INTERNATIONAL STANDARD

ISO 15187

First edition
2000-11-01

Manipulating industrial robots — Graphical user interfaces for programming and operation of robots (GUI-R)

Robots manipulateurs industriels — Interfaces graphiques utilisateur pour la programmation et l'utilisation de robots (GUI-R)

Reference number
ISO 15187:2000(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

International Standard ISO 15187 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 2, *Robots for manufacturing environment*.

Annex A of this International Standard is for information only.

Introduction

ISO 15187 is part of a series of International Standards dealing with manipulating industrial robots. Other International Standards cover such topics as safety, general characteristics, co-ordinate systems, performance criteria and related test methods, terminology, and mechanical interfaces. It is noted that these standards are interrelated and are also related to other International Standards.

Annex A provides examples of possible platform with GUI for open architecture.

The purpose of this International Standard is to standardize the use of graphical user interfaces for robot programming and operation (called GUI-R). Important customer requirements for programming and operation of robots are

— simplification and

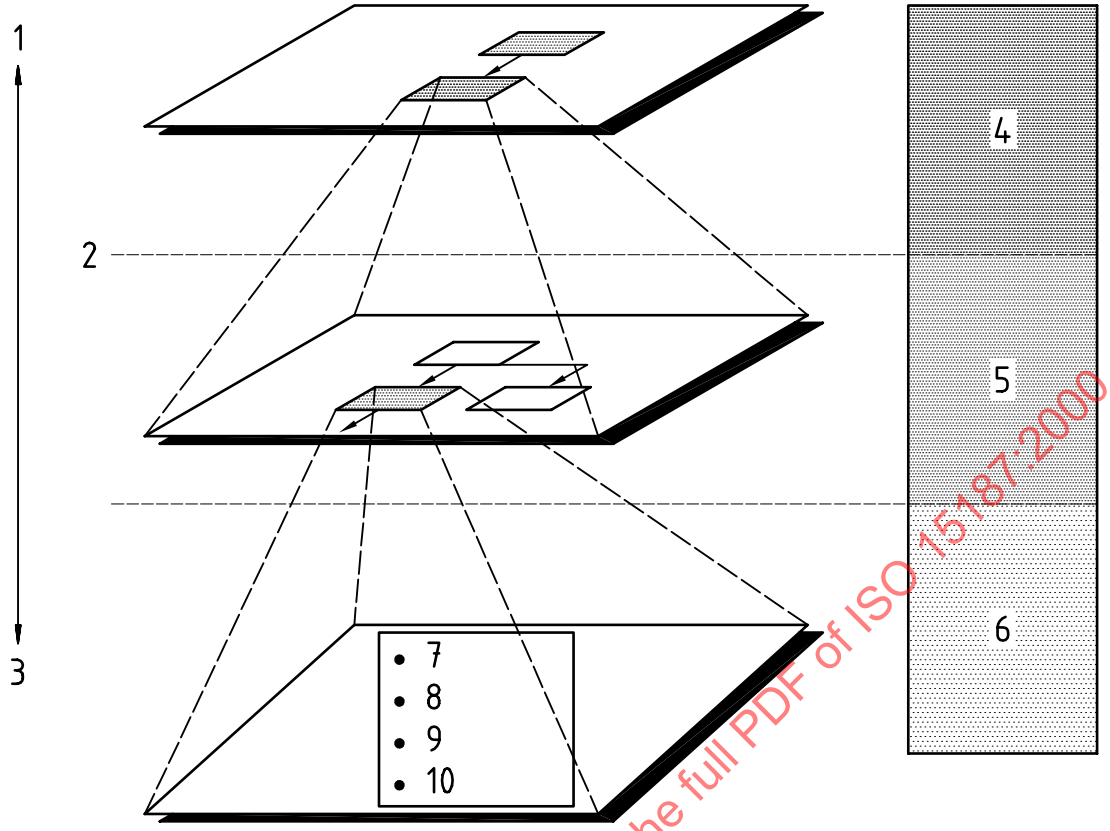
— standardized robot operation.

Programming and operation of industrial robots require a set of tools, e.g. editor, compiler, debugger, which help the user to specify a number of more or less complex functions to be performed by the robot control. There are many different functions for the robot control units in use world wide, and the GUI-R standard should simplify the programming with the help of graphics. In a first step GUI-R will give a style guide for the definition and use of graphical elements for the programming procedure of robots.

