



**HAL**  
open science

## A Framework for Organizing Web Usability Guidelines

Dominique Scapin, Corinne Leulier, Jean Vanderdonckt, Céline Mariage,  
Christian Bastien, Christelle Farenc, Philippe Palanque, Rémi Bastide

► **To cite this version:**

Dominique Scapin, Corinne Leulier, Jean Vanderdonckt, Céline Mariage, Christian Bastien, et al.. A Framework for Organizing Web Usability Guidelines. 6th Conference on Human Factors and the Web (HF Web 2000), Jun 2000, Austin, United States. hal-01319628

**HAL Id: hal-01319628**

**<https://hal.science/hal-01319628>**

Submitted on 24 May 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Framework for Organizing Web Usability Guidelines

**Dominique Scapin and Corinne Leulier**

[Dominique.Scapin@inria.fr](mailto:Dominique.Scapin@inria.fr), [Corinne.Leulier@inria.fr](mailto:Corinne.Leulier@inria.fr)

Institut National de Recherche en Informatique et en Automatique

Unité de recherche Rocquencourt, Domaine de Voluceau – B.P. 105, F-78153 Le Chesnay Cedex, France

Web: <http://www.inria.fr/Dominique.Scapin>

**Jean Vanderdonckt and Céline Mariage**

[vanderdonckt@qant.ucl.ac.be](mailto:vanderdonckt@qant.ucl.ac.be), [mariage@qant.ucl.ac.be](mailto:mariage@qant.ucl.ac.be)

Université catholique de Louvain (UCL), Place des Doyens, 1 – B-1348 Louvain-la-Neuve, Belgium

Web: <http://www.qant.ucl.ac.be/membres/jv>

**Christian Bastien**

[Christian.Bastien@ergo-info.univ-paris5.fr](mailto:Christian.Bastien@ergo-info.univ-paris5.fr)

Laboratoire d'Ergonomie Informatique, Université René Descartes, 45, rue des Saints-Pères – F-75270 Paris Cedex 06, France

Web: <http://www.univ-paris5.fr/LEI>

**Christelle Farenc, Philippe Palanque, and Rémi Bastide**

[Farenc@univ-tlse1.fr](mailto:Farenc@univ-tlse1.fr), [Philippe.Palanque@univ-tlse1.fr](mailto:Philippe.Palanque@univ-tlse1.fr), [Remi.Bastide@univ-tlse1.fr](mailto:Remi.Bastide@univ-tlse1.fr)

Université Toulouse I, UFR d'Informatique, Laboratoire d'Interaction Homme-Système (L.I.H.S), Place Anatole France – F-31042 Toulouse, France

Web: <http://lis.univ-tlse1.fr/~farenc/>, <http://lis.univ-tlse1.fr/palanque/>, <http://lis.univ-tlse1.fr/bastide/>

## Abstract

If web usability guidelines can play an important role for ensuring usability when designing and evaluating web sites, the moment and the way to use them still remain unclear. To address this problem, one goal of the [EvalWeb](#) research project was to come up with a general framework of web usability guidelines helping people to structure the way they are working with such guidelines.

## Introduction

Usability required for use of web sites is today widely recognized as an important requirement for user acceptance. This requirement becomes more critical for web sites where concurrence is hard. For instance, for electronic commerce web sites: a customer unsatisfied by poor usability is likely to become a competitor's

customer. On the other hand, the user population is expanding in age (ranging from young users to elderly people), in expectations (ranging from private use for leisure to professional use), in information needs (ranging from simple information to compound multimedia resources), in task types (ranging from basic text searches to complex problem-solving methods), and in user abilities (ranging from the able-bodied person to any person with special needs, such as for motor-, auditory- or visually-impaired persons). Web sites should accommodate all these variations over time.

Appropriate web site design and evaluation methods help ensure that web sites are usable ([Nielsen & Mack, 1994](#)), but they are so numerous and hard to differentiate that identifying which ones are suitable for a particular web site is challenging. This identification is practically constrained by several needs, namely:

- The need for a cost-effective method: project resources are so limited in time, budget, and personal that they prohibit any use of evaluation method ([Lynch, Palmiter & Tilt, 1998](#)).
- The need for remote evaluation: to reach a wider audience and to cover multiple evaluation circumstances, evaluators may be separated in time and/or space from the users.
- The need for challenging designs and evaluations ([Cooper, 1999](#)), such as very large web sites and dynamic web sites: very large web sites are characterized by being any large-scale, information abundant, interactively rich web site installed in a distributed environment (e.g., in different physical locations) with heterogeneous software and hardware (e.g., on different servers).
- The need for real-time evaluation: when a web site is rapidly changing over time, the organization often wants to keep the evaluation accurate and up to date with respect to the current state of the web site. In this case, performing large-scale or expensive evaluations is prohibited and incompatible with the rapidly changing nature.
- The need for raising the usability awareness of designers and evaluators ([Scholtz, Laskowski & Downey, 1998](#)): teaching usability to designers who are unaware of it as they use it or improving the experience of those who already are aware of it can be a very valuable process; their usability awareness and their skills to conduct it are likely to grow as their experience is accumulated.
- The need for reusable evaluation results: an evaluation method should produce tangible results that should be interpreted, processed in a straightforward manner so as to reuse them when the web site is updated or improved. If an evaluation method produces results that designers cannot practically consider or interpret, the method is doomed to failure. For this purpose, concrete guidelines should be preferred to general guidelines.

To address these needs, any design and evaluation method that is easy to use, cheap to manipulate, and quick to conduct is often preferred. Moreover, rapid lightweight methods partially or completely supported by software tools are often considered relevant.

For these reasons and others, the EvalWeb project decided to focus on the **use of web usability guidelines in heuristic evaluation**. We hereby define a *web usability guideline* as any statement ensuring some adequacy of a particular user interface of a web site with respect to a particular context of use where a given user population has to fulfill interactive tasks with a given system. On the one hand, guidelines have become a popular form for conveying some usability wisdom that can be expressed, communicated, and propagated effectively and

efficiently. On the other hand, a profusion of sources containing guidelines is available. At first glance, these resources can be considered able to provide a valuable starting point for conducting heuristic evaluation.

The application of such guidelines has already been proved to have a positive impact on usability ([Comber, 1995](#); [Grose, Forsythe & Ratner, 1998](#)). For instance, Borges *et al.* showed that the average time to carry out 5 tasks on a web site respecting 17 guidelines has been reduced by at least by 16% ([Borges, Morales & Rodríguez, 1996](#)). But this does not necessarily imply that a web site that does not consider guidelines is unusable. Nor it is proved that a web site addressing all guidelines is the most usable site. In other words, guidelines are necessary, but insufficient. They should not be considered in isolation: often guidelines need to be supplemented by a suitable method and a clear process that leads them to unambiguous interpretation.

Although guidelines can be used in conjunction with guidelines checklists, standard inspection, and consistency inspection, a variant of heuristic evaluation has been set up where the usual set of 10 heuristics ([Nielsen & Mack, 1994](#)) is replaced by a set of specific guidelines. This replacement came from the observation of heuristics which, albeit cost-effective, have been considered too general and not expressive enough to be interpreted and applied effectively. Conversely, lists of guidelines are potentially long. If they are not specific, a lot of time to interpret them is likely to be devoted since the whole user interface and its components should be checked against them. One major obstacle for using such guidelines lies in their difficulty to properly interpret them according to the context and to know when and how they can be used. To overcome these shortcomings, the main goal in this work is consequently to define a **framework that facilitates the structuring and the use of guidelines for design and evaluation of web sites**.

The rest of this paper is structured as follows: the next section reports on the knowledge contained in web usability guidelines, where they come from, what are some of their shortcomings. Due to some of their shortcomings, they will be augmented by usability guidelines that are manipulated for traditional graphical user interfaces. Then, the different steps of the method used to obtain this framework, their results and their rationales are described. This framework for organizing web usability guidelines will then be illustrated by some examples. A conclusion will outline several dimensions towards which this framework could evolve.

## Web Usability Guidelines

In this section, possible sources for collecting web usability guidelines are first indicated. Second, some shortcomings of guideline usage are discussed as they mostly affect the results produced by their exploitation. Their level of expressiveness is one major cause of concern.

### **Sources for Web Usability Guidelines**

Many guidelines have been published in document sources throughout the literature. These sources fall into five categories ([Bastien & Scapin, 1995](#); [Vanderdonckt, 1999](#)):

1. *Design rules*: they usually consist in a set of functional and/or operational specifications that clarify the design of a particular user interface. These specifications are presented in a form that should not require any further interpretation, either from designers or from developers. Their straightforward format allows an immediate exploitation (Fig. 1).
2. *Ergonomic algorithms*: such algorithms typically aggregate single design rules into a comprehensive and systematic procedure that can be applied more quickly than a series of single guidelines. In this way, they

introduce more flexibility by enabling designers to select parameters required for design rules and they prevent designers to inadvertently forget some design rules. They more usually appear as a software component that implements an algorithm rather than a paper procedure. Such algorithms are primarily intended to design a series of web pages that automatically respect guidelines by construction.

