3). If human–computer interaction and software ergonomics provide most guidelines, some related domains, sometimes distant, also provide input: neurophysiology, graphic arts, semiotics. Translating initial guidelines from such a variety of disciplines is no easy task [41]. Guidelines quality also varies by discipline. Some authors regret that the contribution

[3] These statistics were computed from the value of the different attributes appearing in the report format of each ergonomic source (Section 2.3.1.1).

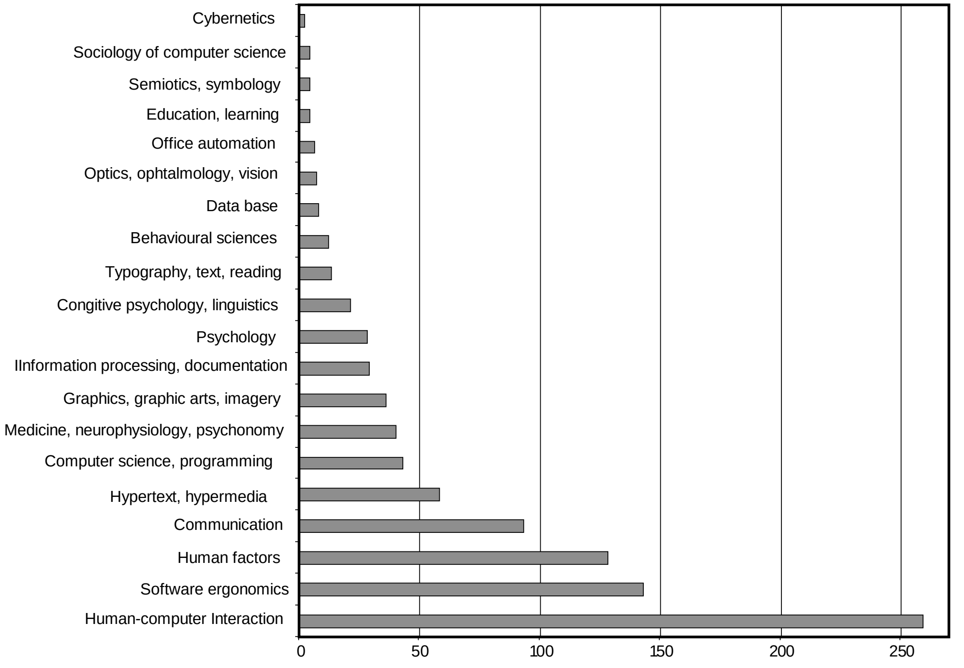


Fig. 4. Guidelines distribution by source discipline.

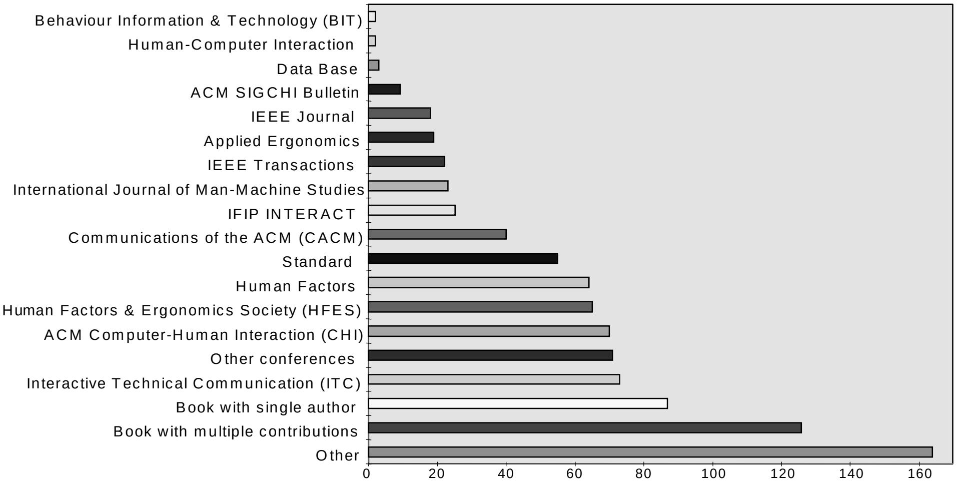


Fig. 5. Guidelines distribution by source origin.

of some disciplines such as cognitive psychology to software ergonomics remain insufficient due to their immaturity [33,42].

Initial guidelines strongly vary by coverage [37]. Guidelines of CUIs could be extended to GUIs, but only to a certain extent [26]. Some guidelines are old (as in Ref. [14]), sometimes wrong (as in Ref. [43]). Guidelines rain down for some application domains (e.g. for text editing), for some widgets (e.g. for selecting them), for some interaction media (e.g. for selecting them), for some interaction styles, for user participation, for philosophic, ethic motivations and physical details. Their contents are presented in different styles across a range of applications and foci.

Initial guidelines also vary considerably by level of detail [37,44]. This variation starts with the five types of ergonomic sources (Table 1) and is continued at the level of individual guidelines. Some guidelines are very general [8,20] in the sense they consist in statements of generic principles, others are very precise and are overly detailed [41]. Avoiding generality or specificity does not solve anything: the more specific a guideline becomes, the less it could be applied. Continua spanning from a general guideline into further decompositions into refined, specific guidelines is rare. Some guidelines are tied to changing contexts [45]: for instance, “dialog control should vary according to the user’s experience level” cannot always be satisfied by a one-off design decision. Some other guidelines are unequivocal [4]: for instance, “to input a Boolean information, select a check box”. Yet other guidelines can be broad and vague; still others can be narrow, hard to pin down, difficult or too easy to assess. This can cause disorientation [8].

The two major problems encountered at this step are the direct consequence of these previous problems:

1. Guidelines should all be translated into some canonical form. Since initial guidelines are written with vocabulary and conventions that are local in the different disciplines, they all vary along the axes identified above and require a constant translation into a common format.
2. Guidelines are insufficiently classified, a much regretted flaw [3,8,32,35,40], which is unfortunately aggravated by an even greater range of classifications:
   - by ergonomic criteria, as in Scapin’s ergonomic guide [46];
   - by usability factor, as in ISO 9241 standard [47];
   - by interaction style, as in Mayhew’s guide [13];
   - by widget, as in IBM CUA style guide [38];
   - according to an object-oriented model on input/output, as in ITHACA report [8];
   - by linguistic level, as in Marcus’s “User Interface Standard Manual” [48];
   - by importance level, as in Banks’s standard [43];
   - by type of widget, as in Farenc’s ERGOVAL automatic evaluation tool [49].

Every attribute of the general guidelines model could potentially become an organising category at a high or intermediate level. Unfortunately, no known study underpins any preference for a particular classification. The taxonomical approaches remain the most frequently used support for classification. However, a particular organisation (for instance, in order to certify a standard compliance) that suits one goal can become totally inappropriate for another (for instance, in order to learn about usability from the standard).

3.3.2. Other experiences in guidelines organisation

Reed et al. report various experiences for organising initial guidelines in American and international standards [15], where the organisation is mostly based on a consensus vote by participants.

Kemp and Buckner [18] present a more thorough analysis that reinforces our criticisms on the inadequacy of existing classifications (Sections 1.1 and 1.2 of their paper). In their Section 2.2, they detail their method for classifying hypermedia guidelines and, in Section 2.3, they report on their experience of applying it:

1. First, an initial classification, based on a set of principal categories, was set up from the categories discovered in ergonomic sources.
2. Then, by consciously raising the question of appropriate length and detail, these principal categories were subdivided into secondary categories during the guidelines collecting.
3. Finally, from a compilation of several sub-categories found in the literature and from an argumented reasoning about required and missing sub-categories, a final list was set sup according to the parameters: base discipline, author type, detail level (primary, secondary, tertiary), functional and operational characteristics, procedural orientation.