Therefore the GUI-R standard is nothing but the interface for the user to pick up elements for programming and operation by an easy manner. The function represented by these elements is defined in general, e.g. as a straight line robot move. The explicit, complete and detailed syntax and semantic meaning is not the purpose of GUI-R, e.g. all move parameters, the exact meaning of "straight line" by the linear interpolation including the behaviour of smoothing, and others. This is defined by the underlying robot system control and programming features: GUI-R is not a robot language expressed by graphical elements and icons. The user is obliged to read the robot system manuals and to learn about the special robot programming and control functions. GUI-R is a help to find the programming elements and to work with a graphic layout in the same manner for different robot systems. The program structure might be displayed on the screen by a program flow chart.

This International Standard only specifies the detailed specifications of GUI-R level and the concept of hierarchization, see Figure 1. The contents of the low-level hierarchy will not be specified intentionally, and consideration will be taken so that implementation will be easier.

Of course, a standard should be complete and precisely defined to cover the subject to be standardized. But there is a variety of robot programming languages and controls with different concepts, functions, syntax, and semantics. Because it is not possible to include all (partly contradictory) concepts and functions of existing (and future) robot languages for a representation by graphical elements, GUI-R covers only the main elements and functions for programming and operation. It doesn't specify all elements of one function level, e.g. icons for all possible statements of a robot language, and it doesn't specify all parameters or elements of a function or statement, e.g. the parameters of a move statement. This means GUI-R is concentrated on the main functions (defined horizontally) and on the main syntactical elements or parameters (defined vertically), see Figure 2.



Key

1 High
2 Abstract Level
3 Low
4 GUI Level
5 Function Level
6 Language Level (Implementation)
7 C
8 Visual Language
9 Flow Chart
10 Others (Interpreter, etc.)

Figure 1 — Image of hierarchies

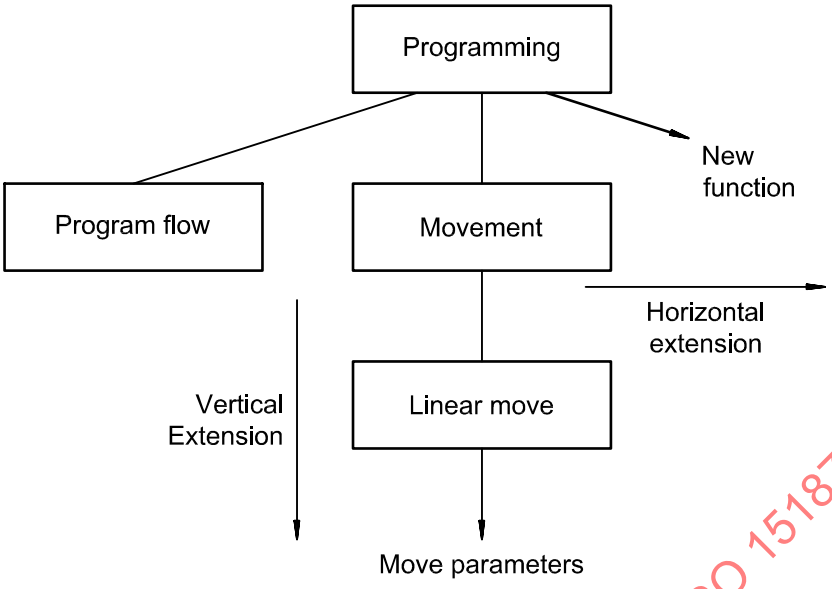


Figure 2 — Horizontal and vertical extensions

The GUI-R standard cannot cover all elements of a specific robot programming language, and it should be possible to use GUI-R also in the future for new functions. Therefore the definition of GUI-R allows the addition of new elements and rules as well as user or robot system vendor specific elements. The user or vendor can specify new elements on high functional level (horizontal extension), like for new statements or new functions, or new parameters can be added (vertical extension), e.g. for move control. Especially, it is not intended that GUI-R represents a complete functional or language definition.

Whenever it is possible, GUI-R contains reference (and perhaps a short explanation in addition, it depends on the background) to a standard or de-facto standard used for its definition base. The objective of the development of GUI-R is the use on robotic applications and not for general use in other fields like graphics. A decision was made to develop rules which are realistic and can find practical applications in industrial production. This resulted in sometimes developing vigorous GUI-R standard specifications and in other occasions provide references to other standards. The first case was used only if it was strongly needed.