3. *Style guides*: they comprise a set of guidelines and/or functional or non-functional specifications aiming at consistency for a collection of distinct user interfaces. This collection can be based on an operating system, on a software editor, by a particular physical environment such as SUN ([Levine, 1996](#)), by a domain of human activity or by a corporate. In particular, Ratner, Grose & Forsythe ([1996](#)) identified that only 20% of recommendations from established style guides that apply to the web were found in web style guides. They also found out that little consistency existed among the 21 considered web style guides, with 75% of recommendations appearing in only one style guide.
4. *Compilations of guidelines*: they comprise several prescriptions written for a wide range of user interfaces. 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 among guideline users. This consensus is less relevant once a prescription is experimentally tested and verified. The scope of the compilations can range from a small set of guidelines dedicated to a particular usability feature (e.g., an interaction technique) to an extensive collection of guidelines covering a family of tasks and domains. Such guidelines are appropriate for multimedia applications, for interactive kiosks, to ensure accessibility ([Vanderheiden et al., 1997](#)), to design web site in general ([Detweiler & Omanson, 1996](#)). Some guidelines are validated by experimental results provided by user testing, laboratory experiences or other techniques, while others are not.
5. *Standards*: they comprise a set of functional and/or operational specifications intended to standardize design. Standards are promulgated by national or international organizations for standardization. They can be military, governmental, civil or industrial. There is no web dedicated standard we are aware of, but some important general standards are good candidates to tailor their guidelines to suit the exact issues faced by web sites, e.g., ISO 9241 ([ISO, 1999](#)), HFES/ANSI 200 ([HFES, 1997](#)).

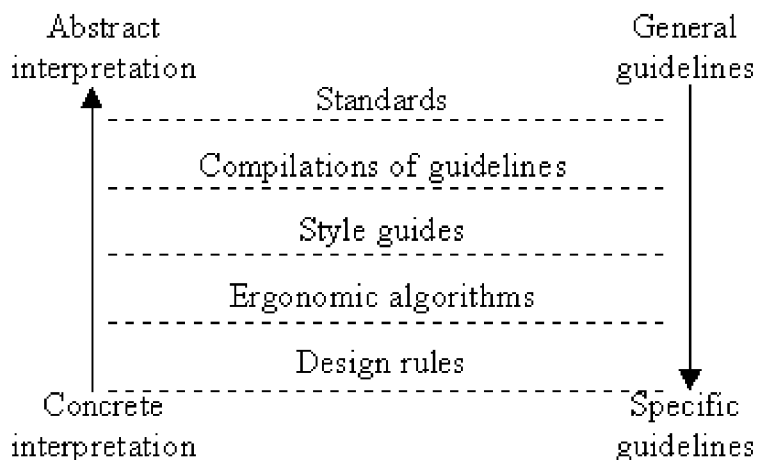
- ☞ Each page should be terminated by the name of the person who is responsible for the information contained in the page and by a clickable e-mail of the person who authored, designed the page.
- ☞ Each page should contain a KEYWORDS meta-tag.
- ☞ The navigation toolbar should reflect the position of the page in the structure.

**Fig. 1.** Examples of design rules.

## ***Some shortcomings of guidelines***

Despite the fact that usability guidelines have been proved useful as indicated, they still suffer from a series of shortcomings that impede their use and significantly reduce their scope. We focus here on only some of these shortcomings that are relevant for web sites as an extensive discussion of these shortcomings can be found elsewhere ([Nielsen & Mack, 1994](#); [Bastien & Scapin, 1995](#); [Vanderdonckt, 1999](#)).

First, the level of guideline expressiveness and the confidence in applying guidelines heavily depends on the source where the guidelines come from. Fig. 2 depicts that guidelines found in the five types of sources range from general guidelines requiring an abstract interpretation to a specific guideline only needed concrete interpretation. The more a guideline is general, the more their applicability domain is wide and the more their interpretation becomes abstract. For example, a guideline stating which widget to select for a particular task in a given computing platform (this type of guideline usually appears in a style guide), may be regarded as more specific than a guideline stating that a usable user interface objects should be selected for each domain object (this type of guideline usually appears in a standard). Moreover, this last standard guideline can be considered in turn more general than a design rule specifying that an edit box should be selected to enable users to enter their search query. To interpret this last design rule and to apply it, one only needs to know what concrete interactor could be used for this purpose. Conversely, to interpret the standard guideline and to apply it, one needs to determine a specific domain object, to identify a set of candidate interactors, and to define criteria allowing the selection of an interactor in the set of candidates (e.g., user preference, performance, legibility). Although the five types of sources contain multiple types of guidelines, we believe that they often contain guidelines requiring some interpretation at a certain level of abstraction (Fig. 2).



**Fig. 2.** Location of sources containing guidelines.

Second, as a consequence of the previous observation, general guidelines cannot be applied per se, thus requiring some concrete interpretation for the intended context of use. On one hand, the format of general guidelines can withdraw experimental conditions under which the guideline has been tested and validated. On the other hand, the lack of these conditions, which are required to ensure a correct interpretation, may invalidate any such interpretation. Specific guidelines no longer require such interpretation, but are so specific that they prevent designers to apply them in other situations without any risk of invalidity. General guidelines are difficult to interpret when and how they need to be applied at design time or evaluated at execution time.

Third, the jargon or specific vocabularies used in the sources coming from various disciplines (e.g., cognitive modeling, psychology, human factors, ethnography) may also prevent designers to easily understand them and applying them correctly. Once again, extensive relevant experience may be needed to avoid incorrect generalization or specialization of guidelines that will invalidate their results.

Fourth, guidelines can be classified according to the linguistic level of interaction model ([Nielsen, 1986](#)), i.e. goal, pragmatic, semantic, syntactic, lexical, alphabetical, and physical. Guidelines located at the lower levels of this layered model tend to be easier to interpret and to apply than those located at the higher levels.

Fifth, the workload involved by considering a single guideline depends on several guideline properties namely, but not limited to: its linguistic level, the quality of its statement, and their scope. A guideline can be relevant for one individual page at a time in a specific context of use (Fig. 3a), for a series of related web pages (Fig. 3b), and for the whole site (Fig. 3c).

- ☞ Since most viewers will use 13" color monitors, make sure your web page fit comfortably within the parameters of these smaller displays ([Detweiler & Omanson, 1996](#)).
- ☞ Give users control to specify the extent of their searches, e.g., locally within a section or subsection ([Detweiler & Omanson, 1996](#)).
- ☞ Help orient users by establishing useful expectations about the site's organization and/or functionality ([Detweiler & Omanson, 1996](#)).

**Fig. 3.** Scope of different guidelines (a,b,c).

## Method for building the framework

The impact of the above shortcomings is varying according the goals for which guidelines are considered. If developing a tool for automated or computer-aided evaluation of these guidelines is the ultimate goal, then these shortcomings are important and hard to solve. If providing people with assistance and guidance in applying and evaluating guidelines is the ultimate goal, then these shortcomings are less important. To address the above shortcomings, the need for organizing guidelines into a practical framework that would in turn facilitate the structuring and use of guidelines rapidly emerged. The steps of the method use to build this framework are: guidelines collection, guidelines organization, and the incorporation of guidelines into approach ([Vanderdonckt, 1999](#)).

### **Guidelines collection**

This step involves two consecutive activities: web guidelines were first collected and then augmented with guidelines for traditional graphical user interfaces. The rationale behind this augmentation is briefly discussed.

### **Web guidelines collection**

First, web-related only guidelines have been collected from existing sources ([Leulier et al., 1998](#)). Collected guidelines had different origins. Some guidelines came from Web design guides: the "Apple Web Design Guide", the "IBM Web Guidelines" ([IBM Corporation, 1997](#)), the Sun Microsystems "Guide to Web Style" ([Levine, 1996](#)), the guidelines compilation of the "Australian Department of Primary Industries and Energy" ([DPIE, 1997](#)), the "Ameritech Web Page User Interface standards and guidelines" ([Detweiler and Omanson, 1996](#)), and the "Yale C/AIM WWW Style Manual" ([Yale, 1997](#)).

Some recommendations were found in scientific papers (e.g., [Borges et al., 1996](#); [Hardman and Sharrat, 1989](#); [Levi and Conrad, 1996](#), [Vanderheiden et al., 1997](#)); some other in position papers (Nielsen's "[alertboxes](#)"), or in reports (e.g., [Balasubramanian and Turoff, 1993](#)). Each document was consulted to extract guidelines with the aim of selecting the appropriate ones. During that stage, some recommendations were not selected:

- Guidelines that were too much related to HTML. For instance, "Verify your HTML syntax and construction" ([Levine, 1996](#)).