Another problem identified by Goffinet and Noirhomme [50] deals with the disappearance of connections existing between initial guidelines. When underlying connections between the initial guidelines are preserved, guidelines foster valuable intellectual paths through guidelines. As soon as these connections vanish, either in their sources or during organisation, these paths are hidden. To restore them and to validate those that are kept, Goffinet and Noirhomme [50] propose a method for automatic (re)discovery of syntactic links between initial guidelines. This method is grounded on neighbourhood statistical formula common in information retrieval. The authors applied these formula with varying degrees of success to the well-known Smith and Mosier document [14] where such connections exist. Their results not only reinforce some existing links but also reveals missing ones. Since their method is supported by a software tool, it could be applied to every set of initial guidelines. It becomes as useful as the amount of guidelines is huge. For instance, the NUREG-0400 standard [17] could greatly benefit from this method since it contains more than one thousand of guidelines without any connection.

Syntactic links manipulated by Goffinet and Noirhomme follow the “is neighbour to” relationship. A more elaborate link typology could include: “is the inverse of”; “inherits from” vs “is inherited by”; “generalises” vs “specialises”; “precedes” vs “follows”; “precises”, “is a”; “is comprised in” vs “belongs to”; “includes” vs “is included in”; “is a user of” vs “is the owner of”; “has property”; “is an example of”; “is provided by”, “uses” vs “is used by”; “is a requirement for”; “is alternative to”; “is contrary with”; “is conflicting with”,…

3.4. Validation of guidelines organisation

Let us immediately recall that we have to cope with incompleteness from the previous step, along with its inevitable inconsistency and the possible incorrectness. Thus, the procedure applied in the current step cannot improve these initial shortcomings; at the best, it will make it no worse. At the most, both translation and classification errors will make things worse.

3.4.1. Completeness of guidelines organisation

Although the general guidelines model was intended to be the most general possible to avoid loss of information, it nevertheless causes an important reduction of guidelines expressiveness, and thus of their completeness. Indeed, each model attribute is a simplifier and no guideline cannot be necessarily and unequivocally fitted into these attributes: some fragments do not match any model attributes (for instance, the cognitive type of the envisioned task); some fragments match to several attributes (for instance, design and evaluation ergonomic criteria); and some fragments match to a single attribute, but in a way that is not obvious (for instance, the linguistic level). An attribute sometimes remains unfilled as it is unmatched by any guideline fragment. In particular, an important loss of information is introduced by a presumed unique linguistic level. Although the underlying aim is well intent, the result can be an arbitrary allocation. Many guidelines could really be attached to several criteria and linguistic levels with a weight related to each assignment. To derive such figures from initial guidelines would be a complex process exacerbated by the residual incompleteness of the guidelines (for instance, link incompleteness [50]).

The classification of translated guidelines into a taxonomy also appears as a reductive operation: despite the wish for comprehensive sections. Sections are sometimes irrelevant, others are poorly defined [18]. Pinsky regrets that classifying guidelines into simple sections such as screen, data input, commands, messages; input/output induces a reductionist vision of the interactive task addressed by guidelines [33]. This fragmentation can only be overcome by a complete task analysis.

3.4.2. Consistency of guidelines organisation

Notwithstanding the desire for systematisation, guidelines translation is not immune from subjective interpretation: the examination of any initial guideline and the assignment of fragments to model attributes remains subjective, even if we want to make the contents objective or to “de-subjectivise” it. This “de-subjectivisation” reveals itself as hazardous as guidelines strongly vary in scope. To master this, initial guidelines which are objective (for instance, experimentally verified guidelines) are separated from those which are subjective (for instance, common practices, personal taste).

3.4.3. Correctness of guidelines organisation

At the level of initial guidelines, there is some probability of incorrectness because of their exposure to Meyer’s seven deadly sins: noise, silence, overspecification, contradiction, ambiguity, repentance, and inconsistency. Kemp and Buckner [8] observe that the probability of incorrectness grows with the volume of initial guidelines.

Incorrectness inevitably increases when any selection action is performed, as in the classification of a translated guidelines using a taxonomy. Kemp and Buckner claim that bias or error in taxonomy although inescapable, is discernible if the strategy used for building the taxonomy is made explicit and open to analysis. They also recommend to allocate taxonomy building to several independent simultaneous teams to maximise the expertise applied to consolidation.

4. Incorporation of guidelines into a methodological approach

4.1. Definition and goals of guidelines incorporation

Early consideration of software ergonomics requires an iterative and eager UI design process [20]. It is thus important [8] to associate sections of guidelines with the phases identified in Section 1, i.e.: specification, design, prototyping, development, evaluation, documentation, certification or teaching and learning. The goal of this step is to locate points within these phases where organised guidelines should be considered, to specify which should be considered by identifying local guidelines (for a phase), global guidelines (for all phases) or pervasive ones (for several continuous phases).

4.2. Procedure of guidelines incorporation

To incorporate guidelines within a development methodology, we must:

- select a methodology to develop interactive applications that is refined and practical enough for a given family of interactive tasks (for instance, MUSE [51], LUCID [52], MOST [53], TRIDENT [54], DIANE + [55]);
- list questions raised during each phase of the selected methodology;
- browse the organised guidelines;
- mark each (section of) guideline(s) providing some answer to the questions listed earlier.

4.3. Problems and analysis of guidelines incorporation

Mosier and Smith [56] reported that their guide [14] was exploited during design by 41% of persons, during prototyping by 25% of persons, and during evaluation by 18% of persons. It would be a pity to restrict guidelines to these sole phases.

4.3.1. Our experience in guidelines incorporation

The above procedure was applied on guidelines from Ref. [10] by attaching them to phases and subphases of the TRIDENT methodology [54]. Attaching guidelines to (sub)-phases proved to be effective only in the framework of a specific methodology: if the methodology changes, the association of guidelines could change.

The browsing strategy needs to be considered in advance. Either guidelines can be examined to attach them to (sub)-phases or each (sub)-phase can be examined to attached it to one or more (sections of) guidelines (for instance, the sub-phase "selection of abstract interaction objects" is attached to the taxonomic section "selection of widgets for acquiring/producing simple, complex information" [10]). It proved to be easier to follow the first strategy. If a guideline remains basically dedicated to a particular phase, it may still be relevant to other phases. Clarifying the relationship between (sub)-phases and guidelines is challenging not only because of ambiguity but more because of guideline contents which do not answer questions such as: where should this guideline be used? When should it be used? How? Under what conditions? [3,37].

Organised guidelines can also be associated with specific development roles. Guidelines should therefore be associated with these roles [57]:

- a task analyst hopes that guidelines are organised according to a taxonomy characterised by task attributes, and not according to interaction styles (for instance, information retrieval, decision taking, form fill-in, multi-criteria search, problem solving);
- a project leader only pays attention to high level guidelines (for instance, selecting an appropriate metaphor) which are the most likely to influence positively or negatively the success story of UI implementation; this person is responsible for selecting high level design options;
- a human factors expert may assume that guidelines are sorted by the cognitive principles from which the guidelines are derived;
- designers expect that guidelines organisation is such that they could instantly isolate specific guidelines and solve conflicts between them by ranking them into ordered sequences;
- programmers think that guidelines should be organised with respect to the widgets of the inter-active application because their responsibility mainly covers this part;
- a UI evaluator prefers guidelines expressing UI usability in terms of measures.