GUI-R specifies how to position icons, how to structure the different areas on the screen for user programming and operation input and gives rules for the design of icons. But this could not be a "precise" definition: For example, it is not specified that the area for language statement icons is from screen co-ordinates (22,34) to (345,57) or that only blue is used for icon background. However, the scope of this International Standard has been defined in a pragmatic manner, e.g. the rules for a movement icon. There is a recommendation for GUI-R elements or screen layout, to help the user apply the standard, but not to restrict him too much. GUI-R was designed in such a manner that it is possible to convert the graphical elements into a robot language representation and vice versa.

Manipulating industrial robots — Graphical user interfaces for programming and operation of robots (GUI-R)

1 Scope

This International Standard specifies the structure and the elements of a graphical user interface for programming and operation of robots (GUI-R). Figure 3 shows the relation to the robot system, to the programming and simulation system, and to the program editor.

It is important that a graphic programming system can be used both off-line, working in a separate programming station, and on-line, working in direct connection with a robot system. The same kind of graphical user interface for off-line systems might then be used in combination with any existing robot and any existing textual language. The produced code might be stored on disk or other media for downloading to a robot system, or downloaded directly, e.g. via a serial link. An on-line system might be connected to a robot system via a high speed serial link or might be fully integrated into the robot control system.

The content of this International Standard is focussed on GUI-R for programming. The robot program itself, and its representation are thus beyond the scope of this International Standard.

GUI-R for operation will be covered in a future edition of this International Standard.

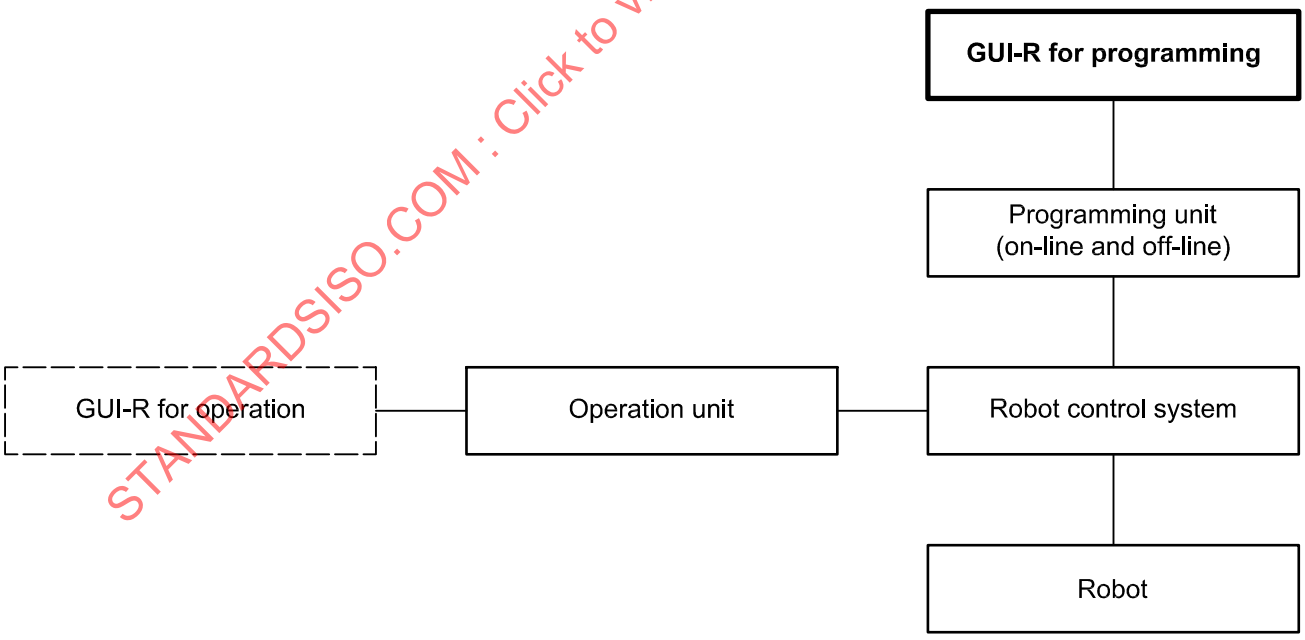


Figure 3 — Model of the context and environment for the robot programming and operation

2 Normative reference

The following normative document contains provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent edition of the normative document indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

IEC 60416, *General principles for the formulation of graphical symbols.*

3 Terms and definitions

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

3.1
dialogue box
secondary window that gathers additional information from a user

3.2
icon
pictorial representation of an object

3.3
icon family
icon family consists of a set of icons which have a common basic icon combined with additional elements

3.4
menu
list of textual and/or graphical choices from which the user can choose

NOTE Different kinds of menus exist, such as pop-up (3.7) and pull-down (3.8) menus.