- Guidelines related to the "netiquette", i.e., advices or conventions related to "good manners" on the Internet. For instance, "Respond to people who comment on your pages", "Don't insult or flame people", "Don't publish copyrighted material without permission of the owner" ([Levine, 1996](#)).
- Guidelines that were too tightly related to a particular type of sites. For instance, some guidelines concerning web sites of universities or other institutions like "Descriptive information about the institution should be placed just below the header and kept to a minimum (a link to a secondary page is preferred)" ([Borges, et al., 1996](#)).

Contrarily to the domain of usability for graphical user interfaces, web usability guidelines often appeared to be produced by consensus, common sense or observation of best practices. Guidelines validated by experimental results are rare and hard to find out. Moreover, web usability guidelines tend to address web specific usability issues, but not necessarily usability issues for graphical user interfaces adapted or specialized for the web. These observations also formed one of our motivations: usability guidelines that are valid for graphical user interfaces could also be considered as potential usability guidelines for the web, unless some contradiction or invalidation appears. Ratner, Grose & Forsythe ([1996](#)) provided ample support for this conclusion as they identified the lack of integration of guidelines for graphical user interfaces into web guidelines. Other reasons for transferring guidelines for graphical user interfaces to the web are discussed in ([Scapin et al., 2000](#)).

Only categories of rigorous sources providing experimentally validated guidelines are considered the most reliable.

## **Adding usability guidelines for traditional graphical user interfaces**

The goal of this second activity is to gather a useful subset of guidelines suitable for designing web pages. In this study, three sources containing guidelines have been selected for inclusion ([Mayhew, 1992](#); [ISO, 1999](#); [Vanderdonck, 1994](#)) as they consist in a compilation of guidelines (respectively, 293, 488, more than 3,000!). In the ISO general standard, only Part 12 "Presentation of Information", Part 13 "User guidance", Part 14 "Menu dialogues", Part 15 "Command dialogues", Part 16 "Direct manipulation dialogues", and Part 17 "Form-filling dialogues" have been selected because the other parts have been considered too general. These different sources are not especially dedicated to web usability. Rather, they focus more on graphical user interfaces in general, thus requiring some modification, adaptation, extension, and so forth ([Ratner, Grose & Forsythe, 1996](#)). However, a closer examination of guidelines contained in these three sources revealed a high amount of guidelines and an important redundancy among them. For these reasons, the ISO standard has been retained first. A complementary and future work will consist in incorporating non-redundant guidelines from the other two sources. It turned out that most guidelines that are applicable to graphical user interfaces are still applicable to the web, although experiments are still needed to make them valid. Four hundred guidelines out of 466 guidelines resulted from this activity and were merged with the guidelines resulting from the first activity.

## **Guidelines organization**

This initial set of guidelines is copious and ranges over many levels of rigor and credibility. Both can be ameliorated by a good organizational structure, which is the main goal of guidelines organization. The guidelines organization basically involves two activities:

### **Classifying each guideline by ergonomic criteria.**

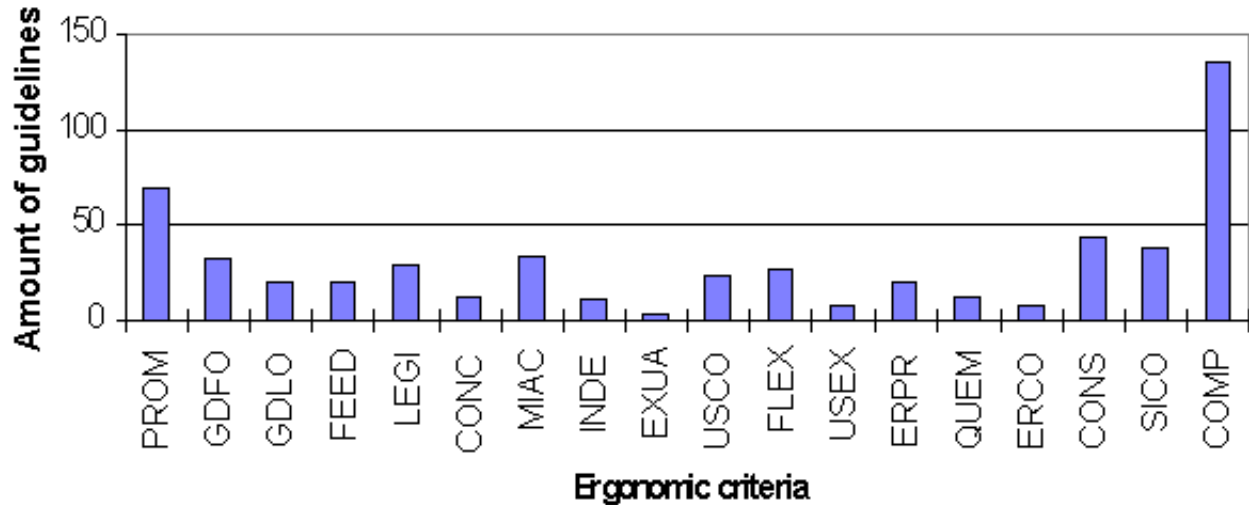


An *ergonomic criterion* is hereby defined as a well-recognized usability dimension in human-computer interaction whose reliability effectiveness and impact on usability have been experimentally assessed ([Bastien & Scapin, 1995](#); [Scapin & Bastien, 1997](#)). These criteria have been successfully used for classifying web usability guidelines ([Bastien, Scapin & Leulier, 1999](#)). Table 1 reproduces these ergonomic criteria. Leaf nodes in the decomposition are called *elementary criteria* as they are not further decomposed into sub-criteria. Other criteria that can be decomposed into other sub-criteria are called *composite criteria*. Each guideline has therefore been assigned to a unique ergonomic criterion based on its definition. For this purpose, guidelines have been submitted to operations like unfolding and coupling into two, three or even four parts. Indeed, the statement of some guidelines remained too general or was condensing several guidelines at the same time. Therefore, to individualize each guideline, these operations have been performed and assessed by a panel of experts. Some guidelines have been affected to some criteria by consensus between the judging parties. Resulting from these activities, the guidelines pool has been augmented by 74 individual guidelines, thus totalizing 474 guidelines ([Scapin et. al., 1999](#)). Fig. 4 graphically depicts the distribution of resulting guidelines by elementary criteria. It is worth noting here that the compatibility criterion received the most important attention in having the largest number of resulting guidelines. The “Prompting” and the “Compatibility” criteria contain, respectively, 69 and 134 individual guidelines, whereas “Explicit actions” and “User experience” only have, respectively, 3 and 7 guidelines. This does not necessarily mean that these criteria are unimportant for designing and/or evaluating web sites, but that they are understated or underrepresented in the guidelines set. Classifying each guideline by ergonomic criteria goes beyond the simplicity of indexing: it can provide designers with a first idea on when and where the related guideline can be applied as well as a first idea of their absolute level of importance. For instance, a guideline related to task compatibility may be considered more important than a guideline related to grouping items by format.

1. Guidance
  - 1.1. Prompting (PROM)
  - 1.2. Grouping / Distinction between items
    - 1.2.1. Grouping / Distinction by location (GDLO)
    - 1.2.2. Grouping / Distinction by format (GDFO)
  - 1.3. Immediate feedback (FEED)
  - 1.4. Legibility (LEGY)
2. Work load
  - 2.1. Brevity
    - 2.1.1. Concision (CONC)
    - 2.1.2. Minimal actions (MIAC)
  - 2.2. Information density (INDE)
3. Explicit control
  - 3.1. Explicit actions (EXUA)
  - 3.2. User control (USCO)
4. Adaptability
  - 4.1. Flexibility (FLEX)
  - 4.2. User experience (USEX)
5. Error management
  - 5.1. Protection to errors (ERPR)
  - 5.2. Quality of error messages (QUEM)
  - 5.3. Error correction (ERCO)
6. Homogeneousness / Consistency (CONS)

- 7. Significance of codes (SICO)
- 8. Compatibility (COMP)

**Table 1.** Hierarchy of ergonomic criteria and sub-criteria.



**Fig. 4.** Distribution of guidelines by elementary ergonomic criteria.

**Further classifying each guideline**

As the resulting set of guidelines is still huge, guidelines are further classified by alternate index keys. This classification allows multiple and flexible access paths to each guideline, rather than merely by ergonomic criteria (Bastien, Scapin & Leulier, 1999). Such accesses permit automatic identification of guidelines for a particular page element (e.g., for an image map), for a particular design aspect (e.g., information layout), or any combination (e.g., how to lay out information in an image map). Such identifications are required for specific evaluation targets and/or to make guidelines directly linked to design options, making the design/evaluation process more constructive towards the ends of designers. These identifications also improve the access efficiency to guidelines: when a given page element is modified across a set of pages, corresponding guidelines can be evaluated more rapidly as they are specific to the context. Each guideline is assigned to one or many index keys relevant for web sites (Table 2). These index keys have been deduced from the guidelines statements and organized into a hierarchy.