As organised guidelines are detached from contexts of use this tremendously limits associating guidelines with task types. Some guidelines are sometimes unsupported by experimental designs that served to validate their applicability [21]. This information loss makes guidelines absolute when they should be considered as relative to a context, e.g. the task type or the user's cognitive profile. If we were able to explicit all implicit conditions, the cognitive cost of applying guidelines would have been reduced [58].

4.3.2. Other experiences in guidelines incorporation

Rather than attaching guidelines to (sub)-phases of an existing development methodology, Reed et al. [15] detailed a methodology for applying guidelines based on the organisation of their standard (Section 3). This methodology benefits from its autonomy, but still has the disadvantage of requiring revisions to existing development life cycles.

Carter takes a similar approach proposing "Usability First" [23], a methodology for developing interactive applications that naturally includes access points to guidelines [53]. The most interesting part consists in its direct identification of guidelines organised into available taxonomies. For this purpose, the five types of ergonomic sources that emerged in Section 2 are considered. This methodology also supports development roles in:

- conducting a task analysis to elicit informal specifications;
- identifying guidelines that respect these specifications and applying them;
- developing usage models by transforming these specifications into design options;
- evaluating usability of these design options with respect to appropriate guidelines.

One also observes that guidelines used in specifications or design could differ from those used in evaluation; does this mean that evaluation should be restricted to guidelines overlooked during previous phases in the methodology?

When the activity domain is known, organised guidelines that are relevant for each (sub)-phase become more obvious to discover: Moussa et al. take advantage of the domain-specific nature of their work to more precisely locate such guidelines into their global design and evaluation methodology (it is dedicated to dynamic and continuous process control (Figs. 1 and 2 in [25]).

In other respects, finding out that some guidelines can be used in similar (sub-)phases in different methodologies, even in different activity domains, could support the conclusion of Javaux et al. [59]. That there are more general guidelines which are activity/domain independent (e.g. the presentation design of visual displays).

Grammenos et al. [30] and Henninger [24] focus on the support which a software tool could provide for guidelines. Although their SHERLOCK and GUIDE tools are open and flexible in their approach to methodology usage, they induce additional intervention points in a methodology more than they specify them.

4.4. Validation of guidelines incorporation

4.4.1. Completeness of guidelines incorporation

From all problems identified and analysed up to here, the detachment of guidelines from their contexts of use introduces the greatest risk of incompleteness. Without any context understanding and without explanation where it can be used within a development method, the resulting usability depends on random evolution or a developer's experience and skills. Once guidelines are incorporated into development methods, they are consequently more incomplete than at the previous step.

4.4.2. Consistency of guidelines incorporation

The variance of experimental contexts, the potential risk to apply guidelines in a different context and the risk of attaching a guideline to an inappropriate all introduce new risks of inconsistency.

4.4.3. Correctness of guidelines incorporation

Risks to consistency also introduce risks of incorrectness. "De-contextualisation" of guidelines could be addressed by formulating guidelines in terms of task and user parameters. This generates the need for a task representation formalism, a user formalisation [33], where correct interpretation would require a solid competence not always acquired by developers [21,32]. Where task and user specifications are misinterpreted, the resulting UI usability could be radically different from the one expected, thus discouraging future use. In conclusion, correctness depends heavily on the skills of potential guidelines users.

5. Guidelines operationalisation

5.1. Definition and goals of guidelines operationalisation

Currently, guidelines integrated into methodologies are used manually. Such guidelines have often provided for development teams, but have tended to be ignored, underestimated and underused for several reasons [44]:

1. First, the current form of the guidelines themselves, as already noted, implies a difficulty in interpretation (designers namely experience trouble with 30% of accessed guidelines [58]), their decontextualisation limits application, variations in importance and appropriate target development role.
2. Second, there are reasons related to the onerous and inefficient demands of guideline management in their current form. The huge amount of guidelines exceeds human mental capability [60]: the paper format turns developers into medieval scholars (up to 15 min for a sole guideline according to [7]). Only 58% of interviewed designers looked for guidelines related to a particular problem [7], they infringed on average 11% of guidelines because they did not know how to access them [58].

The appreciation of these inconveniences led many researchers to envision alternative support to paper support [30]: audio support, video support, and software support. In this special issue, we exclusively concentrate on software support. We call this software "tools for working with guidelines". Methodologically incorporated guidelines do not lend oneself in their current form to a direct implementation within such a tool. A further stage of guidelines operationalisation is thus aimed at re-expressing for each design problem a subset of relevant guidelines that can be directly reused and exploited in a software tool.

5.2. Procedure of guidelines operationalisation

The procedure for reaching this goal has three steps: first, a contraction-extension process is applied to isolate a subset of guidelines for to a design problem; then, a formalism for internal representation of guidelines in software is selected; finally, the pertaining guidelines are expressed in terms of the selected formalism.

5.3. Problems and analysis of guidelines operationalisation

5.3.1. Our experience in guidelines operationalisation

In our first experience (SIERRA [35]), a decision was taken to not privilege a design option or a particular problem. Since no guideline can be a priori separated, we did not apply the contraction-extension process.

The internal representation formalism chosen was a record with fields matched to the attributes of the general guideline model introduced in step 2. The initial data was mainly text, but rapidly evolved to other types [9]: images for screen dumps, examples, sounds for vocal annotation or guidelines related to sound, video sequences to depict the guidelines application in a form where dynamics can be better understood, e.g. records of interactive sessions, interactive application. Guidelines were stored in a Microsoft Word 2.0 file with external yet linked references to multimedia resources, records and interactive applications. Fig. 6 reproduces a screen dump of SIERRA displaying guidelines from Smith and Mosier [14].

In a second experience (SEGUIA), a decision was taken to restrict coverage to the selection of widgets [61,62] and their placement in a container [63]. This time, all steps of the above recommended operationalisation procedure were entirely applied. The results of this are now described.

5.3.1.1. Application of the contraction–extension process. First, from methodologically incorporated guidelines, the most important guidelines are to be identified. The heart of ergonomic know-how related to each problem should be located in this guidelines subset: this is a *contraction movement*. The more contracted the guidelines are, the more general they become. The more general guidelines become, the less they could be applied to particular situations. The process underlying contraction is:

- make assumptions about the likely context of use (for us, business oriented applications);
- focus on a design option (here, to select widgets and to place them in a container);
- reduce the set of methodologically incorporated guidelines to ones related to the focus and the assumed context of use;
- generalise specific guidelines from this subset to more general ones via *abstraction* (for instance, with respect to a given parameter, to a portion of guidelines attributes), *assimilation* (for instance, a specific guidelines could be assimilated to a more general guidelines), or *comparison* (for instance, a guideline with higher priority could be preserved rather than a guideline with lower priority).

Second, to make the resulting guidelines usable, they need to handle exceptions and support conflict resolution for particular situations; this is an *extension movement*. Extending guidelines involves extracting the rationale from the general guideline model and to develop it so that guidelines join particular situations again. The process underlying extension is:

- consider a set of contracted guidelines;
- specialising these into specific guidelines to cover a range of possible applications (for instance, a guideline valid for a class of abstract interaction objects could be stated for all concrete interaction objects that realise the abstract objects), counter-elimination (for instance, a guideline conflicting with a pernicious guideline leads to an elimination of the secondary guideline), conflict resolution (for example, only the most important of two conflicting guidelines is extended, the other is dropped),…

The main rationale of this process is the wish to weight ergonomically incorporated guidelines relative to the expected context of use. These guidelines are not valid in any absolute sense. On the contrary, their usage is contingent: thus while the core rationale of the guideline remains apparent, this is accompanied by specific applications for some circumstances.