See Figure 10.

3.5
menu bar
horizontal bar at the top of a window that contains menus

3.6
object
entity or component identifiable by a user that can be distinguished by its properties, operations, and relationships

3.7
pop-up menu
menu that is displayed at the location of a selected object, containing commands that are contextually relevant to the selection

3.8
pull-down menu
menu that is displayed from a menu bar

3.9
status bar
area, typically placed at the bottom of a window, that allows the display of the current status of the information being viewed in the window and the progress of work

See Figure 7.

3.10
tool bar
frame or special area that contains a set of other controls in the form of icons

NOTE Specialized toolbars are sometimes called ribbons, tool boxes, or palettes.

See Figure 9.

3.11
tool bar (or status bar) button
command button used in a tool bar (or status bar)

3.12
window
standard object in the form of a separately controllable area of the screen that typically has a rectangular border and serves to display information

3.13
picture box
rectangular box that includes a picture inside

See Figure 5.

3.14
label
string usually attached to a graphical element for the explanation

See Figure 6.

3.15
command button
button for command input that contains textual and/or graphical elements for operation

See Figure 8.

3.16
radio button
button to select one item from the group that consists of two or more optional items

See Figure 11.

3.17
check box
button to select optional items

NOTE Usually a check mark (✓) or cross mark (x) is used for graphical expression for the selection.

See Figure 12.

3.18
list box
list of choices for the selection of one of them

NOTE It is usually used for the selection of a file by its file names, for example, and is associated with the scroll bar as shown in Figure 15.

See Figure 13.

3.19
combo box
combination of other selection elements such as a list box

NOTE The selection item can be input directly or be selected from a list box.

See Figure 14.

3.20
scroll bar
indication and the control of the relative position in the whole data, usually operated by a pointing device (e.g. mouse)

NOTE Scroll bars are used for horizontal and/or vertical positioning.

See Figure 15.

3.21
slide
element to set the value in a consecutive range

See Figure 16.

3.22
data field
area for inputting its value of the data directly or increasing/decreasing the data usually by the pointing device operation (e.g. mouse)

See Figure 17.

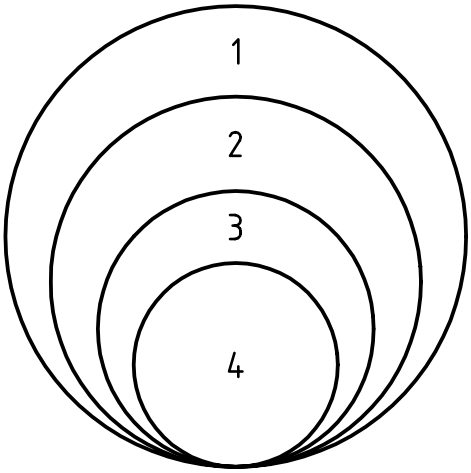
4 Design considerations

Devices that are used for robot programming can be general purpose computers, like a work station, PC etc., or more often, a highly specialized teach pendant, customized for the functionality of a special robot and working on-line. The current trend is to complement the on-line alternative of robot teaching with a PC-based tool, having more and more of the programming work done off-line on general purpose programming systems. Such programming systems might also be used for on-line programming and control, if the workshop requirements of robustness are fulfilled.

A general programming system includes hardware (HW), e.g. PC system, an operating system with graphical user interface, e.g. window-based operating systems, general applications, like text editors, calculation sheets etc., and specific tools for robot programming. Each such application usually complies with the rules and conventions set up by the operating system (OS). This means that a number of common and usual functions, like opening/saving data, editing data, marking data, etc., are similar in different applications making the user familiar with them, also when he works in different applications. The guide describing these rules is commonly referred to as a style guide.

Each general application normally also incorporates a number of unique specific functions. Such functions might be the way to program expressions with mathematical functions in a calculation sheet or the way to pick and program robot instructions and their arguments in a robot programming system. It is highly recommended that the GUI for such more customized general applications, e.g. GUI-R, complies with the operating system style guide.

The different levels can be illustrated as in Figure 4.



