INTERNATIONAL STANDARD

ISO
9241-161

First edition
2016-02-15

Ergonomics of human-system interaction —

Part 161:
Guidance on visual user-interface elements

Ergonomie de l'interaction homme-système —

Partie 161: Éléments de l'interface utilisateur

COPYRIGHT PROTECTED DOCUMENT

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 159, *Ergonomics*, Subcommittee SC 4, *Ergonomics of human–system interaction*.

ISO 9241 consists of the following parts, under the general title *Ergonomic requirements for office work with visual display terminals (VDTs)*:

— *Part 1: General introduction*

— *Part 2: Guidance on task requirements*

— *Part 5: Workstation layout and postural requirements*

— *Part 6: Guidance on the work environment*

— *Part 11: Guidance on usability*

— *Part 12: Presentation of information*

— *Part 13: User guidance*

— *Part 14: Menu dialogues*

— *Part 15: Command dialogues*

— *Part 16: Direct manipulation dialogues*

ISO 9241 also consists of the following parts, under the general title *Ergonomics of human-system interaction*:

— *Part 20: Accessibility guidelines for information/communication technology (ICT) equipment and services*

— *Part 100: Introduction to standards related to software ergonomics* [Technical Report]

— *Part 110: Dialogue principles*

Introduction

In different communities in the interactive system development ecosystem, the use, the names and the understanding of user-interface elements differs significantly. One of the results is that users have to cope with elements which differ in terms of keyboard entry and control, mouse behaviour, visual presentation of functionality and different options to control elements. **Consistent element behaviour, functionality and rendering is crucial for the usability of user interfaces.** This causes added efforts in all stakeholders in human-centred design activities, since this multitude needs to be managed in order to ensure high-quality collaboration of various specialists. Especially in the light of new emerging user-interface concepts and designs, a common definition of visual user-interface elements and the rationale for their selection, as well as their use can be regarded as an effort to sustain cooperation and ensure a sound basis for professional conversation. It is also of importance to state that this part of ISO 9241 of visual user-interface elements in no ways predetermines a visual style of the elements themselves, thus avoiding to impress determinants in creation, brand usage and style development. In addition, this part of ISO 9241 is laid out in an independent of platform specifics, so that no specific industrial user-interface styleguide, implementation technology or development process needs to be observed in order to be compliant with this part of ISO 9241.

This part of ISO 9241 aims to provide information on visual user-interface elements to help those responsible for managing software design and re-design processes, create user interface specifications, styleguides and visual concepts to identify, plan and design effective, efficient and satisfactory interactive systems.

Visual user-interface elements described in this part of ISO 9241 complements existing systems design approaches, methods or processes. They can be referenced in any kind of user interface strategy, regardless of the technology used for the user interface.

Table 1 — Overview of different visual user interface properties that are used to build a user interface design

User Interface Design				
Interactive Properties		Informative Properties		Decorative Properties

Ergonomics of human-system interaction —

Part 161:
Guidance on visual user-interface elements

1 Scope

This part of ISO 9241 describes visual user-interface elements presented by software and provides requirements and recommendations on when and how to use them. This part of ISO 9241 is concerned with software components of interactive systems to make human-system interaction usable as far as the basic interaction aspects are concerned.

This part of ISO 9241 provides a comprehensive list of *generic* visual user-interface elements, regardless of a specific input method, visualization, and platform or implementation technology. The guidance given in this part of ISO 9241 is intended to be used in conjunction with ISO 9241 guidance on dialogue techniques. It recognizes that additional elements can evolve. It also addresses *derivates*, *compositions* (assemblies) and *states* of user-interface elements. It gives requirements and recommendations on selection, usage and dependencies of user-interface elements and their application. It is applicable regardless of a fixed, portable or mobile interactive system.

It does not provide detailed coverage of the methods and techniques required for design of user-interface elements. This part of ISO 9241 does not address implementation (e.g. graphical design of elements) and interaction details for specific input methods or technologies. It does not cover decorative user-interface elements that are intended to address solely aesthetic (hedonic) qualities in the user interface, e.g. background images.

The information in this part of ISO 9241 is intended for use by those responsible for the selection and implementation of visual user-interface elements in interactive systems and for evaluating user interfaces. It is intended for use by those planning and managing platform specific aspects of user interface screen design. It also provides guidance for human factors/ergonomics and usability professionals involved in human-centred design. It addresses technical issues only to the extent necessary to allow users of this part of ISO 9241 to understand the relevance and importance of a consistent interface element usage and selection in the design process as a whole.

Annex A provides a guide to selection of different visual user interface elements depending of their appropriate application.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 9241-143:2012, *Ergonomics of human-system interaction — Part 143: Forms*

ISO 9241-171:2008, *Ergonomics of human-system interaction — Part 171: Guidance on software accessibility*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
accessibility
<interactive system>
usability of a product, service, environment or facility by people with the widest range of capabilities

Note 1 to entry: The concept of accessibility addresses the full range of user capabilities and is not limited to users who are formally recognized as having a disability.

Note 2 to entry: The usability-orientated concept of accessibility aims to achieve levels of effectiveness, efficiency and satisfaction that are as high as possible considering the specified context of use, while paying particular attention to the full range of capabilities within the user population.

[SOURCE: ISO 9241-171:2008]

3.2
default action
operation predefined for an object or set of objects (including an entire dialogue box) that will take place when the user actuates the default action mechanism

[SOURCE: ISO 9241-143:2012]

EXAMPLE   ln a search application, pressing the ENTER key activates the Search button.

3.3
canvas
area on the screen that is used to present or manipulate data, user-interface elements or to collect user input

3.4
click zone
area in the user interface that is activated by input device

3.5
deactivation
operation on a user-interface element that renders the object unable to receive user input that enters or changes data

3.6
dialogue
interaction between a user and an interactive system as a sequence of user actions (inputs) and system responses (outputs) in order to achieve a goal

Note 1 to entry: User actions include not only entry of data but also navigational actions of the user.

Note 2 to entry: Dialogue refers to both the form (syntax) and the meaning (semantics) of interaction.

[SOURCE: ISO 9241-110:2006]

3.7
form element
user-interface element applicable to forms

[SOURCE: ISO 9241-143:2012]

3.8
form
structured display of fields and other user-interface elements that the user reads, fills in, selects entries for, or modifies

EXAMPLE   Through choice buttons or radio buttons.

[SOURCE: ISO 9241-143:2012]

3.9
form structure
organization and arrangement of information on individual and sets of forms, and the interconnections among forms for user navigation

[SOURCE: ISO 9241-143:2012]

3.10
hover area
area of the user interface that is sensitive to an overlying pointer

Note 1 to entry: In touchscreen interfaces, often hover events are not used. Alternative means to signify the respective element are to be considered.

3.11
hover effect
change of the user-interface element representation triggered by overlying pointer

3.12
list
set of items

Note 1 to entry: A list can be sorted or unsorted, depending on the task.

3.13
navigation
ability to move from one user-interface element to another within a user interface and to move throughout an interactive system

3.14
picker
selector
chooser
user-interface element that provides means of selection of formatted data

Note 1 to entry: Examples of formatted data are colour values, dates, time.

3.15
pointing device
device that translates a human controlling operation to a controlling operation on the display

Note 1 to entry: Depending on the applied technology, not only machine devices but also parts of the human body (e.g. fingers, arms) can currently be used as pointing devices.

Note 2 to entry: Pointing devices typically have buttons that are used to activate or manipulate user-interface elements.

Note 3 to entry: Almost any hardware can be used to control a pointer with the appropriate software.

[SOURCE: ISO 9241-16:1999]

3.16
state
status
distinct condition of an object

[SOURCE: ISO 11064-5:2008]

3.17
touch area
part of the user interface that is sensitive for touch events

3.18
user-interface element
user-interface object
entity of the user interface that is presented to the user by the software

EXAMPLE   Text, graphic, control.

Note 1 to entry: User-interface elements can be interactive or not.

Note 2 to entry: Both entities relevant to the task and entities of the user interface are regarded as user-interface elements. A user-interface element can be a visual representation or an interaction mechanism for a task object (such as a letter, sales order, electronic part or wiring diagram) or a system object (such as a printer, hard disk or network connection). It can be possible for the user to directly manipulate some of these user-interface elements.

Note 3 to entry: User-interface elements in a graphical user interface include such things as basic objects (such as window title bars, menu items, push buttons, image maps, and editable text fields) or containers (such as windows, grouping boxes, menu bars, menus, groups of mutually-exclusive option buttons, and compound images that are made up of several smaller images). User-interface elements in an audio user interface include such things as menus, menu items, messages, and action prompts.

[SOURCE: ISO 9241-171:2008]

4   Accessibility

Software that provides user-interface elements shall use the accessibility services provided by the platform to cooperate with assistive technologies (ISO 9241-171:2008, 8.5.3).

It is important to recognize that visual user-interface elements will also be used by users unable to interact with them in a visual way.

All visual user-interface elements shall provide alternative text to support accessibility (see ISO IEC 20071-11 for guidance on text alternatives to images).

User-interface elements shall conform to ISO 9241-171.

NOTE   Conformity with ISO 9241-171 implies that every applicable requirement and every applicable recommendation is met. See ISO 9241-171:2008, 7.2 for details.

5   Relationship of input methods and visual user-interface elements

Visual user-interface elements may be operated by various input methods using various technologies including the following:

— typing (e.g. using keyboard);

— pointing (e.g. using mouse, pens, gesture recognition, eye tracking);

— speaking (e.g. voice commands, voice recognition).