5.3.1.2. Selection of an internal representation formalism. With a set of extended guidelines available, one must select an internal representation formalism that allows us to express these guidelines in a software tool. Denley et al. formalise guidelines towards explicit integration according to the following equation [64]:

$$X = f(T, U, C, P_s) \Rightarrow Y \text{ because } R(Y = f(X)) \text{ with guarantee } p > \text{threshold} \quad (1)$$

where X is the set of triggering attributes; T the set of attributes for the interactive task; U the set of attributes for the user; C the set of attributes for the physical environment (computer); P_s the expected performance; Y the prescription; and R is the rationale.

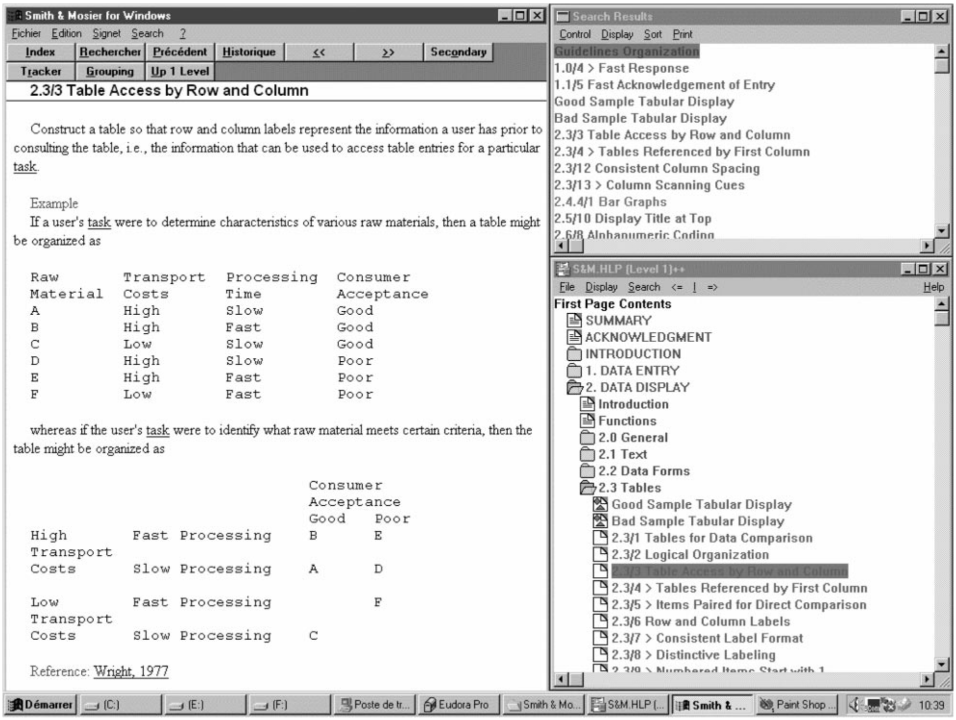


Fig. 6. Display of guidelines in SIERRA.

Eq. (1) probably gives the ideal scheme as it expresses a guideline as a function of all parameters for a task analysis. We deeply regret not to be able to systematically use this scheme, but there were too many insuperable difficulties. Among them, decontextualisation of guidelines completely undermines assignments to T, U, C and P_s. More simply, Scapin developed an operationalisation according to the following coding schema [8]:

$$\text{IF (Premises) FOR (Criteria) THEN (Conclusion)} \tag{2}$$

where (Premises) could be any user type, any task type, any design option, any contextual factor (Criteria) represents the related design ergonomic criteria (Conclusion) could be a design option, an action, an item

This coding schema is fairly close in form to production rules, which are known to be relevant and applicable in many UI development environments. This model expresses a relationship between guidelines and ergonomic design criteria. Since we cannot precisely specify many instances of this relationship or weights for criteria, the clause "FOR (Criteria)" was removed to reach the following equation:

$$\text{IF } (\text{Attribute}_1 = \text{Value}_1) \text{ AND } (\text{Attribute}_2 = \text{Value}_2) \text{ AND} \ldots \text{AND } (\text{Attribute}_n = \text{Value}_n) \text{ THEN (Conclusion)} \tag{3}$$

where $Attribute_i$ could be any potential parameter from task analysis or functional analysis, $Value_i$ the possible value for this parameter, and Conclusion the prescription for a design option, or a problem.

5.3.1.3. Results discussion. For the first focus on widget selection, contraction–extension resulted in 20 contracted guidelines. The process involved expressing which objects should be generally selected, restricting interaction tasks to entry and editing, generalising over input data types, and restricting the range of selected values to 1 (simple choice) and $n > 1$ (multiple choice), and varying the definition of domain values [4]. With 7 additional rules for replacing a previously selected widget on the basis of a user's preferences and experience, and also information attributes, a subset of 276 extended selection guidelines was obtained. This subset brings three advantages:

- during each extension, the ergonomic rationale behind the guideline enabled us to produce an explicit, thus communicable, reasoning to the user, if needed; this reasoning was reused as many times as required;
- all 276 guidelines were expressed in terms of the internal representation as a production rule according to Eq. (3);
- to highlight identical portions of antecedents in production rules, a selection tree technique has been proved as the most efficient, the most visually practical and the most flexible with respect to other formalisms such as bulleted lists, flowcharts, vertical tree, decision table, selection table.

For the second focus on placing widgets, the contraction firstly worked on the following characteristics:

- guidelines were grouped into themes (for instance, grouping versus separation, balance versus instability);
- guidelines related to various alternatives for the same placement problem were identified; for instance, the guideline "identification labels of control widgets should be located above or at the left of the widget, with left or inferior justifying" combines the two guidelines for the two alternatives;
- some senseless or unusable alternatives were withdrawn;
- superordinate guidelines were retained; for instance, the guideline "the margin of widgets should be 10 pixels" has two superordinate guidelines "the margin should be sufficient", "the margin should be consistent";
- guidelines were stated as much as possible in terms of classes of abstract interaction objects [61] rather than in terms of simple abstract interaction objects or various concrete interaction objects; for instance, the guideline "identification labels of an object should be located above or at the left of the identified object" contracts the same guidelines, but stated for the particular widgets, such as in the guideline "the identification label of a list box, drop down or not, should be located at the left and aligned with the bottom of the box".

From a set of 65 contracted guidelines, a subset of about 350 extended guidelines was obtained. Two advantages are identified: any guideline repetition for different objects has been identified and removed; also principal guidelines have been embodied in two ergonomic algorithms for automatic and computer-aided placement of objects in a window [63].

5.3.2. Other experiences in guidelines operationalisation

No method exists yet for obtaining an appropriate subset of guidelines for a particular problem. A real opportunity for conducting research can be found here.

Fritzsche and Michel [64] claim that they restrict themselves to guidelines suited for Electronic Product Catalogues (EPC), but do not present any method for applying this restriction. Similarly, Moussa et al. [25] consider general or specific guidelines for dynamic, continuous dynamic control.

Grammenos et al. [30] claim that their Sherlock tool remains open to every type, old or new, of guideline from any type of ergonomic source. Every guidelines subset could particularly be considered as input for a given problem. They then recognise that a consolidation of guidelines is required preliminary to their exploitation in Sherlock.