Key

1 GUI-R
2 General applications (e.g. programming tools)
3 Operating system
4 Hardware

Figure 4 — Levels of the robot programming system

The standardization work to be undertaken in the working group ISO/TC 184/SC 2/WG 4 deals with the GUI level only, while the OS/style guide should be taken as it is. Since several OS/style guides are known and used today, the standardization work regarding a robotic specific user interface must be possible to apply in any such OS/style guide. This means that the robot specific standard and standardized objects must be independent of any such existing style guide. However, some common features and services must be a prerequisite for the used OS, such as

— window technique, i.e. it must be possible to present data and other information in several separate "windows" with specific attributes such as scroll bars, etc.;

— menu technique, i.e. it must be possible to present a list of available commands in "pull down" or "pop up" menus or with tool bars, etc.;

— icon technique, i.e. it must be possible to represent complex objects with a graphic symbol, which might be "opened" to show its content;

— pointer technique, i.e. it must be possible to select items, commands and objects with a "pointer" tool.

5 GUI-R elements

5.1 Programming concept and general GUI-R elements

Style guides provide a framework of behaviour specifications to guide application or system developers in the design and implementation of new products with consistent user interfaces. The goal is to reach a common "look and feel" for different applications in the same environment. The look and feel includes the types of objects users see on the screen, and the basic conventions for how users work with those objects.

In the following some examples are given on objects, which should be comprised by this style guide:

— programming window: a robot program presented in a window showing the basic program structure with graphical elements;

— instruction parameter: instructions parameters might be programmed by help of dialogue boxes, showing up when selecting and opening an instruction object;

— robot model and work object: views including a model of the robot, the work object and other equipment. Such view might give an animated simulation of robot movements.

There are many GUI-R elements available allowing flexible, application-oriented design and implementation of user interfaces. The following subclauses give examples for GUI elements and how to use them in GUI-R.

5.2 General graphical elements

General graphical elements are composed of the elements listed in 5.2.1 to 5.2.5.

5.2.1 Basic elements

— Picture box (see 3.13 and Figure 5)

— Label (see 3.14 and Figure 6)

Figure 5 — Example of picture box

Figure 6 — Example of label

5.2.2 Elements representing current state and progress of work

— Status bar (see 3.9 and Figure 7)

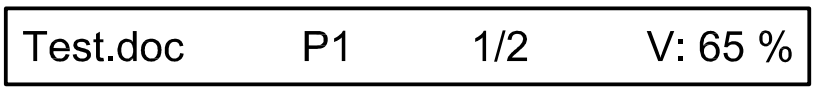


Figure 7 — Example of status bar

5.2.3 Command input

— Command button (see 3.15 and Figure 8)

— Tool bar (see 3.10 and Figure 9)

— Menu (see 3.4 and Figure 10)

Figure 8 — Example of command button

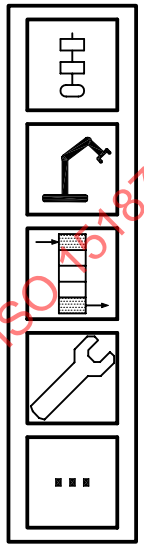


Figure 9 — Example of tool bar

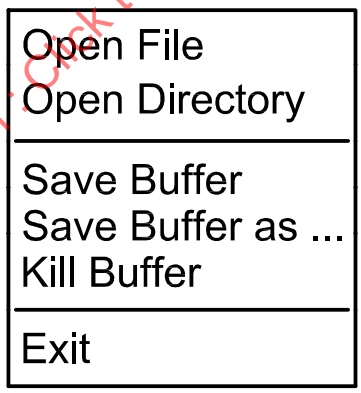


Figure 10 — Example of a menu

5.2.4 Selection elements

— Radio button (see 3.16 and Figure 11)

— Check box (see 3.17 and Figure 12)

— List box (see 3.18 and Figure 13)

— Combo box (see 3.19 and Figure 14)