User-interface elements shall be operable by the input methods available in the interactive system.

NOTE 1   This is important to support accessibility (see ISO 9241-171).

Input methods may be used singularly or in combination.

Some input methods use specific user interface elements.

EXAMPLE 1   Keyboard use requires a clear visualization of the input focus where interaction takes effect.

Use of a pointing method (mouse and/or pen) requires a visual pointer to indicate the virtual position of the device where interaction takes effect.

Due to direct manipulation, touchscreens do not require a pointing device.

EXAMPLE 2   If a keyboard is provided, check boxes are accessible by the keyboard. Checking and unchecking a check box is realized with the use of the same key.

EXAMPLE 3   If a pointing device is provided, check boxes are accessible by the pointing device.

NOTE 2   For touch input, a minimum user-interface element size is to be observed to ensure effective use of the user interface.

NOTE 3   Different platforms have different strategies to implement accessibility of user-interface elements, and these differ further based on the type of disability. The ISO/IEC 13066 multipart standard series includes technical reports on different platform specific accessibility APIs.

6   States of visual user-interface elements

A user-interface element may have the following different states (status) depending on system status and user interaction. Each state shall be visually clearly distinguishable from all other states.

States listed together in a single item within the following list are mutually exclusive.

a)   Visible vs. invisible: user-interface element is visible vs. not visible by the user.

b)   Active vs. inactive (available for use vs. unavailable for use): user-interface element is ready to receive user input to enter or change data or not ready to receive user input.

c)   Focused vs. unfocused: keyboard events are assigned to the user-interface element vs. keyboard events are not assigned to the user-interface element.

d)   Selected vs. unselected: user-interface element has been selected or not been selected.

e)   Checked vs. unchecked vs. conditionally checked: user-interface element value has been set vs. has not been set vs. represents a group of elements with both checked and unchecked status.

f)   Filled vs. empty: user-interface element contains content vs. it does not contains content/is empty.

g)   Pressed vs. not-pressed: user-interface element (e.g. a button) visually appears as pressed vs. not pressed, typically to visualize an on/off state.

h)   Collapsed vs. expanded: user-interface element is displayed with minimized vs. maximized space consumption.

NOTE   There may also be other states beside the ones listed above.

7   Describing visual user-interface elements

The description of each user-interface element in this document is structured into the following sections:

— description

— a short explanation of the element and its purpose;

— components

— a list of visual assets or user interface element from which the respective user interface element is assembled;

— an illustration as one example (if applicable) of the user interface element that indicates also the components of the element with additional text describing the illustration. The illustration is not intended to be a design guidance;

— states

— a list of different states that can be assigned to the user-interface element. For each user-interface element described, the states listed are the standard or minimum set; additional states are possible depending on the design solution;

— when to use [user-interface element name]

— identification when the element is the appropriate choice (or not);

— identification of alternative elements to achieve the same results;

— how to use [user-interface element name]

— guidance on how the user interacts logically with the element;

— guidance on how to incorporate the element into the user interface design.

NOTE   Platform specific user-interface guidelines can constrain the location or use of user-interface elements.

8   Visual user-interface elements

8.1   Accordion

8.1.1   Description

An accordion is an assembly of multiple collapsible containers that each present a group of user-interface elements.

An accordion is either vertically or horizontally arranged. It expands therefore vertically or horizontally.

8.1.2   Components

An accordion consists of the following components:

— A number of sections. Per section:

a)   section label, describing the section;

b)   means to collapse/expand section;

c)   canvas, displaying section content.
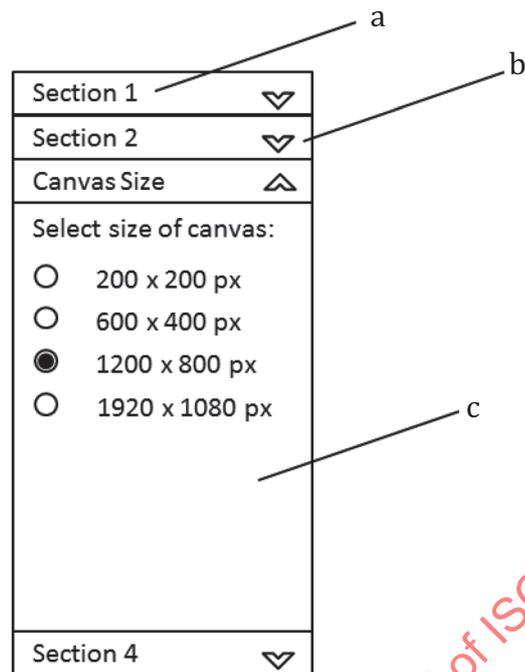
Key

a   section label
b   means to collapse/expand section
c   canvas

Figure 1 — Example of an accordion, vertically expanding

8.1.3   States

The containers of the accordion are collapsible, i.e. they can be in either collapsed or expanded mode. Depending on the implementation of the accordion, 0, 1, or multiple sections can be expanded at a time.

8.1.4   When to use an accordion

An accordion is an applicable design solution, if both of the following conditions are true:

— available space is limited: it is not possible to display all panes at once;

— set of panes is static.

In the case that one or both conditions are not met:

— space is not limited

   — consider using a group;

— the set of panes is variable

   — consider using list or table with detail display for selected rows;

— alternative user-interface elements in case of restricted space

   — consider **Hierarchical list**;

   — consider **Menu**;

   — consider **tab set**;

— consider other mechanisms to select from a set of views. user-interface elements for selection are **combination box**, **check boxes**, **radio buttons**, and **toggle buttons**.

8.1.5 How to use an accordion

The interaction that expands a pane of an accordion shall be the same as the interaction that closes a pane of an accordion.

8.2 Analogue form element/slider

8.2.1 Description

An analogue form element is a user-interface element that allows the user to select a value within a continuous range.

EXAMPLE   Rotary dials, slider.

[SOURCE: ISO 9241-143:2012]

8.2.2 Components

Components of an analogue form element depend on the design solution applied. Common to all design solutions are the following:

a) label;

b) means to change data in a bi-directional way;

c) output of the current value of the data (e.g. percentage or tick marks).

Key
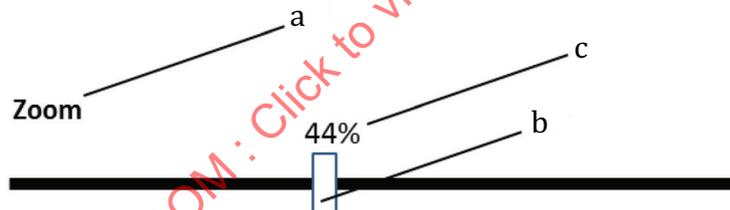
a   label

b   means to change data in a bi-directional way

c   output of current value of data (e.g. percentage or tick marks)

Figure 2 — Example of an analogue form element (slider), where a handle is moved along an horizontal axis, visualized as a bar, to increase or decrease a value represented as percentage

8.2.3 States

An analogue form element can have the following states:

— active/inactive.

8.2.4 When to use an analogue form element

An analogue form element is an applicable design solution, if the following conditions are true:

— bounded values are needed as input to the system;

— changing values where continuous, immediate feedback is needed about the value setting's effect;

— enough space in one direction;

— enough space to visualize values.

In the case that one or both conditions are not met

— consider stepper.

Alternative user-interface elements in case of restricted space

— consider new Window with entry fields.

8.2.5 How to use an analogue form element

An analogue form element should be used for input only.

8.3 Carousel/Carrousel

8.3.1 Description

A carousel is a list of elements displayed in a circular layout that can be moved along that layout.

NOTE 1   Often the element closest to the front of the circle is displayed bigger, the other elements smaller according to the perspective from the user.

NOTE 2   Depending on the design of the carousel or the amount of elements, not all of the elements of a carousel might be visible at the same time.

8.3.2 Components

Components of a carousel depend on the design solution applied. Common to all design solutions are

a) data elements, and

b) means to change the focused element.
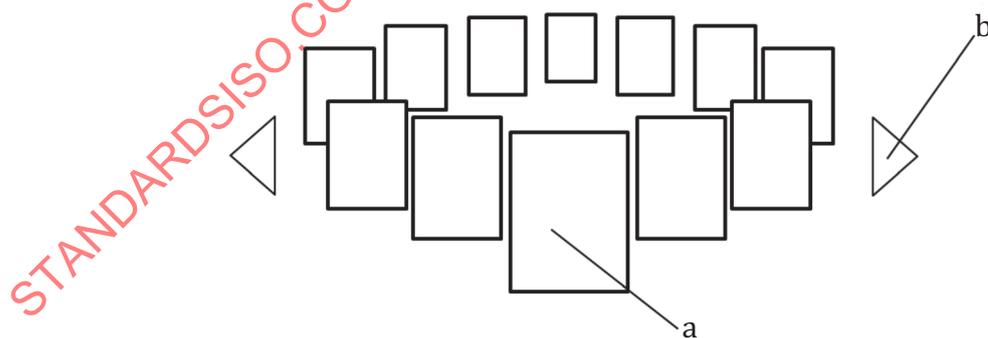
Key

a   data elements
b   means to change focused element

Figure 3 — Example of a carousel with data elements arranged in an ellipse, the one with focus displayed in the foreground, with left and right buttons to change the focus

8.3.3 States

Carousel elements can be

— invisible or visible, and

— focused or unfocused.

8.3.4 When to use a carousel

A carousel is an applicable design solution, if the following conditions are true:

— unordered or ordered list of data elements of similar type;

— graphical representation of data are to be presented;

— visualization of the data element is helpful for the selection decision;

— no comparison of data are required;

— the number of data elements is known.