Henninger argues that guidelines' contextualisation should be the central question [24]. This is why his GUIDE tool does not a priori consider that guidelines are specific to a problem, but is oriented towards the progressive identification of specific guidelines as the tool is used. The principle is to keep in an organisational memory a maximum of design/evaluation UI cases to accumulate experience acquired in usability by any organisation (for instance, a development team) during time. In order to refer to when a new UI development is initiated, when an existing UI is updated, modified. Every time a given UI should compel with a guideline, a case is created for this guideline and it can be specialised into more refined guidelines as the UI development and experience grow.

About the *selection of internal representation formalism*, other experiences are extremely interesting. Fritzsche and Michel [64] completely devoted their Section 4.1 entitled "Formalisation and formulation of guidelines" to this question. In their TASSI tool, they also selected a formulation in the form of production rules close to the one found in Eq. (3). What is more interesting is that they formalised them into Horn clauses (as in the Prolog programming language) into facts, rules and requests written in the following goal oriented formalism:

$$\begin{aligned} &A. && (fact) \\ &A \quad :-B, C, \ldots && (rule) \\ &\quad\;\; :-B, C, \ldots && (request) \end{aligned}$$

This formalism is efficient where a theorem prover can automatically apply, verify guidelines either in backward chaining (for UI design purposes) or in backward chaining (for UI evaluation purposes where UI properties are expressed as facts).

In their ERGO-conceptor tool, Moussa et al. [25] state three reasons for their choice of production rules:

1. A paired (antecedent, subsequent) formal representation is possible.
2. Production rules allow and encourage restructuring, guidelines grouping and, most of all, operations required for contraction–extension.
3. Their feasibility has already been proved elsewhere (as in [6]).

More originally, the UI is stored in ERGO-conceptor as LISP lists containing facts to be exploited by guidelines embodied in the inference engine. In this way, experiences of Fritzsche and Michel and Moussa et al. are corroborated: an expression in terms of predicate logic of a certain order (0 or 1) allows an automatic exploitation of guidelines provided that an automatic tool such as a parser, a theorem prover, and a property verifier is available.

In their Sherlock tool, Grammenos et al. [30] report that Sherlock guidelines should be programmed separately, but this is not a constraint. A guideline could be written in a Visual Basic 5.0 procedure (the implementation language of Sherlock) or in any other programming language provided a dynamic link library (DLL) is made available.

In his GUIDE tool, Henninger [24] is representing guidelines as a hypertextual hierarchy linked to cases and rules. However, rules are used to match guidelines with the various design contexts of cases; these rules are coded as question–answer pairs in a decision tree whose leaf nodes contain the textual statement of guidelines.

In conclusion, we observe that guidelines operationalisation remains *passive* as long as guidelines cannot be executed, tested, evaluated (as in GUIDE [24], SIERRA [35], NaviText Sam [6]); this operationalisation becomes *active* if guidelines could be applied at design time (as in SEGUIA [4]), tested at evaluation time (as in Sherlock [30]) or both (as in TASSI [65], ERGO-conceptor [25]).

In the first case, we successively went from a hypertext technology (as in NaviText Sam [6]), through a hypermedia technology (as in DRUID [66]), multimedia (as in Making It Macintosh [67]), data base management system (as in DRG [17]), or combining both (as in SIERRA [4,35]) to end up with Web technology (as in GUIDE [24]). What seems here fundamental is the ability to couple data base facilities for guidelines access, selection, searching, sorting with hypermedia facilities for guidelines browsing, organisation and contextualisation.

In the second case, basic predicate logic seems feasible, applicable and efficient when automatic tools are available; it then suffices to express a new guideline in its internal representation to consider it automatically, if possible. If this is not possible or if the guideline operationalisation requires a more complex procedure, a dedicated programming would be preferred.

Since no study exists today comparing different internal representations (if…then…else rule, LISP list, Prolog clause, coded procedure, decision tree,…), we cannot yet tell what the preferred notation would be. The only conclusion we can draw is that the greatest expressiveness for a guideline comes with first order predicate logic.

5.4. Validation of guidelines operationalisation

5.4.1. Completeness of guidelines organisation

The nature of the three performed operations (that is contraction, extension, and internal representation) further reduces guidelines. Once again, the process of reduction is not readily understood, although the greatest seems due to internal representation. Scapin [8] found that some guidelines are difficult to express using an internal representation because their contextual characteristics are not expressible as simple conditions on attributes (for instance, is the user experience experimented?) or that are expressible into such conditions, but on missing attributes.

5.4.2. Consistency of guidelines organisation

Contracted guidelines are in contrast internally consistent from the outset since this is a goal of contraction: to reach an internal consistency by sacrificing conflicting less important, out of context guidelines. This internal consistency is preserved through extension and internal representation.

5.4.3. Correctness of guidelines organisation

Guidelines should normally be syntactically correct, but this does not eliminate semantic errors. Contraction can introduce more serious bias by its filtering. The diversity of guidelines before contraction, as regards syntax and semantics, requires care and attention during contraction, especially as the process results in dependencies on a default: models of the user designers' knowledge and software tool users' knowledge. Extension should not introduce further incorrectness, except by amplification. The internal representation introduces two potentially severe drawbacks:

1. The selected final representation (Eq. (3)) is in itself an important reduction with respect to other (currently inapplicable) representations (Eqs. (1) and (2)).
2. The expression of an extended guideline into attributes of antecedent and consequent of a production rules introduces a very important distortion that should not be neglected; indeed, the statement of some guidelines (e.g. "the naming of menu items should not be abstract") is hard, sometimes impossible, to translate into an internal representation.

On the contrary, a substantial advantage results from any representation as a production rule, Horn clause, etc: with adequate tools, it is possible to detect incompleteness, inconsistency, incorrectness, to quasi automatically verify formal properties (for instance, rule reachability, avoidance of redundancy, rule execution uniqueness [64]). We should exploit the wide range of formal methods (as program proof, static analysis, and abstract interpretation) and their software support tools (as automatic or computer-aided theorem provers, syntactic analysers, parsers, automatic generators) that are dedicated to the selected formalism, to ensure the validity of final guidelines.

In conclusion, any subset of guidelines related to a particular problem and operationalised with a software tool for working with guidelines is the most incomplete, potentially incorrect in the sense of reduced expressiveness, but with internal consistency.

6. Usage of guidelines

Today, we know little about software ergonomics as the weight, priority and relevance of guidelines varies and is hard to assess. Similarly, we know little about the adequacy of guidelines, though many authors [3,7,8,13,20,22,26,32] claim their necessity.

We know even less about meta-ergonomics, that is the usability of guidelines themselves during development to ensure UI usability. Refs. [58,68] begin an examination here. We finally know nothing on the actual degradation of guidelines that have passed through all necessary transformations. For instance, what is the impact of replacing a widget selected by a guideline by a more compact widget due to screen constraint? At most, we are able to locate where these degradations occur and to explain them. Thus, all we can do is to cautiously trust in guidelines in current use.

Finally, we have no good knowledge about the usability of software tools for working with guidelines. And yet, it seems fundamental for HCI researchers that the UI of software tools for working with guidelines is usable. An empirical study of the usability of such tools would be of great benefit. The poor status of our knowledge may partially explain why so little comment can be made on the reaching this ultimate milestone in the development of software tools for working with guidelines.