Actions (Seq. of)	Image maps/hot areas	Plug-in
Alignment	Image maps/linked items	Printing
Animations/Video	Images/description	Radio buttons
Buttons/Labels	Information Layout	Reading task
Buttons/Position	Information/Content	Screen Layout
Check Boxes	Information/Organisation	Scrolling
Colours	Items organisation	Search Facilities
Colours/Black & White displays	Items/Presentation	Site maps
Downloading considerations	Language	Sounds
Errors	Lines	Tables
	Links	Terms
	Links/Organisation	Terms/Graphics

Forms	Links/visual codes	Text
Forms/Buttons	List	Text & background
Frames	List/items	Text/Buttons
Graphics	List/Structure	Text anchors
Graphics/Description	Menus	Text colours
Graphics/Downloading	Menus/Scrolling lists	Text fields
Graphics/Layout	Menu/selected items	Text-only Browsers
Graphics/Size	Metaphors	Text-only
Graphics/Text	Navigation structure	Browsers/Lines
Grids	Navigational aids	Text/Headers & footers
Headings	Navigational aids/Links	Title
Help Facilities	Page Length	URL
Helper Applications	Page Length/Links	Vocabulary
Icons	Page size	Windows
Icons/Image maps	Page structure	
Icons/Labels		
Image maps		

**Table 2.** Taxonomy of additional index keys.

Each guideline is then assigned to one or many methods and techniques for web site design/evaluation (e.g., user questionnaire, user testing, user observation, cognitive walkthrough, card sorting) that can be effectively used to assess the guideline. In particular, each guideline is examined to identify the extent to which its application can be software supported, preferably with automation. Five levels of automation are distinguished: *full automatic* (when the guideline can be evaluated by the system), *semi-automatic* (when the guideline can be evaluated partly by the system), *questions-based* (when the system is asking questions to the evaluator to achieve the evaluation), *delegated* (when external methods or tools can provide evaluation support), or *manual* (when the guideline can be evaluated by the user only). Generalization/specialization relations are then exploited to establish more precise links between related guidelines. Finally, a score is attached to each guideline to express the impact the guideline can cause when it is respected or violated (Fig. 5).

☞ **Guideline:** In order to support different levels of user skills, direct manipulation dialogues should be designed to minimize the need for users to alternate between different input devices.

**Example:** To fill in a form a user points and selects every field with the mouse and then enters text with the keyboard. As the experience grows the user moves the cursor from field to field with the tab key before entering text. Thereby the need for the user to alternate between input devices is minimized and efficiency is increased.

**Source:** Ergonomic requirements for office work with visual terminals (VDTs) - Part 16: Direct-manipulation dialogues, ISO/DIS 9241-16.

**Guideline Number:** (5.4.3)

**Ergonomic criteria:** Users' experience

**Index Key:** Dialogue

**Evaluation method:** user observation (Level of automation: manual)

**Score:** rather important.

**Fig. 5.** Example of resulting guideline.

## Incorporation of guidelines into approach

Early consideration of web usability guidelines in the development life cycle of the web site requires an iterative and eager design process. It is thus important to associate sections of guidelines with the different phases of a development life cycle. In this way, it is expected that sets of guidelines suitable for each phase can be more easily identified and accessed. The goal of this step is to locate points within these phases where organized 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). For this purpose, the general development life cycle illustrated in Fig. 6 was used. The work of referring each guideline to phases and sub-phases according to a development life cycle is still an ongoing research. Our method is introduced and justified in ([Scapin et al., 2000](#)).

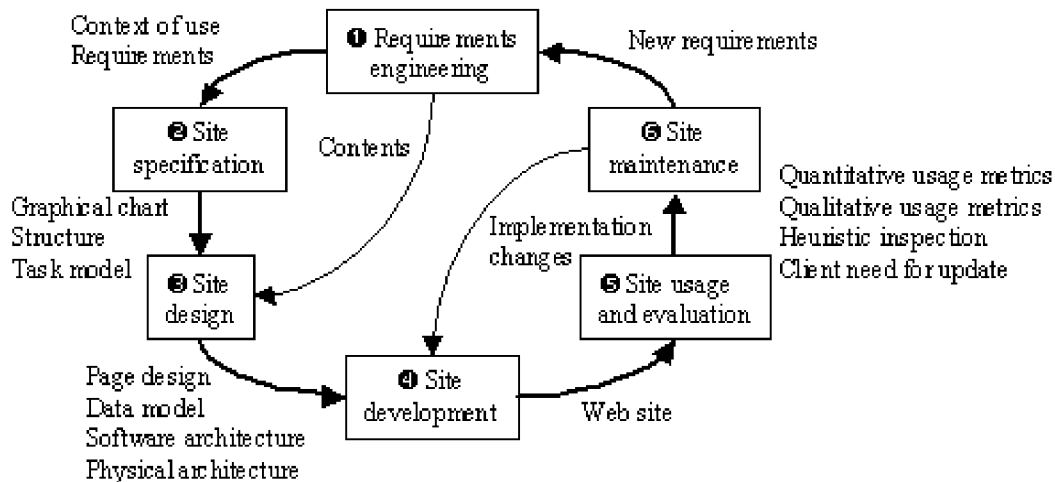


Fig. 6. General development life cycle for a web site.

## Some examples of web usability guidelines organized according to the framework

In this section, some examples will illustrate how the obtained framework can be used to structure guidelines through design and evaluation. Some methods helping people to apply these guidelines are exemplified as well.

### Example #1

Let us imagine two design rules for the inclusion of a particular graphic:

- ☞ Each personal home page should contain the company logo at the top left corner.
- ☞ Each personal home page should contain the photo of the person after the logo.

The first guideline is related to two aspects: the existence of the company logo, which is relevant to the “Prompting” ergonomic criteria (PROM), and its positioning, which is relevant to the “Grouping/Distinction by location” ergonomic criteria (GDLO). According to the framework, the first guideline could be organized as follows:

#### 2.1.2.5. Specification C.E.-Guidance

2.1.2.5.1.3.2.4 Picture: picture=company logo /\*First ergonomic criteria to be respected\*/

2.1.2.5.2 Specification C.E.-Grouping/distinction

2.1.2.5.2.1 Spec. C.E.-By location: location=left corner /\*Second ergonomic criteria to be respected\*/

The second can be formalized similarly. Capture-and-Replay testing tools (CR tools) represent some first attempt to automate compliance with design rules by performing four steps ([Linz & Daigl, 1998](#)):

1. *Capturing*: the tool captures static properties of objects in page (e.g., color, size) as well as dynamic properties (e.g., activation, deactivation).
2. *Programming*: the properties are stored in a test script to be programmed with traditional algorithmic structures.
3. *Inserting checkpoints*: checkpoints regarding properties are inserted into the script.
4. *Replaying*: the test script is run on any page to be evaluated and any discrepancy with respect to reference properties and checkpoints is detected.

Several design rules can be tested this way, such as “Ok and Cancel are always located at the bottom or on the right, but never on top or on the left”, “Submit and Reset push buttons in forms, if any, should be presented in this order”. Fig. 7 reproduces a screen capture where the two above design rules have been captured and subsequently replayed with Watchfire’s MacroBot (<http://www.watchfire.com/products/macrobot>). It basically consists of two test cases testing the presence of the graphics at their correct location, plus the general composition. In this case, the two guidelines are not respected. The level of automation is considered full automatic in this case.

CR tools are dependent on the very specific parameters of a guideline. In the recording process, it recorded the index position of an image the user pointed to in the images array rather than more uniquely identifying information such as its source. Adding or removing a single image to the page would completely disrupt the test. Direct creation of JavaScript representations of guidelines instead of macro recording could help avoid some of this, but the definition of guidelines in this tool is highly dependent on a document's particular structure. A more generic way of expressing guidelines that depends, for example, on an object's resultant position on the screen rather than its position in the document structure should be preferred.