In the case that the conditions are not met

— consider **hierarchical list**, or

— consider **list box**.

8.3.5 How to use a carousel

— Presented data should be of the same semantic type or homogenous.

— visual data rather than alphanumeric data should be used in elements of a carousel.

— When element of the carousel are hidden, the carousel shall convey clearly the means to access this information.

— The carousel should provide as much contextual information as possible, including size of the data set and the current focus.

— The presented data should be organized in a logical order suitable for the task.

8.4 Check box/check button

8.4.1 Description

A check box is a user-interface element used to select an option that can be either {"on" or "off"} or {"yes" or "no"} with a label (usually text) that indicates what the option is and an indicator (e.g. graphic checkmark) that indicates whether or not the option is checked.

NOTE 1   Check boxes in a group are mutually independent; i.e. selecting one option does not affect the selection of other options in that check box group. In a group of check boxes, one or many of them can be selected at the same time.

NOTE 2   Contrast with radio button.

8.4.2 Components

A check box consists of the following components:

a) indicator, whether or not the option is checked;

b) label, indicating what the option is.

a

b

x Choice Number 1
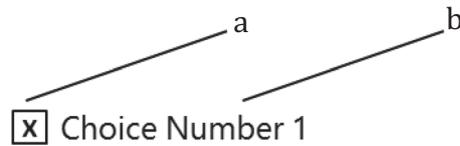
Key

a indicator

b label

NOTE   The physical relationship of indicator and label can take different forms.

Figure 4 — Example of a check box with a label on the right of the check box

8.4.3   States

A check box can have the following states:

— focused/unfocused;

— active/deactivated;

— editable/display only;

— checked/unchecked/conditionally checked.

8.4.4   When to use a check box

A check box is an applicable design solution, if all of the following conditions are true.

In the context of choosing values and options

— choices are not exclusive, more than one choice can be selected by the user,

— the maximum number of choices should not exceed 10 unless choices are structured in groups or sets, and

— the number of choices is static.

In the context of indication in lists

— choices are not exclusive, more than one choice can be selected by the user, and

— the number of choices varies.

Alternative user-interface elements in case of restricted space

— consider Dropdown list box.

8.4.5   How to use a check box

Check boxes that share the same contextual meaning shall be presented in a way that clearly depicts that shared association.

NOTE   This relation is often realized by showing dependent choices in a grouping element and/or using the law of proximity.

Within a design solution, the physical relationship of label and indicator shall be consistent.

Within a design solution, the visualization of the indicator shall be consistent (e.g. always use "x" or always use "✓").

A check box shall not convey other selection states than: selected, conditionally selected, and unselected.

The different states of a check box shall be realized in a way that clearly differentiates the states.

8.5 Collapsible container

8.5.1 Description

A collapsible container is a user-interface element that contains a group of other user-interface elements that can be visually expanded and collapsed.

8.5.2 Components

Components of a collapsible container depend on the design solution and the data that is involved in the context. Common to all design solutions are the following:

a) label of the container;

b) means to collapse or expand the container;

c) indicator, displaying expanded or collapsed status;

d) canvas to display content of the collapsible container;

e) data elements or other user-interface elements.
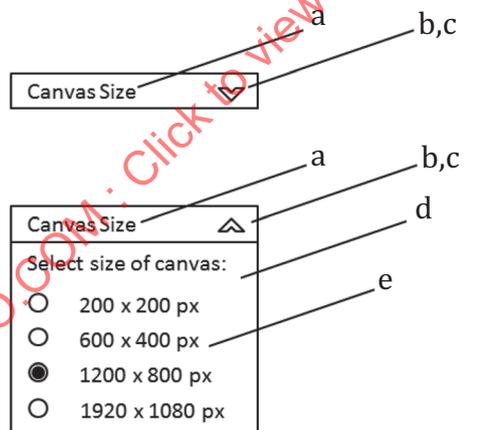
Key

a   label of container
b   means to collapse or expand container
c   indicator
d   canvas to display content of collapsible container
e   data elements or other user-interface elements

Figure 5 — Example of a collapsible container in collapsed status (above) and in expanded status (below)

8.5.3 States

A collapsible container can have the following states:

— collapsed/expanded.

8.5.4 When to use a collapsible container

A collapsible container is an applicable design solution if the following conditions are true:

— limited space;

— data belongs to the same semantic set;

— it is necessary to reduce complexity of the user interface.

8.5.5 How to use a collapsible container

The interaction that expands a collapsible container shall be the same as the interaction that closes the collapsible container.

8.6 Colour picker

8.6.1 Description

A colour picker is a user-interface element that presents selectable colours to the user.

8.6.2 Components

Components of a colour picker depend on the design solution and the data that is involved in the context. Common to all design solutions are the following:

a) selectable colours (either discrete or continuous choices);

b) indicator that visualizes the current selection;

c) history of previously chosen colours (optional).

8.6.3 States

A colour picker can have the following states:

— active/deactivated.

8.6.4 When to use a colour picker

A colour picker is an applicable design solution, if the following conditions are true:

— a large number of available colours (>16);

— complex colour settings available (e.g. saturation, brightness, hue);

— selecting multiple colours of a colour palette.

8.6.5 How to use a colour picker

— If the selectable colours are continuous choices, alternative picking methods should be available, e.g. pick a colour with a pointer device or entering RGB data.

— Within the context of an application, provision of previously selected colours or sets of colours (palettes) should be considered to optimize colour contrast for discriminability.

8.7 Combination box/combo box

8.7.1 Description

A combination box is a user-interface element that combines a text field with a list box and allows the user to type input into a text field or to select an option from a list box, which will fill in the text field.

NOTE   A combination box typically has a label (textual or graphical) that indicates the purpose of the combination box.

8.7.2 Components

A combo box consist of the following components:

a)   label;

b)   entry field;

c)   cursor;

d)   means to expand the list (list button);

e)   indicator, displaying expanded or collapsed status;

f)   selection cursor (when expanded);

g)   list (when expanded).

Key
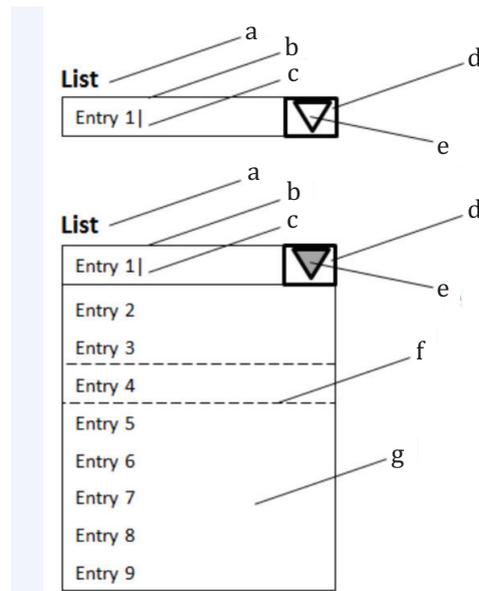
a   label

b   entry field

c   cursor

d   list button

e   indicator

f   selection cursor

g   list

Figure 6 — Example of a combo box in collapsed mode (above) and with expanded list (below)

8.7.3   States

A combination box can have the following states:

— focused/unfocused;

— active/deactivated;

— filled/empty;

— list collapsed/expanded.

8.7.4   When to use a combination box

A combination box is an applicable design solution when

— the desired result is the making of a selection or the setting of a state or assigning a value,

— choices are mutually exclusive,

— there is limited space,

— users need to see which option is currently selected,

— there are five or more items or the number of items may change over time,

— users may be able to type the entry more quickly than they can select it, and

— users may have to type values that cannot be supplied by the application.

8.7.5 How to use a combination box

A combination box shall only present selectable data.

Data presented within a combination should be presented in a persistent sorted order.

8.8 Cursor

8.8.1 Description

A visual indication of where the user interaction via keyboard (or keyboard emulator) will occur.

[SOURCE: ISO 9241-171]

8.8.2 Components

A cursor consists of the following element:

a) visual indicator, identifying where the interaction will occur.

Key
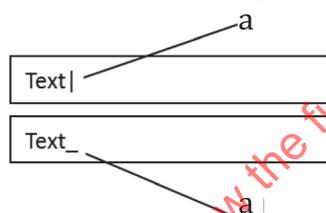
a   visual indicator

Figure 7 — Examples of different cursors positioned right of text in an entry field, vertical cursor (above), horizontal cursor (below)

8.8.3 States

The cursor as such has no states but can be used to visualize user interaction modes, such as insert mode and overwrite mode.

8.8.4 When to use a cursor

A cursor is an applicable design solution if the following condition is true:

— user edits/inputs alphanumeric information.

8.8.5 How to use a cursor

Inside text input elements, the cursor shall be used to highlight where the next interaction will occur.

8.9 Date picker

8.9.1 Description

A date picker is an element that displays days, months and years for selection by the user to define a specific date.

8.9.2 Components

Components of a date picker depend on the the design solution applied. Common to all design solutions are the following:

a)   identifiers visualizing the selected day, month and year;

b)   means to select the year, month, day;

c)   entry field with label (optional).