7. Conclusion

After going through the five milestones (from guidelines collection to their usage), we can enumerate all the problems that we encountered:

P_1 = huge number of guidelines;
P_2 = guidelines increasing with time;
P_3 = guidelines dissemination throughout the literature;
P_4 = guidelines variations across contributing disciplines;
P_5 = need to translate guidelines;
P_6 = guidelines variation in validity;
P_7 = decontextualisation of guidelines;
P_8 = guidelines variation in contents;
P_9 = guidelines variation in presentation;
P_{10} = guidelines variation in level of detail;
P_{11} = guidelines variation in complexity;
P_{12} = guidelines variation in importance;
P_{13} = guidelines variation in scope;
P_{14} = guidelines variation in relevant development phase for use;
P_{15} = guidelines variation in target development role;
P_{16} = necessity yet insufficiency of guidelines;
P_{17} = conflict between guidelines;
P_{18} = loss of connections between guidelines;
P_{19} = contextual independence of guidelines;
P_{20} = reductionof guidelines;
P_{21} = independence of guidelines with respect to users;
P_{22} = insufficient guidelines illustration;
P_{23} = insufficient guidelines references;
P_{24} = insufficient guidelines classification;
P_{25} = insufficient operationalisation of guidelines;
P_{26} = multiplicity of bibliographical references for a same guideline;
P_{27} = onerous guidelines management.

We therefore conclude that we have to cope with numerous problems. The general guideline model introduced in Section 3 attempts to solve some of them. Some of them can be solved (for instance, insufficient guidelines illustration and classification), others are intrinsic to guidelines themselves (for instance, their incompleteness) so that it unlikely that they will ever be solved.

As problems P_1 and P_2 are intrinsic, it is not our responsibility to solve them; research for new guidelines should not decrease. It should be promoted along with the expression of results according to a commonly shared model (such as the one introduced in Section 3).

Problems P_3–P_5 are solved by the systematic application of the general guideline model throughout a uniform vocabulary. Guidelines collection into a unique corpus optimises both accessibility and availability.

Problems P_6, P_7, P_{19}–P_{21} concern contextual difficulties associated with guidelines. Where the ergonomic source indicates the experimental context, the values for contextual parameters can be deduced. We have not been able to go further as we felt limited by the lack of interactive task taxonomy. Creating one is a responsibility shared by ergonomists and work psychologists.

Problems P_8–P_{15} were mainly dealt by trying to honestly classify each guideline with respect to related ergonomic design criteria and to a related linguistic level. Knowing this information provides a first entry point to their contents, their priority and to their destination.

Problem P_{16} does not have any solution, it comes with the territory.

Conflict solving for problem P_{17} could be guided by a double comparison between ergonomic criteria and linguistic level. A question arises here: if a first guideline respects a design ergonomic criteria of higher level, but is related to a linguistic level of a lower level than the second, it is not clear which should have the higher priority. Would it be the first guideline because of the ergonomic criteria or the second because of the linguistic level? Although we generally argue that the one with higher ergonomic criteria might be preferred, we think that it depends of the precise nature of the differences. If both guidelines have similar ergonomic criteria, the linguistic level could be preferred. If there is important deviation between ergonomic criteria, these could be preferred instead.

For the P_{18} problem, we proposed to connect guidelines with a link typology. In the corpus of guidelines [10], only the relationship "is neighbour of" was considered since it seemed the most important to us. This is only functionally appropriate in a software tool for working with guidelines, as adding all the other links into a written volume would undesirably increase its size.

About the P_{22} problem on the illustration of guidelines, we suggest that providing **copious examples** taken from the seven most commonly used computing platforms (i.e. Ms-DOS, Macintosh, Windows'95, X-Windows and OSF/Motif, Open Look, Amiga and Open Windows) could be a remedy. The most significant advantage of these examples does not only reside in their illustration and understanding, but also in their ability to promote ideas. This covers a wide spectrum of alternatives from which a starting solution could be extracted without exempting person to think about it. Examples are especially appreciated for visual design and graphical presentation. This illustration partially fills the gap of lack of operationalisation (P_{29}).

Problems P_{23} and P_{26} were dealt with a chronological list of bibliographic references and links to interaction styles and media.

For the P_{24} problem, a multiple level guideline organisation (in division, by presentation/dialogue, by ergonomic criteria, by linguistic level, by interaction style and media) was followed. We recall that any guidelines organisation remains impossible to simultaneously address all demands for all task types. We believe that the adopted organisation could minimise classification confusions and that its structure as a corpus could accommodate honourably most requests. But it cannot be optimal for all demands together. Therefore, it is indispensable to think in the future about flexible organisation that could be tailored within the tool for working with guidelines.

Essential aspects related to problems P_{27} were addressed in Section 5, where software support has been discussed with respect to paper support.

Let us come back to our initial question: to what extent could we trust guidelines that have passed through the five steps, especially the last one? To answer this question, we can now identify in the first four steps the major points where completeness, consistency and correctness are affected. Since the initial guidelines validity was already sub-optimal, all five-transformation steps can only preserve this; but by knowing the locations of this preservation, we can address related problems.

Consequently, the credibility we can attach to final guidelines is function of the scope, the validity of initial guidelines and the preservation of these throughout the diverse transformations. The model developed in this introduction supports assessment of the credibility of guidelines when incorporated into software tools.

Acknowledgements

The author would like to warmly thank the anonymous referees and all people who largely contributed to make this special issue on “Tools for Working with Guidelines” as interesting as possible: Marie-France Barthet (LIHS, Université Toulouse I, France), Andrew Basden (National Industrial Centre for Virtual Environments, Information Technology Institute, University of Salford, UK), Gilbert Cokton, especially (University of Sunderland, UK), Alain Derycke (Laboratoire Trigone, Université des Sciences et Technologies de Lille I, France), Dan Diaper (University of Bournemouth, UK), Peter Gorny (University of Oldenburg, Germany), Monique Noirhomme-Fraiture (Facultés Universitaires Notre-Dame de la Paix, Belgium), Patricia Plénacoste (Laboratoire Trigone, Université des Sciences et Technologies de Lille I, France), Angel Puerta (Stanford University, USA), Jean-Claude Tarby (Laboratoire Trigone, Université des Sciences et Technologies de Lille I, France). The coordination of this special issue was held under the auspices of the Laboratoire Trigone from January 1998 to July 1998 for which the author would like to greatly thank Alain Derycke. Thanks are also due to the contributing authors for submitting their work, promptly taking into account the referees’ comments. Special thanks are also due to Laurence van Ypersele for her constant encouragement.

References

[1] M. Grislin, Ch. Kolski, Evaluation des interfaces homme–machine lors du développement des systèmes interactifs [Evaluation of human–machine interface during the development of interactive systems], Techniques et Sciences Informatiques 15 (3) (1996) 265–296.