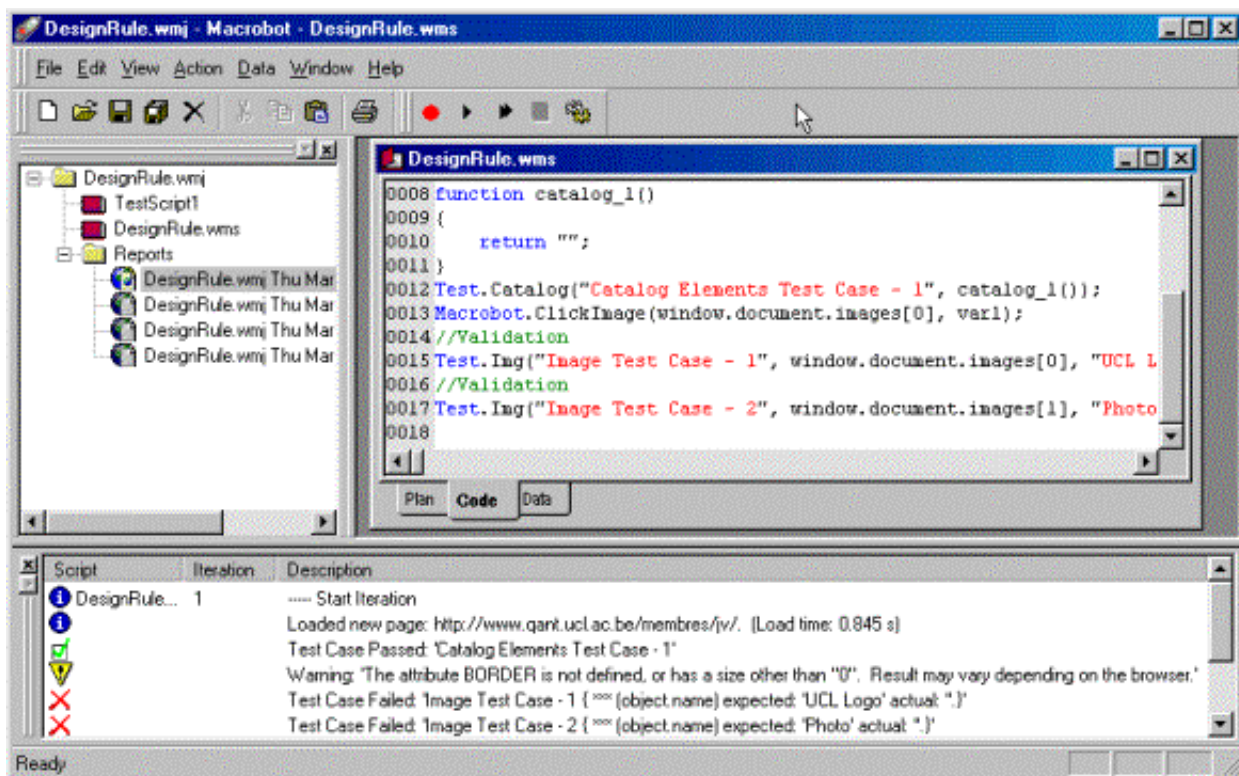


Fig. 7. Replaying a test script for two design rules.

## Example #2

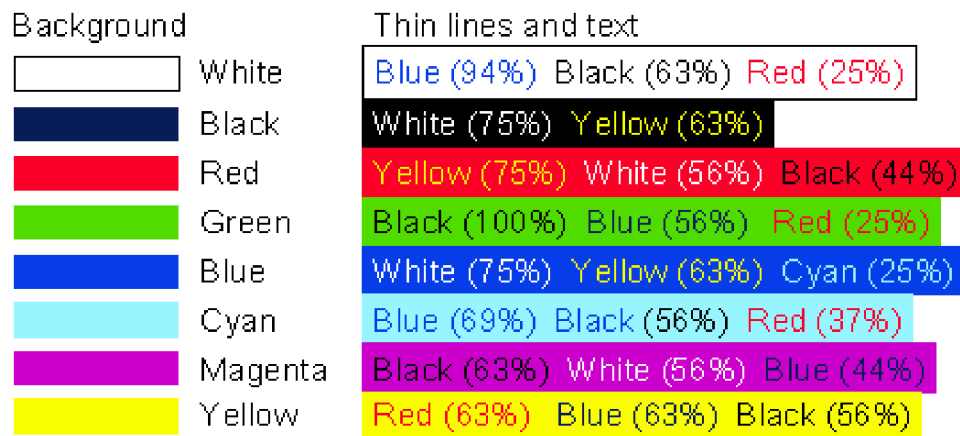
Let us consider now a guideline extracted from a guidelines compilation

- Select colors that will make your page easy to read by people with color blindness. One good test is to see if your page is readable in back and white ([Vanderheiden et al., 1997](#)).

This guideline is a typical one that can be used in requirements engineering (step 1 in Fig. 6), in site specification (step 2 in Fig. 6), and in site evaluation (step 3 in Fig. 6). According to the framework, this guideline could be formalized as follows:

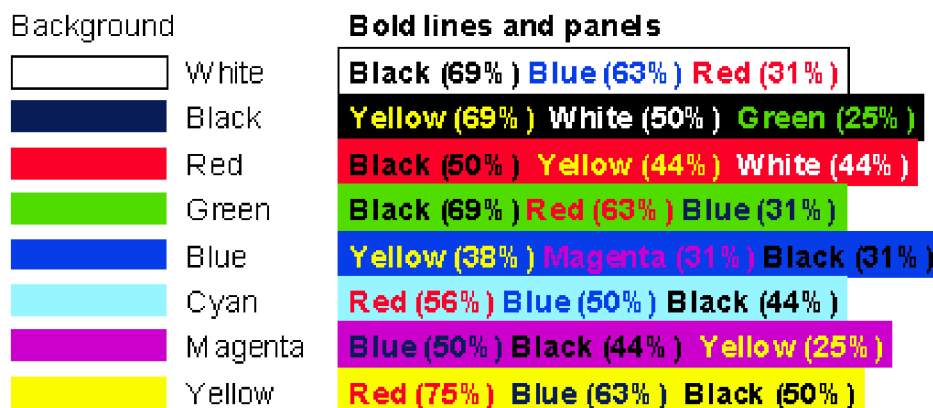
- Requirements engineering
  - Users
    - Handicaps: impairment = color blindness
- Site specification
  - Spec. C.E.-Compatibility
    - Accessibility: user limitation = lack of color differentiation
    - Colors: colors = readable by users
- Site evaluation
  - Color scheme = readable by users; legible in black & white

This guideline, as it is represented, cannot be automated in a straightforward manner. However, if we refer to the research conducted by Murch ([Murch, 1987](#)), a restriction of this guideline can be expressed as: the combination between the background color and the foreground color should belong to the best color combination or should not belong to the worst color combinations. Fig. 8 represents good color combinations for thin lines and text. It can be read as follows: for thin lines and text displayed on a white background, blue is a good choice in more or less 94% of cases, black is in 63% of cases, and red is in 25% of cases (Murch's research is based on legibility tests for user acceptance of color combinations). For thin lines and text displayed on a black background, white is a good choice in more or less 75% of cases, yellow is in 63% of cases. The other lines can be read similarly.



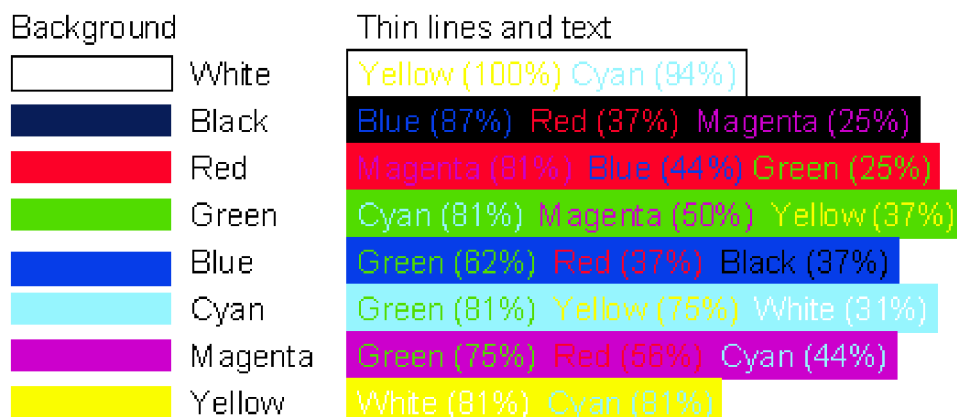
**Fig. 8.** Good color combinations for thin lines and text by order of acceptance.

Fig. 9 represents the good color combinations for bold lines and panels. It can be read as follows: for bold lines and panels displayed on a white background, black is a good choice in more or less 69% of cases, blue is in 63% of cases, and red is in 31% of cases. We observe that usable color combinations also depend of the text style or area. While identifying bold text is easy thanks to the `<B> bold text </B>` tag, identifying bold lines and panels remains more challenging to do automatically.