Key
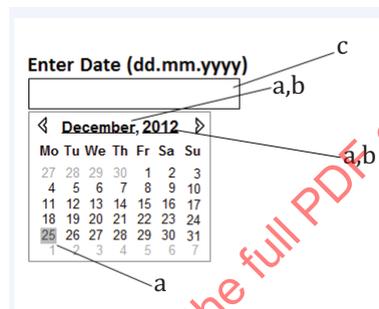
a   identifiers visualizing selected day, month and year

b   means to select year, month, day

c   entry field with label (optional)

Figure 8 — Example of a date picker, visualized as a calendar with identifier that visualizes the selected ("picked") date: December 25th, 2012, showing also adjacent days of previous and following month. Click on "December" accesses a list of available months to enable selection. Click on "2012" accesses a list of available years to enable selection. Click on the left arrow navigates to the previous, on the right arrow to the next month

8.9.3 States

A date picker can have the following states:

— active/deactivated;

— focused/unfocused.

Days displayed in the date picker can have the following states:

— active/deactivated;

— focused/unfocused;

— selected/unselected.

8.9.4   When to use a date picker

A date picker is an applicable solution when the following conditions are true:

—   a specific date is to be identified by the user;

—   only one date is to be identified by the user.

Alternative user-interface elements in case of more dates to be identified

—   consider **entry field**, or

—   consider **List box**.

8.9.5   How to use a date picker

Month and year should be independently selectable in a date picker.

The realization of a date picker should follow the date conventions of the local culture.

The user should be able to set a preferred date format.

NOTE   ISO 8601 gives guidance on date format.

8.10 Dialogue box

8.10.1   Description

A dialogue box is a subordinate type of form (typically presented separately) that supplements and/or supports the interaction that is taking place in the main application dialogue.

NOTE 1   If a message requires user input, it could be considered a dialogue box.

NOTE 2   A Dialogue box can also contain a collection of different user-interface elements.

NOTE 3   See ISO 9241-143 for guidance on design and use of dialogue box.

[Derived from ISO 9241-143:2012]

8.10.2   Components

Components of a dialogue box depend on the the design solution applied. Common to all design solutions are the following:

a)   title;

b)   information related to the application dialogue;

c)   push button with default action (optional with implicit designator);

d)   means to dismiss dialogue (optional, depending on the context).
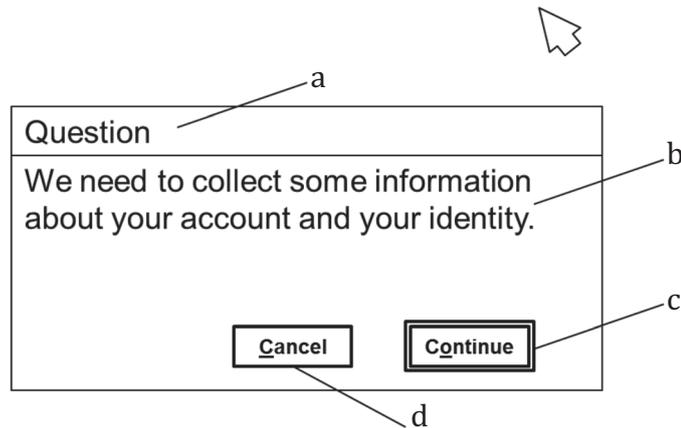
Key

a   title
b   information
c   push button with default action
d   means to dismiss dialogue

Figure 9 — Example of a dialogue box, indicating the default action and providing implicit designators for the push buttons

8.10.3 States

A dialogue box can have the following states:

— focused/unfocused;

— active/deactivated.

8.10.4 When to use a dialogue box

A dialogue box is an applicable design solution if the following conditions are true:

— explicit user decision, action or data entry required;

— question that needs to be answered that affects the whole application.

8.10.5 How to use a dialogue box

A dialogue box shall be presented/appear on the topmost visual layer.

The user should be able to hide or dismiss a dialogue box, whenever its presence is not contextually essential.

8.11 Dropdown list box

8.11.1 Description

A dropdown list box is a user-interface element that combines a field with a list box and allows the user to select an option from a list box, which will fill in the text field.

8.11.2 Components

The dropdown list box consists of the following components:

a)   label;

b)   data field, showing the current active selected data from the list;

c)   means to expand the list (list button);

d)   indicator, displaying expanded or collapsed status;

e)   list (when expanded).
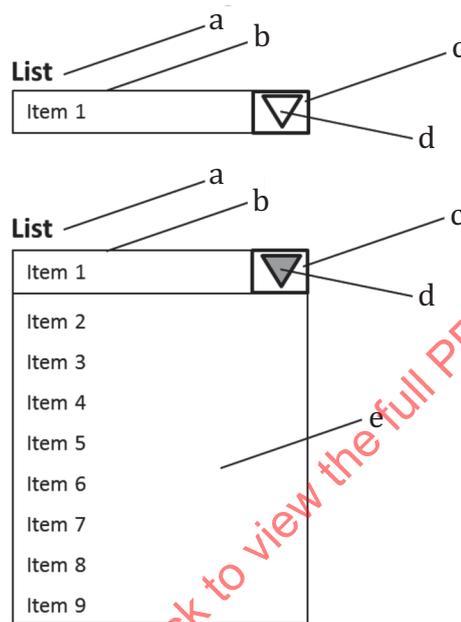
Key

a   label
b   data field
c   list button
d   indicator
e   list

Figure 10 — Example of a collapsed dropdown list box (above) and expanded dropdown list box (below)

8.11.3 When to use a dropdown list box

A dropdown list box is an applicable design solution when the following conditions are true:

—   the desired result is the making of a selection or the setting of a state;

—   choices are mutually exclusive;

—   there is limited space;

—   except when changing the selection, users need to see only the item which is currently selected;

—   there are four or more items or the number of items may change over time;

—   all values can be supplied by the application.

8.11.4 How to use a dropdown list box

When a dropdown list is inactive, it should display a single value.

When a dropdown list box is activated, it should display (drops down) a list of values, from which the user may select one.

When the user selects a new value, the control shall revert to its inactive state, displaying the selected value.

8.12 Entry field/input field

8.12.1 Description

An entry field is a field in which users can input or edit data.

NOTE 1   Entry fields can be optional fields or mandatory (required) fields, for the user.

NOTE 2   Content can be prepopulated with a default value relevant to the task.

NOTE 3   Contrast with text field and ISO 9241-12:1998.

[SOURCE: ISO 9241-143:2012]

8.12.2 Components

An entry field consists of the following components:

a)   label;

b)   field (with boundaries);

c)   cursor (within boundaries).
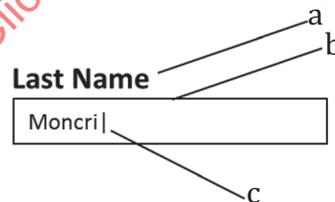
Key
a   label
b   field
c   cursor

Figure 11 — Example of an entry field with vertical cursor

8.12.3 States

An entry field can have the following states:

— focused/unfocused;

— active/deactivated;

— filled/empty.

8.12.4 When to use an entry field

An entry field is an applicable design solution when the following condition is true:

— system needs information from the user that cannot be presented as predefined choices.

8.12.5 How to use an entry field

The boundaries of an entry field shall be clearly visualized (e.g. by contrast, colour or box).

8.13 Entry field with dialogue button

8.13.1 Description

An entry field with dialogue button is a combination of entry field element and push button element where the push button functionality is executed on information that has been entered in the entry field.

8.13.2 Components

An entry field with dialogue button consists of the following components:

a)   label;

b)   field (with boundaries);

c)   cursor (within boundaries);

d)   push button;

e)   contextual information (e.g. data entry is mandatory; optional).
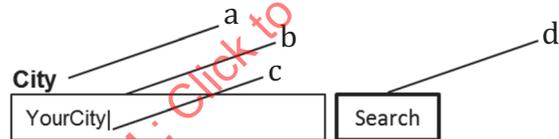
Key
a   label
b   field
c   cursor
d   push button

Figure 12 — Example of an entry field with a dialogue button

8.13.3 States

An entry field with dialogue button can have the states of an entry field and a button in all combinations.

8.13.4 When to use an entry field with dialogue button

The entry field with dialogue button is an applicable design solution when the following condition is true:

— mutually exclusive alphanumeric input is required for a system operation and the input cannot be presented as predefined options.

8.13.5   How to use an entry field with dialogue button

The boundaries of an entry field shall be clearly visualized (e.g. by contrast, colour or box).

8.14   Geographical map

8.14.1   Description

A user-interface element that presents geographical information and enables interaction with different geographical elements.

8.14.2   Components

A geographical map consists of the following components:

a)   label;

b)   geographical information

    1)   graphical representation of areas and regions;

    2)   textual information of areas and regions (optional);

    3)   visual information related to geographical properties (optional);

c)   legend (optional).
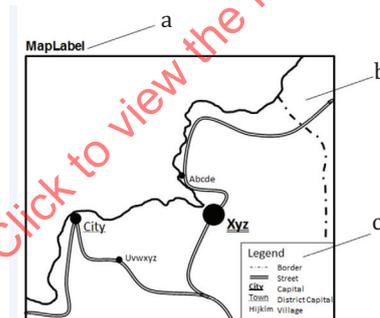
Key
a   label
b   geographical information
c   legend (optional)

Figure 13 — Example of a geographical map

8.14.3   States

A geographical map can be active or deactivated.

8.14.4   When to use a geographical map

A graphical map is an applicable design solution if the following condition is true:

— geographical information is to be presented.

8.14.5 How to use a geographical map

A geographical map should implement geographical conventions if applicable,

— north up,

— clear distinction of different geographical areas, and

— enable zoom in and zoom out.

8.15 Group/group box

8.15.1 Description

A group is a user-interface element that visualizes that information items or user-interface elements belong to the same semantic set.