[2] J. Nielsen, R. Molich, Heuristic evaluation of user interfaces, Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'90), Seattle, 1–5 April, ACM Press, New York, 1990, pp. 249–256.
[3] Ch. Bastien, D.L. Scapin, Evaluating a user interface with ergonomic criteria, International Journal of Human–Computer Interaction 7 (1995) 105–121.
[4] J. Vanderdonckt, Conception assistée de la présentation d'une interface homme–machine ergonomique pour une application de gestion hautement interactive, PhD Thesis, Facultés Universitaires Notre-Dame de la Paix, Namur, 9 July 1997.
[5] Computer-aided design of user interfaces: J. Vanderdonckt (Ed.), Proceedings of the Second International Workshop on Computer-Aided Design of User Interfaces (CADUI'96), Namur, 5–7 June, Presses Universitaires de Namur, Namur, 1996.
[6] G. Perlman, Asynchronous design/evaluation methods for hypertext technology development, in: Proceedings of ACM Conference on hypertext systems (Hypertext'89), Pittsburgh, 5–8 November, ACM Press, New York, 1989, pp. 61–81.
[7] S.L. Smith, Standards versus guidelines for designing user interface software. Standards versus guidelines for designing user interface software, in: M.G. Helander (Ed.), Handbook of Human–Computer Interaction, Elsevier, Amsterdam, 1988, pp. 877–889 chap. 40.
[8] D.L. Scapin, Guidelines for user interface design: knowledge collection and organization, Technical Report ITHACA.INRIA.89.D12.03, Institut National de Recherche en Informatique et en Automatique, Rocquencourt, 30 December 1989.
[9] Ch. Farenc, Ph. Palanque, J. Vanderdonckt, User interface evaluation: is it ever usable, in: Y. Anzai, K. Ogawa, H. Mori (Eds.), Symbiosis of Human and Artifact: Human and Social Aspects of Human–Computer Interaction, Proceedings of 6th International Conference on Human–Computer Interaction (HCI International'95), Yokohama, 9–14 July 1995, Advances in Human Factors/Ergonomic Series, 20B, Elsevier, Amsterdam, 1995, pp. 329–334 http://www.cenatls.cena.dgac.fr/ ~ palanque/Ps/hciiergo95.ps.gz.
[10] J. Vanderdonckt, Guide Ergonomique des Interfaces Homme–Machine, Presses Universitaires de Namur, Namur, 1994.
[11] K. Nordbyn, P.H. Helmersen, D.J. Gilmore, S.A. Arnesen (Eds.), Proceedings of 5th IFIP TC13 Conference on Human–Computer Interaction (INTERACT'95), Lillehammer, 25–29 June Chapman & Hall, London, 1995.
[12] R.W. Root, A Heads up on GUI Styleguides: Report on the CHI'92 Styleguide SIG, SIGCHI Bulletin 25 (3) (1993) 32–35.
[13] D.J. Mayhew, Principles and Guidelines in Software User Interface Design, Prentice-Hall, Englewood Cliffs, 1992.
[14] S.L. Smith, J.N. Mosier, Design guidelines for the user interface software, Technical Report ESD-TR-86-278 (NTIS No. AD A177198), US Air Force Electronic Systems Division, Hanscom Air Force Base, Massachusetts, 1986.
[15] P. Reed, K. Holdaway, S. Isensee, E. Buie, J. Fox, J. Williams, A. Lund, User interface guidelines and standards: progress, issue, and prospects, Interacting with Computers 12 (2) (1999).
[16] Military Standard: Human Engineering Design Criteria for Military Systems, Equipment, and Facilities, document MIL-STD-1472D, Notice 3, US Department of Defense, 1989.
[17] J.M. O'Hara, Advanced human–system interface design review guideline, General Evaluation Model, Technical Development, and Guideline Description, Document NUREG/CR-5908—BNL-NUREG-52333, vol. 1, US Nuclear Regulatory Commission, Brookhaven National Laboratory, July 1994.
[18] B. Kemp, K. Buckner, A taxonomy of design guidance for hypermedia design, Interacting with Computers 12 (2) (1999).
[19] L. Hardman, B. Sharratt, User-centred hypertext design: the application of HCl design principles and guidelines, in: R. McAleese, C. Green (Eds.), Hypertext: State of the Art, Intellect, 1990, pp. 252–259.
[20] J.D. Gould, How to design usable systems, in: M.G. Helander (Ed.), Handbook of Human–Computer Interaction, Elsevier, Amsterdam, 1988, pp. 757–789 chap. 35.
[21] J. Vanderdonckt, F. Bodart, Jusqu'au bout avec nos règles ergonomiques, in: Actes des Sixièmes Journées sur I'Ingénierie des Interfaces Homme–Machine (IHM'94), Lille, 8–9 December 1994, pp. 231–236 (Accessible at http://www.info.fundp.ac.be/cgi-publi/pub-spec-paper?RP-95-002.)
[22] B. Shneiderman, Designing the user interface: strategies for effective human–computer interaction, 3, Addision-Wesley, Reading, MA, 1997.

[23] J. Carter, Incorporating standards and guidelines in an approach that balances usability concerns for developers and end users, Interacting with Computers 12 (2) (1999).
[24] S. Henninger, A methodology and tools for applying context-specific usability guidelines to interface design, Interacting with Computers 12 (3) (1999).
[25] F. Moussa, C. Kolski, M. Riahi, A model based approach to semi-automated generation for process control interactive applications, Interacting with Computers 12 (3) (1999).
[26] R.M. Schumacher, A. Lund, Development of a usable graphical interface design guide, in: M.J. Smith, G. Salvendy (Eds.), Human–Computer Interaction: Software and Hardware Interfaces, Proceedings of the 5th International Conference on Human–Computer Interaction (HCl International'93), Orlando, 8–13 August 1993, Advances in Human Factors/Ergonomics, 19b, Elsevier, Amsterdam, 1993, pp. 21–26.
[27] Apple Macintosh Human Interface Guidelines, Apple Computer, Inc., Addison-Wesley, Reading, 1992.
[28] Windows Interface: An Application Design Guide, Microsoft Press, Redmond, 1992.
[29] OSF/Motif™ Style Guide, Revision 1.2, Open Software Foundation, Prentice-Hall, Englewood Cliffs, 1992.
[30] D. Grammenos, D. Akoumianakis, C. Stephanidis, Integrated support for working with guidelines: the Sherlock guideline management system, Interacting with Computers 12 (3) (1999).
[31] A. Dillon, Guidelines for screen design, Hypermedia 2 (2) (1990) 171–173 C. Rivlin, R. Lewis, R. Davies Cooper (Eds.).
[32] J.E. McDonald, R.W. Schvaneveldt, The application of user knowledge to interface design, in: R. Guindon (Ed.), Cognitive Science and its Applications for Human-Computer Interaction, Erlbaum (Lawrence), Hillsdale, 1988, pp. 289–338.
[33] L. Pinsky, De l'interface à la situation de travail, in: V. De Keyser, A. Van Daele (Eds.), L'ergonomie de conception, Actes du 23ème Congrès de la Société d'Ergonomie de Langue Française (SELF), De Boeck-Wesmael, Brussels, 1989, pp. 125–132.
[34] G. Perlman, An axiomatic model of information presentation, in: Proceedings of the Human Factors Society—31st Annual Meeting, New York, 19–22 October, Human Factors Society, Santa Monica, 1987, pp. 1229–1233.
[35] J. Vanderdonckt, Accessing Guidelines Information with Sierra, in: [11], pp. 311–316. Accessible at http://www.info.fundp.ac.be/cgi-publi/pub-spec-paper?RP-95-020.
[36] W.K. Horton, Designing and Writing On line Documentation—Help files to Hypertext, Wiley, Chichester, 1990.
[37] S. Ravden, G. Johnson, Evaluating Usability of Human–Computer Interfaces, A practical method, Series in Information Technology, Ellis Horwood, Chichester, 1989.
[38] Object-Oriented Interface Design: IBM Common User Access Guidelines, document SC34-4399, International Business Machines, Publisher Que Corp., 1993.
[39] J. Dumas, Designing User Interfaces for Software, Prentice-Hall, Englewood Cliffs, 1988.
[40] F. Bodart, J. Vanderdonckt, Expressing guidelines into an ergonomical style guide for highly interactive applications, in: Adjunct Proceedings of the Conference on Human Factors in Computing Systems (INTER-CHI'93), Amsterdam, 24–29 April, ACM Press, New York, 1993, pp. 35–36.
[41] E.T. Van Ravenzwaai, The Design of Design Guidelines for Man–Machine Interfaces, in: R. Patesson (Ed.), L'homme et l'écran: aspects de l'ergonomie en informatique, Actes du Colloque L'Ergonomie en Informatique, Nivelles, November 1985, Editions de l'Université de Bruxelles, Bruxelles, 1986, pp. 175–182.
[42] M. Gardiner, B. Christie, Applying Cognitive Psychology to User Interface Design, Wiley, New York, 1987.
[43] W.W. Banks, W.E. Gilmore, H.S. Blackman, D.I. Gertman, Human engineering design considerations for cathode ray tube-generated displays, vol. II, Document NUREG CR-3003/EGG-2230, U.S. Dept. of Energy, Idaho National Engineering Laboratory, Idaho Falls, Idaho, July 1983.
[44] L. Tetzlaff, D.R. Schwartz, The use of guidelines in interface design, in: Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'91), New Orleans, 27 April–2 May, ACM Press, New York, 1991, pp. 329–333.
[45] D.J. Streveler, A.I. Wasserman, Quantitative measures of the spatial properties of screen designs, in: D. Diaper, D. Gilmore, G. Cockton, B. Shackel (Eds.), Proceedings of the Third IFIP TC13 Conference on Human–Computer Interaction (INTERACT'90), Cambridge, 27–31 August, Elsevier, Amsterdam, 1990, pp. 1125–1133.
[46] D.L. Scapin, Guide ergonomique de conception des interfaces homme-ordinateur, Research Report no. 77,