**Fig. 9.** Good color combinations for bold lines and panels by order of acceptance.

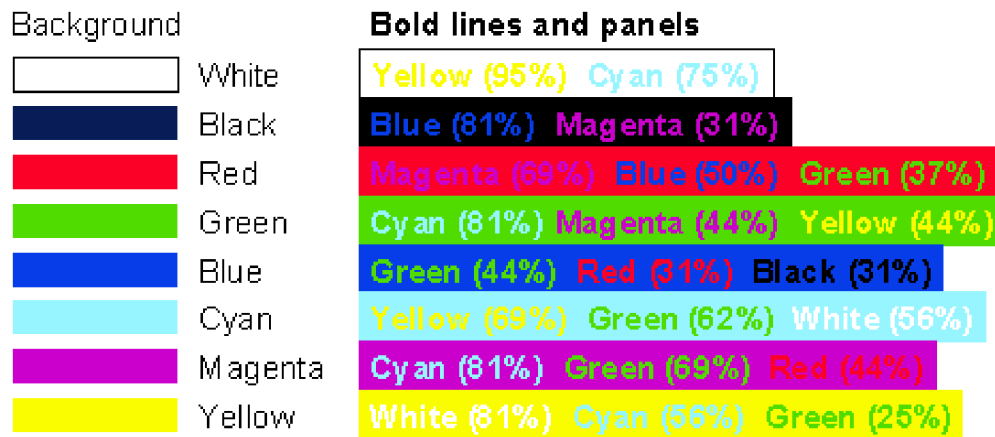
Fig. 10 represents the bad color combinations for thin lines and text. It can be read as follows: for thin lines and text displayed on a white background, yellow induces a legibility problem in almost all cases, while cyan does it in 94% of cases, and so forth.



**Fig. 10.** Bad color combinations for thin lines and text by reverse order of acceptance.

Fig. 11 represents the bad color combinations for bold lines and panels. It can be read as follows: for bold lines and panels displayed on a white background, yellow induces a legibility problem in nearly 95% of cases, while

cyan does it in 75% of cases, and so forth.



**Fig. 11.** Bad color combinations for bold lines and panels by reverse order of acceptance.

Now, having these experimental results in mind, we could see how to automate the testing of this guideline. The different colors used in the HTML code should be identified for this purpose. Color is rendered on computer-based screens as an additive mixture of three primary colors: red, green, and blue, the intensity of which is indicated by a value ranging from 0 to 255. Thus, in the Red-Green-Blue (RGB) model, each color consists in a triple (red-intensity, green-intensity, blue-intensity). This triple is represented in HTML as a hexadecimal code of the type #XXYYZZ, where XX,YY, and ZZ represents the different intensities respectively. Therefore, these codes need to be transformed into decimal for comparison. When these values belong to the set {00,33,66,99,CC, FF}, the resulting colors are said to be principal as no other palette than the basic palette should be loaded to display the colors. The bgcolor and color HTML tags specify the foreground and the background text colors respectively.

This is a basic procedure, which does not support the evaluation of hue, saturation and lightness: each composite color is reduced to its corresponding primary color, thus introducing a restriction of the initial guideline. Introducing another algorithm deriving moderator from the codes can solve this shortcoming: for example, “light blue”, “very light blue”. Graphic backgrounds are not supported although the different colors used in GIF or JPEG files could be identified. But in this case, a spatial algorithm should take care of physical appearance of text on top of graphical background colors, which is far more complex. As we can see here, the cost of developing the complete procedure for automated testing of a single guideline can be very high. In this case, the level of support is partial automation. A user testing technique to test the legibility can be used equally. The level of support is therefore delegated.

### Example #3

Style guide guidelines are important to check adherence to some corporate “Look & Feel” guidelines. Style guide compliance increases usability whereas violation will normally decrease it. Let us consider the following style guide example:

- ☞ All data fields must be accessible via a tabulation key in the same order. If fields are grouped together, then tab must access the fields within the group ([Levine, 1996](#)).

Fig. 12 shows an window where this guideline is not addressed: instead of relying on the tab order required by the task, it is relying on the order generated by the system. The different figures indicate the order according to which the different widgets are read or manipulated.





Fig. 12. Disturbing order of fields (Source: <http://www.iarchitect.com/controls.htm>).

According to the framework, this guideline could be organized as follows:

1. Requirements engineering
  - 1.1.1.2 Task order/sequencing: task order = domain objects sequence
2. Site specification
  - 2.1.1 Spec. C.E.-Compatibility
    - 2.1.1.1.2 Task.order/sequencing: fields order = task order

A procedure to automatically test this guideline would certainly exploit the appearance order of widgets in the HTML code and check whether this order is compatible (hence, the compatibility ergonomic criteria above) with the order recommended by the task. For this purpose, the test can only be achieved if such an order can be provided either manually (e.g., at run-time) or automatically (e.g., from a task model declaratively maintained). In this case, the level of automation would be semi-automatic. If no such support exists, it would become delegated.

## Example #4

Let us consider the following guideline re-expressed from the ISO 9241-Part 12 (ISO, 1999) standard:

☞ Error messages should not use humor.

According to the framework, this guideline could be formalized as follows:

- 3.1.10 Error information: information  $\not\subset$  no humor
- 3.2.2 Conc. C.E.-Quality-ErrorMessages: message  $\not\subset$  humor words and expressions

Evaluating this guideline on web pages automatically is quite complex since identifying error messages on web pages is hazardous; testing if a text contains humor is even more hazardous as this is language-, context-, and user-dependent. To solve these problems, a possible restriction of the above guideline could be to submit the text of each web page generated by HTTP error codes (e.g., HTTP 404 File not found) to a detection of commonly used humorous words and expressions. In this case, the original guideline cannot be tested automatically, but its restriction is. The expressiveness and the completeness of the initial guideline have consequently been reduced in this transformation. Hence, it is important to warn the evaluator of this possible loss of expressiveness. It is indeed almost impossible to encompass all humorous words and expressions in a list to be tested for occurrence, even if this list is made user expandable. The level of automation is consequently semi-automatic.

## Example #5

Let us consider the following guideline ([Levine, 1996](#)):

☞ Links should be consistent with related page titles.

According to the framework, this guideline could be formalized as follows:

2.1.2.5 Spec. C.E.-Guidance

2.1.2.5.1.3.2 Identification/Title

2.1.2.5.1.3.2.1 Windows: title = name of task

3.1 Conc. C.E.-Consistency

3.1.5. Labels

3.1.5.1 Links: label = title

Two situations for testing this guideline may occur: either the link label is exactly the related page title or at least one difference exists. The automation is straightforward in the first situation, while impossible in the second: from a semantic point of view, it is impossible to prove that a link label is semantically consistent with the related page title. However, a similarity score used in Information Retrieval can be computed from the amount of shared words and shared ordering: if the two expressions share a significant amount of terms in a relatively similar ordering, then it can be assumed that a syntactical consistency is established. Again, the resulting guideline is a restriction to the syntactic domain. The level of automation is consequently semi-automatic.

## Conclusion and future work

After building and using the framework we introduced, progress can be imagined independently along several dimensions:

- *Dimension of guidelines*: a complete and consistent definition for a Guideline Definition Language (GDL) is expected to uniformly represent the information that guidelines may involve and to communicate how to apply them. Several guidelines can be automated while some other cannot. A wider taxonomy of the reasons why they can or cannot be automated is needed. Second, an important future work would be to examine each EvalWeb guideline, to list information needed to evaluate them (manually or automatically) in GDL terms, to identify where in these tools and methods these data can be extracted, and to incorporate them in the evaluation process. Of course, not all guidelines can be currently tested automatically without any human control and intelligence. Especially until now, we have not yet fully organized all existing guidelines nor check enough guidelines. They are increasing every day. On the other hand, guidelines that cannot be organized (even after some framework adaptation) are likely to be non-testable and therefore should be considered instead as high-level principles.
- *Dimension of guidelines incorporation into approach*: developing a web site is a particular instance of developing a computer-based program. Therefore, a development life cycle adapted to the web should progressively emerge. We proposed such a life cycle according to which we are trying to attach guidelines. Further investigation and empirical validation are needed.
- *Dimension of the heuristic evaluation method*: usability testing can reveal usability flaws in poorly designed pages but cannot prove that a particular web page is well executed. This is a rather negative approach: if some usability problems are detected, then the corresponding pages are considered not usable and their lack of usability is proportional to the number of flaws revealed in the testing. If no flaws are

detected against the guidelines, one can be sure that they guarantee some usability, but not the complete desired usability. Moreover, it is likely that some other flaws remain unrevealed because no existing guidelines are addressing them. Finally, for some very original, unique web sites (e.g., [www.chman.com](http://www.chman.com)), little or no guidelines can be used to prove their original, yet usable, design. A more positive approach can be desired here.

- *Dimension of involved models and supporting tools*: designers in the study conducted by Löwgren and Laurén (1993) reported that they wanted to customize the results of the critique to their needs, to have not always the same repeatedly message produced by evaluation tools, to choose different levels of severity, and to have more control on the evaluation process. A significant contribution here would be to examine how these concerns can be addressed in a tool that supports the framework and to identify the benefits resulting from its application.