A group should have a distinctive label that describes the semantic set of user-interface elements.

8.15.2 Components

Groups consist of the following components:

a) label;
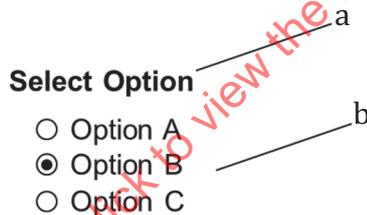
b) data elements.

Key
a label
b data elements

Figure 14 — Example of a group of three dependent radio buttons

8.15.3 States

A group can have the following states:

— filled/empty;

— visible/invisible.

NOTE   Typically, in dynamically created user interfaces, a group is rendered invisible if it is empty.

8.15.4 When to use a group

If all of the following conditions are true, a group is an applicable design solution:

— a visualization of the semantic grouping is required, or there are so many user-interface elements that organizing them in groups increases the efficiency of information consumption;

— semantic sets of user-interface elements can be clearly identified;

— groups should contain more than one element.

In case that one or multiple conditions are not met

— there are so few user-interface elements that no visual structuring is required,

— consider layout without using it,

— there is not sufficient space to clearly identify each group,

— consider sections in an accordion, and

— consider collapsible container.

8.15.5 How to use a group

A group shall be presented in a way that clearly depicts the relation of all elements of that group.

NOTE   This presentation is often realized by close proximity of those elements (using the law of proximity) or a visual indicator (frame around elements of a group; common background design of the area of the group).

8.16 Handle

8.16.1 Description

A handle is a user-interface element that identifies areas of visual information that can be used to manipulate properties of that information.

8.16.2 Components

A handle consists of the following component:

a)   visual indicator.

a

Key
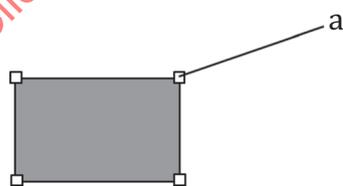
a   visual indicator

Figure 15 — Example of handles, used in all four corners of a visual object in rectangle shape

8.16.3 States

A handle can have the following states:

— visible/invisible;

— active/deactivated;

— pressed/not-pressed.

NOTE 1   Typically, visibility and focus are coupled: the handle becomes visible when the corresponding area of visual data also has the keyboard focus, and disappears when the focus is transferred somewhere else.

NOTE 2   Typically, if a pointer is available, hovering over a handle changes the pointer to indicate possible interaction.

8.16.4 When to use a handle

A handle is an applicable design solution if one or all of the following conditions are true:

— a pointer device is available;

— properties of displayed elements can be changed in a linear way.

In the case that one or both conditions are not met

— consider using a properties dialogue.

8.16.5 How to use a handle

A hande shall be visualized in a way that enables a clear differentiation between handle and the element that is to be manipulated.

Alternative means to manipulate properties of an element shall be provided.

EXAMPLE   The size of a rectangle can be changed by a pointer device, using a handle in the lower right corner of that rectangle. Also, a properties dialogue is available, where data can be entered to change the size of the rectangle.

8.17 Hierarchical list/tree view/tree lists

8.17.1 Description

A hierarchical list is a series of lists which are structured in a hierarchical or "tree-like" manner, where the selection of the initial item leads to another list containing additional items which may lead to another list, etc., until the desired item is located.

[SOURCE: ISO 9241-143:2012]

8.17.2 Components

Components of a hierarchical list depend on the design solution. Common components are the following:

a)   label;

b)   data elements;

c)   indicators that visualize the hierarchical position in the data structure;

d)   means to expand or collapse levels of the hierarchy (optional).

Key
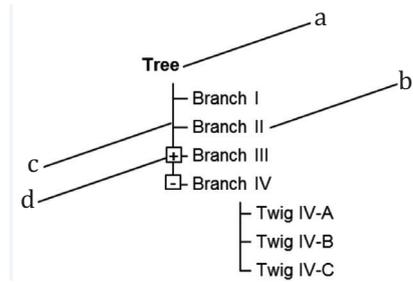
a   label

b   data elements

c   indicators

d   means to expand or collapse levels of hierarchy (optional)

Figure 16 — Example of a hierarchical structure in a list of data elements

8.17.3 States

A hierarchical list can have the following states:

— filled/empty.

The nodes of the hierarchical list can be

— collapsed/expanded.

8.17.4 When to use a hierarchical list

A hierarchical list is an applicable design solution if the following conditions are true:

— when ordering data elements that have an inherent hierarchical structure/relation, e.g. file system, headings in a document;

— all data elements do not have to be visible at all times.

8.17.5 How to use a hierarchical list

A hierarchical list shall be visualized in a way that indicates the hierarchical position of a node and data.

Data within a node should be ordered in an appropriate way.

8.18 Implicit designator

8.18.1 Description

An implicit designator is a portion of an option name or control label used for keyboard or keyboard equivalent selection.

[Derived from ISO 9241-171:2008, 3.16]

8.18.2 Components

An implicit designator consists of the following component:

a)   indicator that visualizes the available keyboard selection.
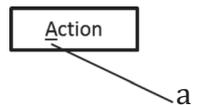
Key

a   indicator

Figure 17 — Example of an implicit designator in a push button, showing an underlined "A" in the label "Action" that keystroke on the letter "A" will activate the action of that push button

8.18.3 States

An implicit designator inherits its state from the corresponding functional element.

In addition: Visible/invisible.

8.18.4 When to use an implicit designator

An implicit designator is an applicable design solution if the following condition is true:

— a keyboard is an available input method.

8.18.5 How to use an implicit designator

An implicit designator shall be realized in a way that makes it detectable and distinctive from the respective user-interface element.

An implicit designator shall have an explicit key mapping.

An implicit designator shall be realized in a way that clearly identifies the relation of the implicit designator and the respective user-interface element.

Within an interactive system, the access to the implicit designator should be consistent.

8.19 Instructive information

8.19.1 Description

Instructive Information is a user-interface element that provides users with additional guiding information on how to use the user interface, a user-interface element or the context of use.

8.19.2 Components

Instructive information consists of the following component:

a)   information in alphanumeric and/or graphical form.
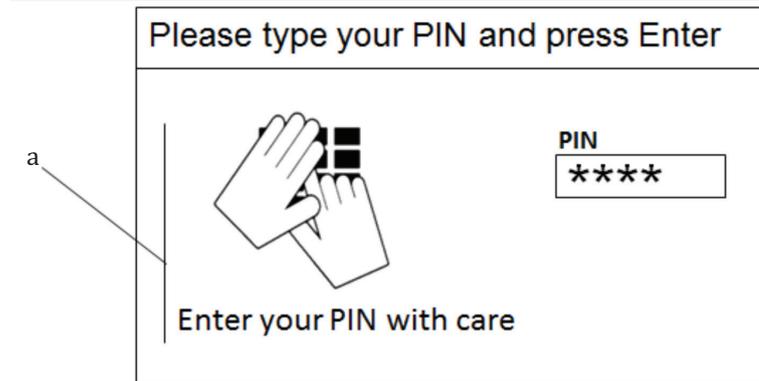
Key

a   information in alphanumeric and/or graphical form

Figure 18 — Example of instructive information

8.19.3 States

Instructive information can have the following states:

— visible/invisible

8.19.4 When to use instructive information

Instructive information is an applicable design solution if one or the following conditions are true:

— when further information is useful to understand the use of an interactive system, the context of use and its environment;

— when further information is needed beyond that provided in the title, label or other user-interface elements;

— when considering accessibility.

8.19.5 How to use instructive information

Every user-interface element and user interface design should integrate instructive information to the greatest extent possible.

The provision of instructive information should not be used as an alternative of good user interface design.

If instructive information is applied, it shall be available for assistive technologies.

8.20 Input tokenizer

8.20.1 Description

An input tokenizer is information in an entry field that is interpreted by the system to point to meta-information.

EXAMPLE   In an e-mail application, a person's name points to her mail address and contact data.

8.20.2 Components

An input tokenizer consists of the following components:

a) label;

b) entry field;

c) labelled token;

d) means to turn entered information into token.

Key
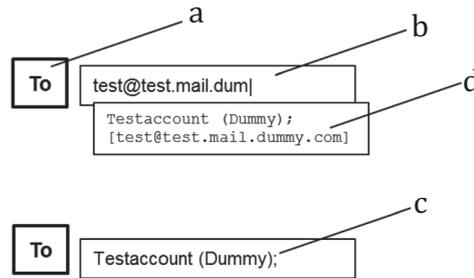
a   label

b   entry field

c   labelled token

d   means to turn entered information into token

Figure 19 — Example of an input tokenizer with data entered (above) and the related token (below), where the label is used in a push button and an entered email address is translated in a token

8.20.3 States

An input tokenizer can have the following states:

— focused/unfocused;

— active/deactivated;

— filled/empty.

NOTE   Before being turned into an input tokenizer, input text appears and behaves as ordinary text in an entry field.

8.20.4 When to use an input tokenizer

An input tokenizer is an applicable design solution if the following condition is true:

— a link to a complex set of data are required.

8.20.5 How to use an input tokenizer

An input tokenizer should be designed in a way that clearly identifies its nature.

NOTE   An input tokenizer is not editable.

8.21 Label

8.21.1 Description

A label is a short descriptive title for an entry or read only field, table, control, or other user-interface element.

NOTE   In some applications, labels are classified as protected fields.