Institut National de Recherche en Informatique et en Automatique, Rocquencourt (Le Chesnay), October 1986.
[47] ISO/WD 9241 Ergonomic requirements for Office Work with Visual Displays Units, International Standard Organization, 1992.
[48] A. Marcus, Graphic Design for Electronic Documents and User Interfaces, ACM Press Books Tutorial Series, New York, 1991.
[49] Ch. Farenc, ERGOVAL: une méthode de structuration des regles ergonomiques permettant I'évaluation automatique d'interfaces graphiques, thèse de doctorat de 3ième cycle, Universitè de Toulouse I, January 1997.
[50] L. Goffinet, M. Noirhomme-Fraiture, Automatic cross-referencing of HCI guidelines by statistical methods, Interacting with Computers 11 (3) (1999).
[51] K.Y. Lim, The Muse Method for Usability Engineering, Cambridge University Press, Cambridge, 1994.
[52] C.B. Kreitzberg, LUCID Software Design FAQ, Cognetics Corp., 1997. Accessible at http://www.cognetics.com/cafe/p_lucidfaq.html.
[53] J. Carter, Combining task analysis with software engineering in a methodology for designing interactive systems, in: J. Karat (Ed.), Taking Software Design Seriously: Practical Techniques for Human–Computer Interaction Design, Academic Press, New York, 1991, pp. 209–234.
[54] F. Bodart, A.-M. Hennebert, J.-M. Leheureux, I. Provot, G. Zucchinette, J. Vanderdonckt, Key activities for a development methodology of interactive applications chap. 7, in: D. Benyon, Ph. Palanque (Eds.), Critical Issues in User Interface Systems Engineering, Springer, Berlin, 1995, pp. 109–134 Accessible at http://www.info.fundp.ac.be/cgi-publi/pub-spec-paper?RP-96-025.
[55] J.C. Tarby, M.F. Barthet, The Diane + Method, in: [5], pp. 95–119.
[56] J.N. Mosier, S.L. Smith, Application of Guidelines for Designing User Interface Software, Behaviour and Information Technology 5 (1) (1986) 39–46.
[57] C. Farenc, V. Liberati, J. Vanderdonckt, Applying ergonomic rules for interactive computer systems, in: A.F. Ozok, G. Salvendy (Eds.), Proceedings of 1st International Conference on Applied Ergonomics (ICAE'96), Istanbul, 21–24 May, USA Publishing, Istanbul–West Lafayette, 1996, pp. 117–122.
[58] F. de Souza, The use of guidelines in menu interface design: evaluation of a draft standard, in: Proceedings of the 3rd IFIP TC13 Conference on Human–Computer Interaction (INTERACT'90), Cambridge, 27–31 August, Elsevier, Amsterdam, 1990, pp. 435–440.
[59] D. Javaux, M.-I. Colard, J. Vanderdonckt, Visual display design: a comparison of two methodologies, in: A.F. Ozok, G. Salvendy (Eds.), Proceedings of 1st International Conference on Applied Ergonomics (ICAE'96), Istanbul, 21–24 May, USA Publishing, Istanbul—West Lafayette, 1996, pp. 662–667.
[60] N. Bevan, Usability is quality of use, in: Y. Anzai, K. Ogawa, H. More (Eds.), Symbiosis of Human and Artifact: Human and Social Aspects of Human–Computer Interaction, Proceedings of 6th International Conference on Human–Computer Interaction (HCI International'95), Yokohama, 9–14 July, Advances in Human Factors/Ergonomics Series, 20b, Elsevier, Amsterdam, 1995, pp. 349–354.
[61] J. Vanderdonckt, F. Bodart, Encapsulating knowledge for intelligent automatic interaction objects selection, in: Proceedings of ACM Conference on Human Factors in Computing Systems (InterCHI'93), Amsterdam, 24–29 April, ACM Press, New York, 1993, pp. 424–429 Accessible at http://www.info.fundp.ac.be/cgi-publi/pub-spec-paper?RP-93-005.
[62] F. Bodart, J. Vanderdonckt, On the problem of selecting interaction objects, in: G. Cockton, S.W. Draper, G.R.S. Weir (Eds.), Proceedings of British Conference on Human-Computer Interaction (HCI'94), Glasgow, 23–26 August, Cambridge University Press, Cambridge, 1994, pp. 163–178 Accessible at http://www.info.fundp.ac.be/cgi-publi/pub-spec-paper?RP-94-018.
[63] F. Bodart, A.-M. Hennebert, J.-M. Leheureux, J. Vanderdonckt, Towards a dynamic strategy for computer-aided visual placement, in: T. Catarci, M.F. Costabile, S. Levialdi, G. Santucci (Eds.), Proceedings of 2nd Workshop on Advanced Visual Interfaces (AVI'94), Bari, 1–4 June, ACM Press, Reading, MA, 1994, pp. 78–87 Accessible at http://www.info.fundp.ac.be/cgi-publi/pub-spec-paper?RP-94-024.
[64] I. Denley, L. Hedman, K. Papadopoulos, B. Hill, A. Clarke, N. Hine, A. Whitefield, Usability principles for service design, in: P. Byerley, Ph. Barnard, J. May (Eds.), Computer, Communication and Usability: Design Issues, Research and Methods, Elsevier, Amsterdam, 1993.
[65] H. Fritzsche, T. Michel, Formalization and proof of design guidelines within the scope of testing formally specified electronic product catalogues, Interacting with Computers 12 (3) (1999).

[66] J.A. Fox, S.L. Smith, Dynamic rules for user interface design, Document M89-22, The MITRE Corp., Bedford, 1989.
[67] L. Alben, J. Faris, H. Saddler, Making It Macintosh: Designing the message when the message is design, Interactions 1 (1) (1994) 10–19.
[68] H. Thovtrup, J. Nielsen, Assessing the usability of a user interface standard, in: Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'91), New Orleans, 27 April–2 May, ACM Press, New York, 1991, pp. 335–341.