## Acknowledgements

This work has been carried out under the auspices of [EvalWeb](#) research project, funded by research program “Cognition et Conception”, [G.I.S. Sciences de la Cognition](#), of the Centre National de la Recherche Scientifique (CNRS), the Metro Web research project funded by Région Wallonne and by a Tournesol mobility program. It is also an initiative from the International Special Interest Group (SIG) on [Tools for Working with Guidelines](#). The authors would like to thank HFWeb’2000 anonymous reviewers for their insightful comments on earlier version of this manuscript and Jacob Eisenstein, [RedWhale Software Corp.](#), for early proofreading of this document

## References

1. Balasubramanian, V. and Turoff, M., *User Interface Design Guidelines for a Hypertext Framework*, Proceedings of ACM Hypertext’93, ACM Press, New York, 1993, p. 33.
2. Bastien, J.M.C. and Scapin, D.L., *Evaluating a User Interface with Ergonomic Criteria*, International Journal of Human-Computer Interaction, Vol. 7, 1995, pp. 105–121.
3. Bastien, J.M.C., Scapin, D.L., Leulier, C., *The Ergonomic Criteria and the ISO 9241-10 Dialogue Principles: A comparison in an evaluation task*, Interacting with Computers, 11(3), 1999, pp. 299–322.
4. Borges, J.A., Morales, I. and Rodríguez, N.J., *Guidelines for Designing Usable World Wide Web Pages*, in Companion Proceedings of ACM Conference on Human Aspects in Computing Systems CHI’96 (Vancouver, April 14-18, 1996), ACM Press, New York, 1996, pp. 277–278. Accessible at [http://www.acm.org/sigchi/chi96/proceedings/shortpap/Rodriguez/rn\\_txt.htm](http://www.acm.org/sigchi/chi96/proceedings/shortpap/Rodriguez/rn_txt.htm)
5. Comber, T., *Building Usable Web Pages: An HCI Perspective*, in Proceedings of Australian Conf. on the Web AusWeb’95 (Ballina, August 1995), Nordsearch Ltd, Ballina, 1995, pp. 119–124.
6. Cooper, M. *Evaluating Accessibility and Usability of Web Pages*, in Proc. of 3<sup>rd</sup> International Conference on Computer-Aided Design of User Interfaces CADUI’99 (Louvain-la-Neuve, October 21-23, 1999), Kluwer Academics, Dordrecht, 1999, pp. 33–42.
7. Detweiler, M.C and Omanson, R.C., *Ameritech Web Page User Interface Standards and Design Guidelines*, Ameritech Corp., Chicago, 1996. Accessible at [http://www.ameritech.com/corporate/testtown/library/standard/web\\_guidelines/index.html](http://www.ameritech.com/corporate/testtown/library/standard/web_guidelines/index.html)
8. *PIENet Web Publishing Standards and Guidelines*, Australian Department of Primary Industries and Energy, 1997. Accessible at <http://www.dpie.gov.au/dpie/web/construction.html>

9. Grose, E.M., Forsythe, Ch., Ratner, J. (eds.), *Human Factors and Web Development*, Lawrence Erlbaum Associates, Mahwah, 1998.
10. Hardman, L. and Sharratt, B.S., *User-Centred Hypertext Design: The Application of HCI Design Principles and Guidelines*, in “Designing Hypertext and Hypertext and Design “, Proceedings of HYPERTEXT’II, 1989, pp. 252-259.
11. HFES/ANSI 200 Standard: Draft, Human Factors & Ergonomics Society, Santa Monica, 1997.
12. IBM WEB Guidelines - Complete set, IBM Corporation, 1997. Accessible at [http://www-3.ibm.com/ibm/easy/eou\\_ext.nsf/publish/572](http://www-3.ibm.com/ibm/easy/eou_ext.nsf/publish/572)
13. ISO Draft International Standard (DIS) 9241, *Ergonomic Requirements for office work with visual display terminals*, International Standards Organization, Geneva, 1999.
14. Leulier, C., Bastien, J. M. C., Scapin, D.L., *Compilation of ergonomic guidelines for the design and evaluation of Web sites*, Commerce & Interaction Report, Institut National de Recherche en Informatique et en Automatique, Rocquencourt, France, 1998.
15. Levi, M.D. and Conrad, F.G., *A Heuristic Evaluation of a World Wide Web Prototype*, Interactions, Vol. 3, No. 4, 1996, pp. 50-61.
16. Levine, R., *User Interface Design for Sun Microsystem’s Internal Web*, Sun Microsystems Inc., 1996. Accessible at <http://www.sun.com/styleguide>
17. Linz, T. and Daigl, M., *How to Automate Testing of Graphical User Interfaces*, ESSI PIE 24306 research report, imbus GmbH, Möhrendorf, 1998. Accessible at [http://www.imbus.de/html/GUI/AOUIIS-full\\_paper-1.3.html](http://www.imbus.de/html/GUI/AOUIIS-full_paper-1.3.html)
18. Löwgren, J., Laurén, U., *Supporting the use of guidelines and style guides in professional user interface design*, Interacting With Computers, Vol. 5, 1993, pp. 385-396.
19. Lynch, G., Palmiter, S. and Tilt, Ch., *The Max Model: A Standard Web Site User Model*, in Proceedings of the 4<sup>th</sup> International Conference on Human Factors and the Web HFWeb’98 (Basking Ridge, June 5, 1998), J. Cantor (ed.), 1998. Accessible at <http://zing.ncsl.nist.gov/hfweb/proceedings/lynch/index.html>
20. Mayhew, D., *Principles and Guidelines in Software User Interface Design*, Prentice Hall, Englewood Cliffs, 1992.
21. Murch, G.M., *Colour Graphics - Blessing or Ballyhoo?*, Chapter 7: The Visual Channel, in R.M. Baecker & W.A.S. Buxton (eds.), *Readings in Human-Computer Interaction - A Multidisciplinary Approach*, Morgan Kaufmann Publishers, Los Altos, 1987, pp. 333–341.
22. Nielsen, J. *A Virtual Protocol Model for Computer-Human Interaction*, International Journal of Man-Machine Studies, Vol. 24, No. 3, 1986, pp. 301-312
23. Nielsen, J., Mack, R.M., *Usability Inspection Methods*, John Wiley & Sons, New York, 1994.
24. Ratner, J., Grose, E.M., Forsythe, Ch., *Characterization and Assessment of HTML Style Guides*, in Proceedings of ACM Conference on Human Factors in Computing Systems CHI’96 (Vancouver, 14-18 April 1996), ACM Press, New York, 1996, pp.115–116. Accessible at [http://www.acm.org/sigchi/chi96/proceedings/intpost/Ratner/rj\\_txt.htm](http://www.acm.org/sigchi/chi96/proceedings/intpost/Ratner/rj_txt.htm)
25. Scapin, D.L., Bastien, J.M.C., *Ergonomic criteria for evaluating the ergonomic quality of interactive systems*, Behaviour & Information Technology, 16, 1997, pp. 220–231.

26. Scapin, D.L., Guarrigues, S., Farenc, C., Vanderdonckt, J., Palanque, P., Bastide, R., Bastien, J.M.C. and Leulier, C., *Conception ergonomique d'interfaces web: démarche et outil logiciel de guidage et de support*, Rapport final du projet EvalWeb, Appel d'offre "Cognition et Conception", GIS Sciences de la Cognition, CNRS, Paris-Toulouse-Louvain-la-Neuve, December 1999.
27. Scapin, D.L., Vanderdonckt, J., Farenc, Ch., Bastide, R., Bastien, Ch., Leulier, C., Mariage, C., and Palanque, P., *Transferring Knowledge of User Interfaces Guidelines to the Web*, in Proceedings of the International Workshop on Tools for Working with Guidelines TFWWG'2000 (Biarritz, October 7-8, 2000), Springer-Verlag, London, 2000.
28. Scholtz, J., Laskowski, S. and Downey, L., *Developing Usability Tools and Techniques for Designing and Testing Web Sites*, in Proceedings of the 4<sup>th</sup> International Conference on Human Factors and the Web HFWeb'98 (Basking Ridge, June 5, 1998), J. Cantor (ed.), 1998. Accessible at <http://www.research.att.com/conf/hfweb/proceedings/scholtz/index.html>
29. Vanderdonckt, J., *Guide ergonomique des interfaces homme-machine*, Presses Universitaires de Namur, Namur, 1994.
30. Vanderdonckt, J., *Development Milestones towards a Tool for Working with Guidelines*, Interacting with Computers, Vol. 12, N°2, 1999, pp. 81-118. Accessible at <http://www.elsevier.nl/gej-ng/10/23/72/31/21/show/toc.htm>
31. Vanderheiden, G.C., Chisholm, W.A., Ewers, N., Dunphy, S. M., *Unified Web site accessibility guidelines*, Version 7.2, Trace Center, University of Wisconsin-Madison, 1997. Accessible at <http://trace.wisc.edu/text/guidelns/htmlgide/htmlgide.htm>
32. *Yale C/AIM Web Style Guide*, Yale-New Haven Medical Center, 1997. Accessible at <http://info.med.yale.edu/caim/manual/contents.html>