[SOURCE: ISO 9241-12:1998, 3.9]

8.21.2 Components

A label consists of the following component:

a)   text or graphical element.
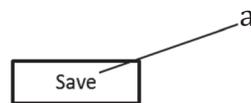
Key

a   text or graphical element

Figure 20 — Example of a label in a push button, indicating the function in a short and concise manner

8.21.3 States

A label can have the following states:

— visible/invisible.

NOTE 1   Invisible labels need to be accessible to assistive technology, such as screen readers.

A label should only be invisible if enough context is presented in the user-interface to make it obvious.

8.21.4 When to use a label

— Every user-interface element except Pointer shall have a label.

NOTE   Having a label for every user-interface element is essential to enable assistive technologies.

8.21.5 How to use a label

Labels should be short and concise.

The design of the label shall be realized in a way that support both legibility and readability.

The label should be visualized in a way to distinguish its role as a label as distinct from other text.

8.22 Legend/chart key

8.22.1 Description

A legend is a user-interface element that describes visualized data.

8.22.2 Components

The components of a legend depend on the design solution. Common to all design solutions are the following:

a) label of the legend;

b) key visual to the visualized data;
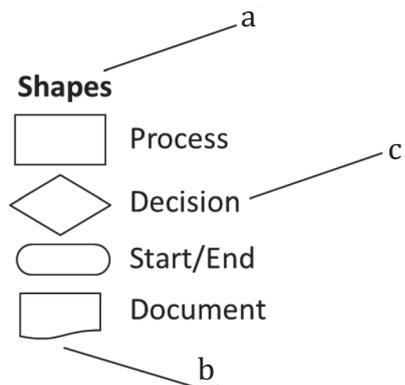
c) label of data.

Key

a   label of the legend

b   key visual to the visualized data

c   label of data

Figure 21 — Example of a legend, showing some flow chart shapes and their meaning

8.22.3 States

A legend can have the following states:

— visible/invisible.

NOTE   Invisible legends need to be accessible to assistive technology, such as screen readers.

8.22.4 When to use a legend

A legend is an appropriate design solution if the following conditions are true:

— data elements is to be explained;

— coding of data elements is to be explained.

8.22.5 How to use a legend

A legend shall present information for all visualized data.

8.23 Link/hyperlink

8.23.1 Description

A user-interface element that allows navigation to a specified location within an interactive system.

[Derived from ISO 14915-2:2003]

8.23.2 Components

A link consists of the following components:

a) alphanumerical or graphical information;

b) visual indicator to declare that information to be a link;

c) reference address of the specified location.

> The novel is based on some events during the last days
> of the _Roman Empire_, especially in the nobility around
> _Marc Antony_ and its opponent _Octavian_, who later
> became the first roman emperor
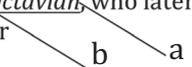
Key

a   alphanumerical or graphical information
b   visual indicator

Figure 22 — Example of links, indicating more information available at the target references (the reference address is invisible)

8.23.3 States

A link can have the following states:

— focused/unfocused;

— active/deactivated.

NOTE   Where feasible, links can also have the states unvisited/ visited, in order to communicate to the user that the link has already been followed.

8.23.4 When to use a link

A link is an applicable design solution if one or all of the following conditions are true:

— navigation to another page;

— navigation to another service;

— reference description is longer than a label.

In case that one or multiple conditions are not met

— consider push button.

8.23.5 How to use a link

A link should be designed in a way that differentiates it clearly from regular text or other user-interface elements.

NOTE 1   A link is very often designed by using a dedicated text colour and/or underlined text and/or font. A link may be activated by the system, based on time, system condition, etc.

NOTE 2   Further information on the use of links is contained in ISO 14915-2:2003, Clause 8.

8.24 List box

8.24.1 Description

A list box is a user-interface element that provides a presentation (usually vertical) of items from which the user can select only one (single selection) or can select more than one (multiple selection).

NOTE   The items in the list can be represented by a text and/or graphic label.

[Adopted from ISO 9241-143:2012]

8.24.2 Components

Components of a list box depend on the the design solution applied. Common to all design solutions are the following:

a)   label;

b)   data items;

c)   canvas to display data items (with boundaries);

d)   scroll bar (in case of longer lists; optional).

Key

a   label

b   data items

c   canvas

Figure 23 — Example of a list box with eight entries

8.24.3 States

A list box can be active or deactivated.

List items can have the following states:

—   focused/unfocused;

—   selected/unselected.

8.24.4 When to use a list box

A single selection list should be used as appropriate when:

— the desired result is the making of a selection or the setting of a state;

— choices are mutually exclusive;

— there is adequate space to display three or more items simultaneously without scrolling;

— there are more than five items or the number of items may change over time;

— there is value in having a number of the choices simultaneously visible;

— the number of items might change dynamically.

NOTE   If a scroll bar will be used with the list box, it is important that the list box is large enough to show at least three items.

A multiple selection list should be used as appropriate when:

— the desired result is the making of a selection or the setting of one or more states;

— choices are not mutually exclusive;

— there is adequate space to display three or more items simultaneously without scrolling;

— there are more than five items or the number of items may change over time;

— there is value in having a number of the choices simultaneously visible;

— the number of items might change dynamically.

8.24.5 How to use a list box

The boundaries of a list box shall be clearly visualized (e.g. by contrast, colour or box).

8.25 List button/menu button

8.25.1 Description

A list button is a user-interface element used to access a list – when the button is pressed, a drop down list of items is displayed.

NOTE   The list button may contain the name of the currently selected list item and has a graphic (usually a triangle) indicating that additional selections are available.

[SOURCE: ISO 9241-143:2012]

8.25.2 Components

a)   list box;

b)   push button;

c)   indicator, displaying expanded or collapsed status.
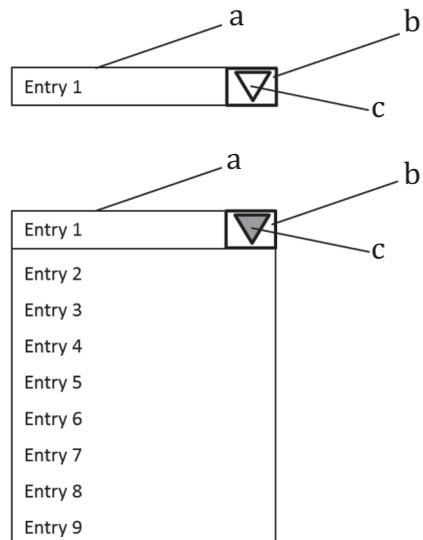
Key

a   list box

b   push button

c   indicator

figure 24 — Example of a list button (on the right of a list box), list not expanded (above), and with list expanded (below)

8.25.3 States

A list button can have the following states:

— active/deactivated;

— focused/unfocused.

8.25.4 When to use a list button

A list button is an applicable design solution when one or both of the following conditions are true:

— limited space for a display of a list box;

— multiple lists are to be displayed.

8.25.5 How to use a list button

The list button shall be visualized in a way that communicates the relation to the list that is to be accessed.

8.26 Menu/menu bar

8.26.1 Description

A menu is a user-interface element that represents a set of selectable options giving access to objects or actions.

NOTE   A menu primarily provides access to functions and/or specified locations within an interactive system.

8.26.2 Components

Components of a menu depend on the the design solution applied. Common to all design solutions are the following:

a) Menu items

— Labels, corresponding to each selectable objects and/or action;

— Selectable objects and/or actions;

b) Title of the menu (optional);

c) Instructional information (optional).
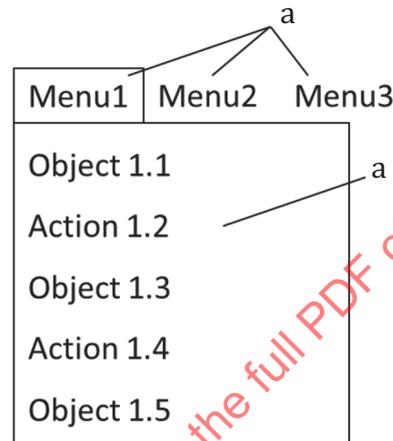
Key

a   labels/selectable objects

Figure 25 — Example of an hierarchical menu, showing, on first level, three objects that enable access to next level of menu (in this case, the first object of that first hierarchical level is selected, showing the second-level options and actions)

8.26.3 States

A menu can have the following states:

— visible/invisible;

— active/deactivated.

Menu items can have the following states:

— active/deactivated;

— focused/unfocused;

— checked/unchecked/conditionally checked.

8.26.4 When to use a menu

A menu is an applicable design solution if one or more of the following conditions are true:

— user selects a system function;

— user navigates to a specified location.

If conditions are not true

— consider **list box**,

— consider **combo box**, or

— consider **carousel**.

8.26.5 How to use a menu

Functions and segments shall be grouped according to their context.

Depending on the context on use, a menu should give contextual help on how to use the menu (e.g. **tool tip,** Text).

A menu may offer access to a secondary set of selectable options or actions, e.g. by a cascading presentation.

8.27 Output pane

8.27.1 Description

An output pane is a user-interface element that displays variable information.

NOTE   Displayed information can be of different types. The user cannot edit this displayed information.

8.27.2 Components

An output pane consists of the following components:

a)   label;

b)   data.

a   b

Current Temperature: **25° C**

Key
a   label
b   data

Figure 26 — Example of an output pane displaying temperature

8.27.3 States

An output pane can be visible or invisible.

8.27.4 When to use an output pane

An output pane is an applicable design solution if the following condition is true:

— non-editable data are to be displayed.

8.27.5 How to use an output pane

The output pane shall be realized in a way that clearly depicts the non-editable nature of the displayed data.

8.28 Pointer

8.28.1 Description

A pointer is a graphical symbol that is moved on the screen according to operations with a pointing device.

NOTE   Users can interact with elements displayed on the screen by moving the pointer to that location and starting a direct manipulation.

[SOURCE: ISO 9241-16:1999, 3.15]

8.28.2 Components

A pointer consists of following component:

a)   graphical symbol.

a

Key

a   graphical symbol

Figure 27 — Conceptual visualization of a pointer, using an arrow to indicate position on the screen

8.28.3 States

The pointer as such has no states.

NOTE 1   A corresponding symbol can be used to visualize various user interaction modes, such as draw, drag and drop, resize, etc.

NOTE 2   The pointer symbol can be used to indicate that an application is currently not able to receive user input. However, the user can still choose a different application.

NOTE 3   Manipulation events initiated by the user (e.g. clicking a button) are to be visualized in the affected control (the button changing into pressed state).

8.28.4 When to use a pointer

A pointer is an appropriate design solution if the following condition is true:

—   a pointing device is available.

8.28.5 How to use a pointer

A pointer shall be designed in a way to be clearly detectable and distinguishable from other user-interface elements and content of the interactive system.

The system should provide means for the user to individualize the speed of the pointing device to adopt movements of the pointer to user preferences.

8.29 Pop-up menu/contextual menu

8.29.1 Description

A pop-up menu is a menu with contextual functions, presented in the foreground of the user interface and close to the current cursor position.

8.29.2 Components

A pop-up menu consists of the following components:

a) Menu items:

— labels, corresponding to each selectable objects and/or action;

— selectable objects and/or actions.

b) Canvas to display menu items (within boundaries).
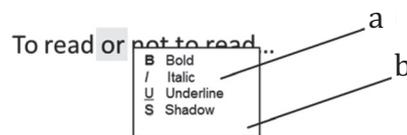
Key

a   menu items

b   canvas to display menu items

Figure 28 — Example of a pop-up menu showing options to change text style of selected text as an overlying canvas on top of text

8.29.3 States

A pop-up menu can visible or invisible.

Menu items can have the following states:

— focused/unfocused;

— active/deactivated;

— selected/unselected;

— checked/unchecked/conditionally checked.

8.29.4 When to use a pop-up menu

A pop-up menu is an appropriate design solution if one or both of the following conditions are true:

— contextual functions are available;

— contextual functions are to be executed on highlighted or selected data.

8.29.5 How to use a pop-up menu

A pop-up menu shall be realized in a way that clearly differentiates the pop-up menu from the underlying user interface/data.

Contextual functions shall be accessible by other means than a pop-up menu.

8.30 Progress indicator

8.30.1 Description

A progress indicator is a user-interface element that visualizes the status of an operation or process.

NOTE   Status can be affected by time, amount of data or progess of processing an operation.

8.30.2 Components

Components of a progress indicator depend on the the design solution applied. Common to all design solutions are the following:

a)   label;

b)   visualization of progress status;

c)   data on extent of progress (optional).
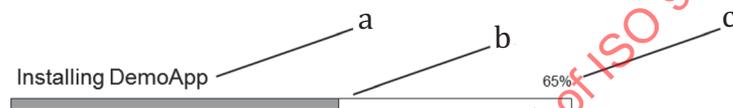
Key

a   label
b   visualization of progress status
c   data on extent of progress (optional)

Figure 29 — Example of a progress indicator, also displaying extent of progress as percentage

8.30.3 States

A progress indicator does not have special states.

8.30.4 When to use a progress indicator

A progress indicator is an applicable design solution if the following condition is true:

—   user is to be informed about ongoing process;

—   the progress of a process is of semantic and/or contextual interest to the user.

8.30.5 How to use a progress indicator

A progress indicator should be designed in a way that it is not mistaken for a Slider.
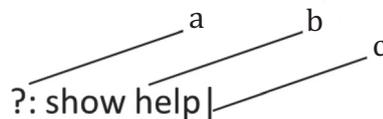
8.31 Prompt

8.31.1 Description

A prompt is a user-interface element that requires the user to enter a command.

8.31.2 Components

The prompt consists of the following component:

a)   label;

b)   entry field;

c)   cursor;

d)   means to execute the entered command (optional).

Key

a   label

b   entry field

c   cursor

Figure 30 — Example of a prompt, showing an entered command to display help information. The means to execute the entered command are not visible, that command would be executed by confirmation on keypress on the "Enter" key

8.31.3   States

A prompt can be active or deactivated.

8.31.4   When to use a prompt

A prompt is an appropriate design solution to enable a command entered by the user.

8.31.5   How to use a prompt

A prompt shall be visualized in a way that indicates where the user input takes place.

8.32   Push button/command button

8.32.1   Description

A push button is a user-interface element used for executing an immediate command or action.

NOTE 1   A push button typically contains a label, which can be textual and/or graphical.
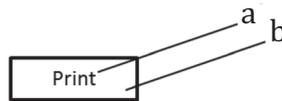
NOTE 2   Sometimes a push button is referred to as command button.

[Adopted from ISO 9241-143:2012]

8.32.2   Components

A push button consists of the following components:

a)   label, describing the command or action;

b)   canvas, to present the label

Key

a   label

b   canvas

Figure 31 — Example of a push button

8.32.3 States

A push button can have the following states:

— active/deactivated;

— focused/unfocused;

— pressed/not pressed.

8.32.4 When to use a push button

A push button is an applicable design solution to initiate a system function.

8.32.5 How to use a push button

Every push button shall have a label.

A push button label should be short and concise.

If additional Information is needed in the system function that is to be initiated, the design of the push button should reflect that.

Push buttons shall be activated by a single activation event, e.g. mouse click, touch; not by double click.

NOTE   Further information on push buttons applied in a multimedia context is contained in ISO 14915-2:2003, Clause 11.

8.33 Radio button

8.33.1 Description

A radio button is a user-interface element used to select an option from a group of mutually exclusive options, which has a (usually text) label that indicates what the option is and an indicator (e.g. graphic dot) that indicates whether or not the option is set.

NOTE 1   Radio buttons exist in groups used for a "one of many" choice, where exactly one option in the group may be selected at any time, and selecting one results in the deselecting of any different option that may have been selected before.
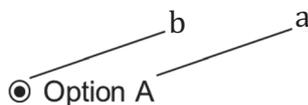
NOTE 2   Contrast with **check box**.

[SOURCE: ISO 9241-143:2012]

8.33.2 Components

A radio button consists of the following components:

a) label;

b) indicator.

b   a

⊙ Option A

Key
a   label
b   indicator

Figure 32 — Example of a radio button (option set) and the label on the right of the indicator

8.33.3 States

A radio button can have to following states:

— active/deactivated;

— focused/unfocused;

— editable/display only;

— checked/unchecked/conditionally checked.

NOTE 1   Checking one radio button in a group unchecks any other checked one.

NOTE 2   Before the user has made a choice, all radio buttons in a group can be unchecked.

8.33.4 When to use a radio button

If all of the following conditions are true, a radio button box is an applicable design solution:

— choices are exclusive, only one choice can be selected by the user;

— maximum number of choices should not exceed 10 unless choices are structured in groups or sets;

— the number of choices is static;

— the options are non-numeric;

— all options need to be considered at once.

In case that one or multiple conditions are not met:

— choices are non-exclusive, none or more choices can be selected by the user

　　— consider **check box**;

— maximum number of choices exceeds 10

　　— consider list box/combination box;

— the number of choices is dynamic

　　— consider list box/combination box;

— the options are numeric

　　— consider stepper/spin button/slider;

— not all options needed to be considered at once

　　— consider list **box/spin button/slider**.

8.33.5 How to use a radio button

Radio buttons that share the same connotation shall be presented in a way that clearly depicts the relation.

NOTE　　This relation is often realized by showing dependent choices in a grouping element and/or using the law of proximity.

A radio button box shall not convey other states than checked and unchecked.

The different status of a radio button shall be realized in a way that clearly differentiates the states.

8.34 Read only field/protected field
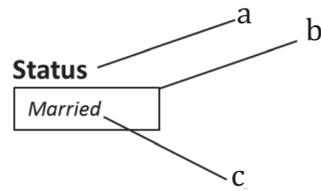
8.34.1 Description

A read only field is a field that contains data that cannot be modified by the user.

[SOURCE: ISO 9241-143:2012]

8.34.2 Components

The read only field consists of the following components:

a)　label;

b)　entry field;

c)　data.

Key

a   label

b   entry field

c   data

Figure 33 — Example of a read only field, showing visual difference to an entry field by displaying alphanumeric data in italics

8.34.3 States

A read only field can have the following states:

— focused/unfocused;

— filled/empty.

8.34.4 When to use a read only field

A read only field is an applicable design solution if alphanumeric data are to be presented that contextually is not to be edited.

8.34.5 How to use a read only field

A read only field shall be able to take focus to enable reading content of that field by assistive technology.

The design of the read only field should clearly depict the read-only nature of the element.

8.35 Scroll bar

8.35.1 Description

A scroll bar is a user-interface element that allows a user to view objects that extend beyond the size of a displayed related window or list by moving them into or out of the available display area.