## Appendix. Framework for Organizing Web Usability Guidelines

### 1. REQUIREMENTS ENGINEERING

#### 1.1. TASKS

##### 1.1.1. Task Structure

- 1.1.1.1. Task organization
- 1.1.1.2. Task order/ Sequencing
- 1.1.1.3. Task stability
- 1.1.1.4. Timing
- 1.1.1.5. Specific tasks
  - 1.1.1.5.1. Visual search
  - 1.1.1.5.2. User attention
  - 1.1.1.5.3. Discrimination 2 categories

##### 1.1.2. Information needed

- 1.1.2.1. Nature
- 1.1.2.2. Status
- 1.1.2.3. Order
- 1.1.2.4. Groups
- 1.1.2.5. Localization

##### 1.1.3. Information about Tasks

- 1.1.3.1. General
- 1.1.3.2. General conventions
- 1.1.3.3. Application domain conventions
- 1.1.3.4. Language conventions
- 1.1.3.5. Coding conventions

#### 1.2. USERS

##### 1.2.1. Users characteristics

- 1.2.1.1. Several Populations w. Different Skills/ Experience levels
- 1.2.1.2. Level of experience in the tasks
  - 1.2.1.2.1. Reading level
- 1.2.1.3. Level of experience with computers
  - 1.2.1.3.1. Unfamiliar with field format?
- 1.2.1.4. Cultural characteristics (e.g., language, reading order)
- 1.2.1.5. Handicaps

### 1.3. EQUIPMENT

- 1.3.1. Compatibility color vs. monochrome displays

## 2. SITE SPECIFICATION

### 2.1. SPEC. CONTENTS

#### 2.1.1. Spec. C.E.-COMPATIBILITY

##### 2.1.1.1. Spec. C.E.-COMPATIBILITY-Tasks

- 2.1.1.1.1. Structure Tasks
  - 2.1.1.1.1.1. Task Organization
  - 2.1.1.1.1.2. Task Order/Sequencing
  - 2.1.1.1.1.3. Stability
  - 2.1.1.1.1.4. Timing/ Shortcuts
- 2.1.1.1.2. Information Needed
  - 2.1.1.1.2.1. Nature
  - 2.1.1.1.2.2. Order
  - 2.1.1.1.2.3. Numbering
  - 2.1.1.1.2.4. Grouping
  - 2.1.1.1.2.5. Localization
  - 2.1.1.1.2.6. Coding
  - 2.1.1.1.2.7. Formats
  - 2.1.1.1.2.8. Cursors
- 2.1.1.1.3. Information about Tasks
- 2.1.1.1.4. Other Task related

##### 2.1.1.2. Spec. C.E.-COMPATIBILITY-USERS

- 2.1.1.2.1. Several Populations w. Different Levels of Experience/ Skills
- 2.1.1.2.2. Task experience level
- 2.1.1.2.3. Experience with computers
  - 2.1.1.2.3.1. Inexperienced
  - 2.1.1.2.3.2. Experienced
- 2.1.1.2.4. Reading levels
- 2.1.1.2.5. Cultural differences
- 2.1.1.2.6. Accessibility (User Limitations)
- 2.1.1.2.7. Spec. C.E.-USERS' EXPERIENCE

- 2.1.1.2.7.1. Other User related

##### 2.1.1.3. Spec. C.E.-COMPATIBILITY-EQUIPMENT

#### 2.1.2. Spec. Contents Non Task/User/Equipment-Specific

##### 2.1.2.1. General

- 2.1.2.1.1. General Spec. Displays
- 2.1.2.1.2. General Spec. Dialogue & Controls

##### 2.1.2.2. Spec. C.E.-SIGNIFICANCE/CODING

- 2.1.2.2.1. General
- 2.1.2.2.2. Uniqueness
- 2.1.2.2.3. Distinctness
- 2.1.2.2.4. Coding schemes

- 2.1.2.2.5. Showing Relationships
  - 2.1.2.2.5.1. Windows (Primary/Secondary)
  - 2.1.2.2.6. Windows Control
  - 2.1.2.2.7. Abbreviations
  - 2.1.2.2.8. Colors
  - 2.1.2.2.9. Icons
  - 2.1.2.2.10. Geometric Shapes
  - 2.1.2.2.11. Blinking
  - 2.1.2.2.12. Sizes
  - 2.1.2.2.13. Voice
- 2.1.2.3. Spec. . C.E.-Explicit Control
  - 2.1.2.3.1.1. Spec. . C.E.-EXPLICIT USER ACTION
  - 2.1.2.3.1.2. Spec. . C.E.-USER CONTROL
- 2.1.2.4. Spec. . C.E.-WORKLOAD
  - 2.1.2.4.1. Spec. . C.E.-BREVITY
    - 2.1.2.4.1.1. Spec. . C.E.-CONCISENESS
    - 2.1.2.4.1.2. Spec. . C.E.-MINIMAL ACTIONS
      - 2.1.2.4.1.2.1. Cursor
      - 2.1.2.4.1.2.2. etc.
  - 2.1.2.4.2. Spec. . C.E.-INFORMATION DENSITY
- 2.1.2.5. Spec. C.E.-GUIDANCE
  - 2.1.2.5.1. Spec. C.E.-PROMPTING
    - 2.1.2.5.1.1. Contents
      - 2.1.2.5.1.1.1. General
    - 2.1.2.5.1.2. Prompting Semantics
    - 2.1.2.5.1.3. Prompting Syntax
      - 2.1.2.5.1.3.1. Cues
      - 2.1.2.5.1.3.2. Identification/ Title
        - 2.1.2.5.1.3.2.1. Windows
        - 2.1.2.5.1.3.2.2. Icons
        - 2.1.2.5.1.3.2.3. Forms
      - 2.1.2.5.1.3.3. Special Prompting Codes
        - 2.1.2.5.1.3.3.1. Markers
        - 2.1.2.5.1.3.3.2. Line orientation
      - 2.1.2.5.1.3.4. Status
      - 2.1.2.5.1.3.5. Entry Fields
      - 2.1.2.5.1.3.6. Units of Measurements
      - 2.1.2.5.1.3.7. Showing Relationships
        - 2.1.2.5.1.3.7.1. Any Text
        - 2.1.2.5.1.3.7.2. Text Fields
        - 2.1.2.5.1.3.7.3. Lists
        - 2.1.2.5.1.3.7.4. Tables
      - 2.1.2.5.1.3.8. Auditory
      - 2.1.2.5.1.3.9. Cursors
      - 2.1.2.5.1.3.10. Dragging
      - 2.1.2.5.1.3.11. Sizing
      - 2.1.2.5.1.3.12. Rotation
    - 2.1.2.5.1.3.10. Dragging
    - 2.1.2.5.1.3.11. Sizing
    - 2.1.2.5.1.3.12. Rotation
  - 2.1.2.5.2. Spec. C.E.-GROUPING/DISTING
    - 2.1.2.5.2.1. Spec. C.E.-BYLOCATION
    - 2.1.2.5.2.2. Spec. C.E.-BYFORMAT

- 2.1.2.5.2.2.1. Cursor
- 2.1.2.5.2.2.2. Windows
- 2.1.2.6. Conc. C.E.-LEGIBILITY
- 2.1.2.7. Spec. C.E.-FEEDBACK

### **3. SITE DESIGN**

#### **3.1. CONC. C.E.-CONSISTENCY**

- 3.1.1. Windows Format
- 3.1.2. Areas to Distinguish
- 3.1.3. Prompts
- 3.1.4. Commands & Controls
- 3.1.5. Labels
- 3.1.6. Coding (& Messages)
- 3.1.7. Home position
- 3.1.8. Markers
- 3.1.9. Status Information
- 3.1.10. Error Information
- 3.1.11. On line Help
- 3.1.12. Modes
- 3.1.13. Options Location

#### **3.2. CONC. C.E.-ERROR/MANAGEMENT**

- 3.2.1. Conc. C.E.-ERROR PROTECTION
- 3.2.2. Conc. C.E.-QUALITY-Error Messages
- 3.2.3. Conc. C.E.-ERROR CORRECTION

#### **3.3. CONC. C.E.-FLEXIBILITY**

#### **3.4. CONC. HELP on LINE**

#### **3.5. CONC. DOCUMENTATION**

### **4. SITE DEVELOPMENT (to be expanded)**

### **5. SITE USAGE AND EVALUATION (to be expanded)**

### **6. SITE MAINTENANCE (to be expanded)**

A Framework for Organizing Web Usability Guidelines