NOTE   A scroll bar for an area indicates whether or not additional information is available and may also indicate the relative position of the displayed part of the information.
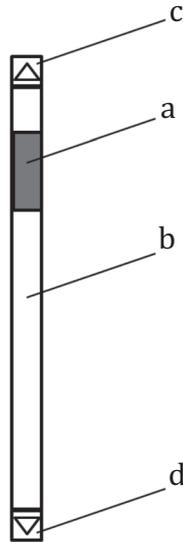
[SOURCE: ISO 9241-143:2012]

8.35.2 Components

A scroll bar has the following components:

a)   scroll handle;

b)   slide track;

c)   up button;

d)   down button.

Key

a   scroll handle

b   slide track

c   up button

d   down button

Figure 34 — Example of a vertical scroll bar

8.35.3   States

A scroll bar can be visible or invisible.

Alternatively, the scroll handle and up and down buttons can be active or deactivated.

NOTE   Hiding the scroll bar altogether is preferable to deactivating its elements.

8.35.4   When to use a scroll bar

A scroll bar is an applicable design solution when one or more of the following conditions are true:

— a list is to be presented;

— limited space available to present a list;

— the size of list exceeds visible area on screen.

8.35.5   How to use a scroll bar

A scroll bar should be available when the corresponding list gets input focus.

## 8.36   Scroll handle/elevator/scroll box/thumb

8.36.1   Description

A scroll handle is a user-interface element within a scroll bar that allows the user to move to a specific region of a displayed file by dragging the rectangle to the appropriate location within the scrollbar (shaft).

[SOURCE: ISO 9241-143:2012]

8.36.2 Components

A scroll handle has the following components:

a)   handle (within scroll bar).
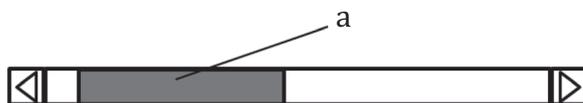
Key

a   handle

Figure 35 — Example of a scroll handle in a horizontal scroll bar

8.36.3 States

A scroll handle can be active or deactivated.

8.36.4 When to use scroll handle

Whenever a scroll bar has been implemented, a scroll handle shall be realized for the respective scroll bar.

8.36.5 How to use scroll handle

The position of the scroll handle on the scroll bar should represent the position of the visual area of the displayed, scrolled content.

The display size of handle should be relative to its view's current size.
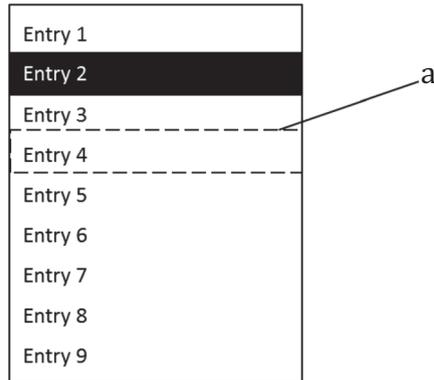
8.37 Selection cursor

8.37.1 Description

A selection cursor indicates the item whose selection state can be changed by a defined confirming action.

[SOURCE: ISO 9241-143:2012]

8.37.2 Components

A selection cursor has the following component:

a)   visual indicator, that shows the current position of the selection.

Key

a   visual indicator

Figure 36 — Example of a selection cursor, selection cursor position on entry 4, entry 2 being already selected

8.37.3 States

A selection cursor can be active or deactivated.

NOTE   In some contexts, it is more appropriate to communicate the deactivated state by rendering the selection cursor invisible.

8.37.4 When to use a selection cursor

A selection cursor is an applicable design solution if the following condition is true:

—   keyboard as input device is available.

8.37.5 How to use a selection cursor

A selection cursor shall be designed in a way that clearly differentiates it from content, data or other visual user-interface elements.

If a keyboard is available, the user should be able to navigate a selection cursor by arrow and tab keys.

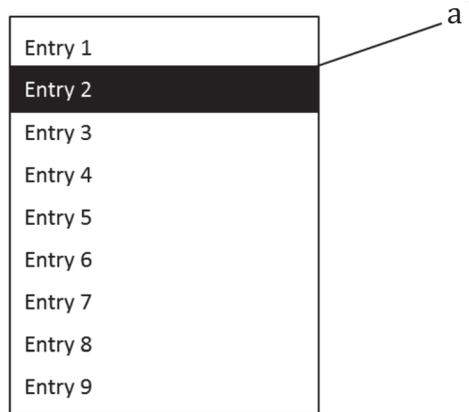## 8.38 Selection indication

8.38.1 Description

A selection indication is a visual or other cue that indicates the selected element on the display, to which the user may apply a subsequent action.

[SOURCE: ISO 9241-16:1999]

8.38.2 Components

A selection indication has the following component:

a)   visual indicator, that shows the current selection.

Key

a   visual indicator

Figure 37 — Example of a selection indication, with entry 2 selected

8.38.3 States

A selection indication can be visible (indicating a selection) or invisible.

NOTE   If the respective user-interface can be active/disabled, the selection indication may also have to reflect those states.

8.38.4 When to use a selection indication

A selection indication is an applicable design solution if the following condition is true:

— a visual user-interface element can be selected.

8.38.5 How to use a selection indication

A selection indication shall be designed in a way that clearly differentiates it from content, data or other visual user-interface elements.

If a keyboard is available, activation of the selected information should be released by pressing the Return key.

8.39 Selection list/choice list

8.39.1 Description

A selection list is a user-interface element presented as a list containing a number of items from which a user can select.

NOTE 1   Single selection or multiple selection can be possible. The number of items can be fixed or can change during the dialogue.
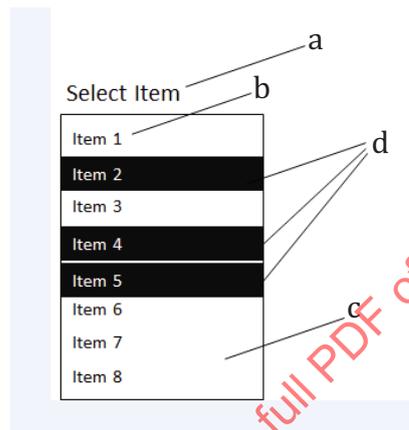
NOTE 2   Selection lists are often presented in a box, i.e. list box.

[SOURCE: ISO 9241-143:2012]

8.39.2 Components

A selection indication has the following component:

a) label;

b) data items;

c) canvas to display data items (with boundaries);

d) selection indicators;

e) scroll bar (in case of longer lists; optional);

Key

a   label
b   data items
c   canvas to display data items
d   selection indicators

Figure 38 — Example of a multiple selection list, with entries 2, 4 and 5 selected

8.39.3 States

A selection list can have the following states:

— active/deactivated;

— filled/empty.

8.39.4 When to use a selection list

A selection list is an applicable design solution if all of the following conditions are true:

— more than 10 options are available to choose;

— number of choices is not static/can vary.

8.39.5 How to use a selection list

If multiple selections are possible, this capability should be made clear to the user.

A selected list item should be presented in a way that clearly differentiates it from unselected ones.
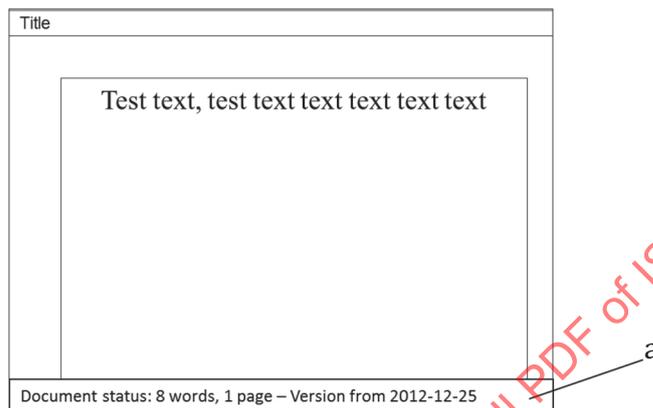
8.40 Status information

8.40.1 Description

Information that is used to describe a status.

8.40.2 Components

A selection indication has the following component:

a)   output pane.

Key

a   output pane

Figure 39 — Example of status information in a text editing application, displaying information on the amount of written words, the amount of used pages and the saving date of the actual document

8.40.3 States

Status information can be visible or invisible.

8.40.4 When to use status information

If available, status information should be presented on demand from the user.

8.40.5 How to use status information

A status information should reference the context of use from which it was accessed or in which it is presented.

Status information shall be visualized in a way that differentiates it from other visualized data.

8.41 Stepper/spin button

8.41.1 Description

A stepper is a user-interface element that allows the user to move through the available alternatives, options, or values, one at a time.

NOTE 1   A stepper typically has an area that displays the current value and one or two graphical elements that allow the user to move through the options.

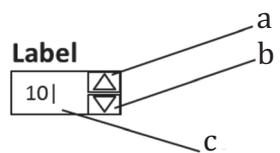NOTE 2   They can be combined with data entry fields.

NOTE 3   There are various types of stepper elements including spin buttons, cycle buttons, little arrows and sliders.

[SOURCE: ISO 9241-143:2012]

8.41.2 Components

A selection indication has the following component:

a)   increment push button;

b)   decrement push button;

c)   output pane.

Key

a   increment push button
b   decrement push button
c   output pane

Figure 40 — Example of a stepper with an entry field (that is also used as output pane)

8.41.3 States

A stepper can be active or deactivated.

When combined with a data entry field, active and deactivated states of both UI elements should be synchronized.

8.41.4 When to use a stepper

A stepper is an applicable design solution if the following condition is true:

—   the list of available data are continuous.

8.41.5 How to use a stepper

If the list of available data exceeds 10, an entry field should be used in conjunction with a stepper button.

8.42 System message

8.42.1 Description

A system message is a user-interface element that provides messages of the interactive system for the user.