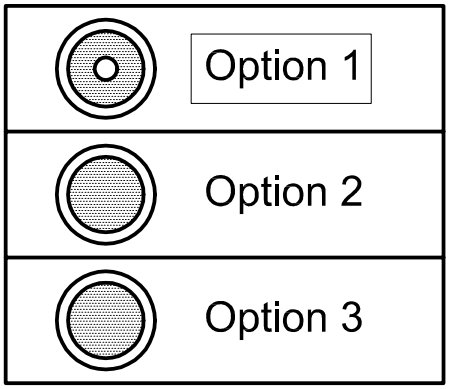


Figure 11 — Example of radio buttons

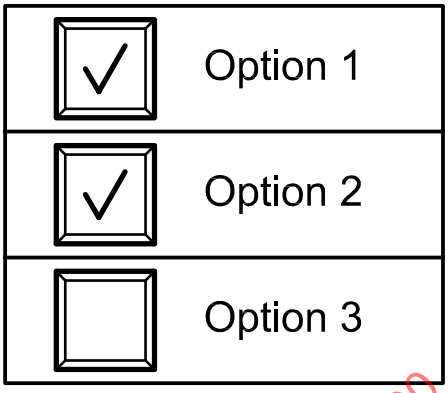


Figure 12 — Example of check boxes

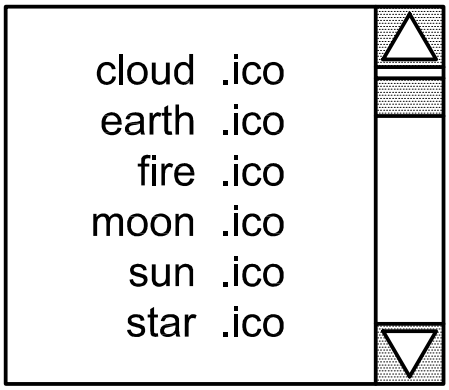


Figure 13 — Example of list box

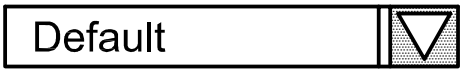


Figure 14 — Example of combo box

5.2.5 Input of values

— Scroll bar (see 3.20 and Figure 15)

— Slide (see 3.21 and Figure 16)

— Data field (see 3.22 and Figure 17)

Figure 15 — Example of scroll bar

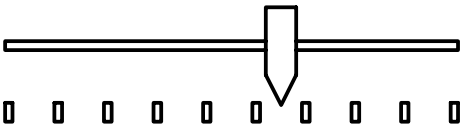


Figure 16 — Example of slide

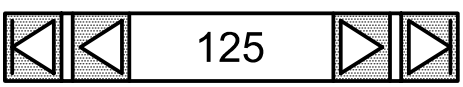


Figure 17 — Example of data field

5.3 Dialogue boxes for robots

GUI-R for robot programming should be strongly based on available graphical environments and according style guides for application design. Robot specific issues are mainly related to suitable selection of basic elements and icon design.

On the following pages some examples of how to use GUI elements in a proper way for robot programming tasks are presented; see Figures 18, 19 and 20.

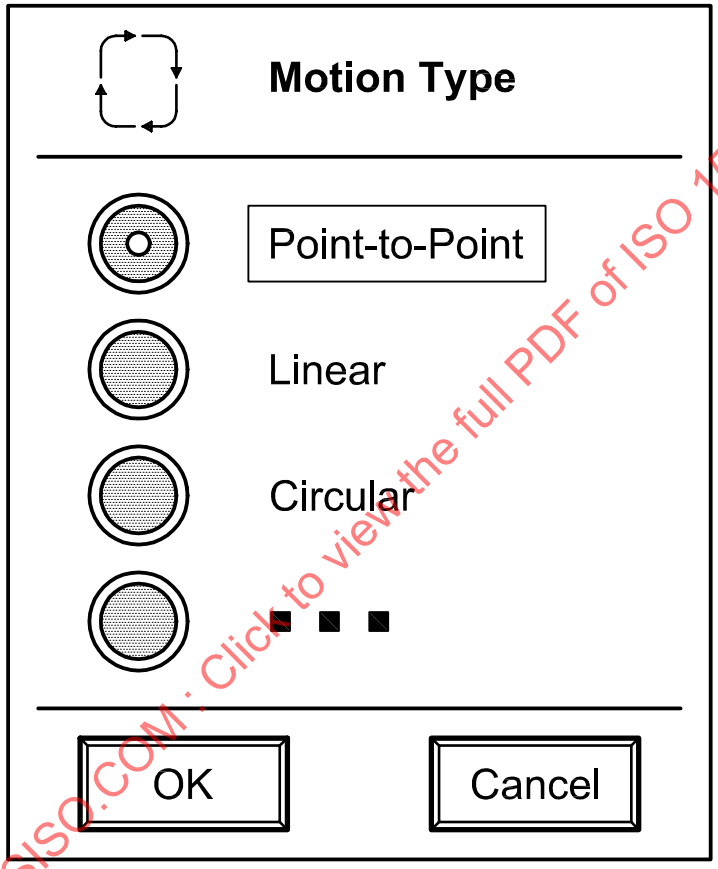


Figure 18 — Selection of motion types using radio buttons

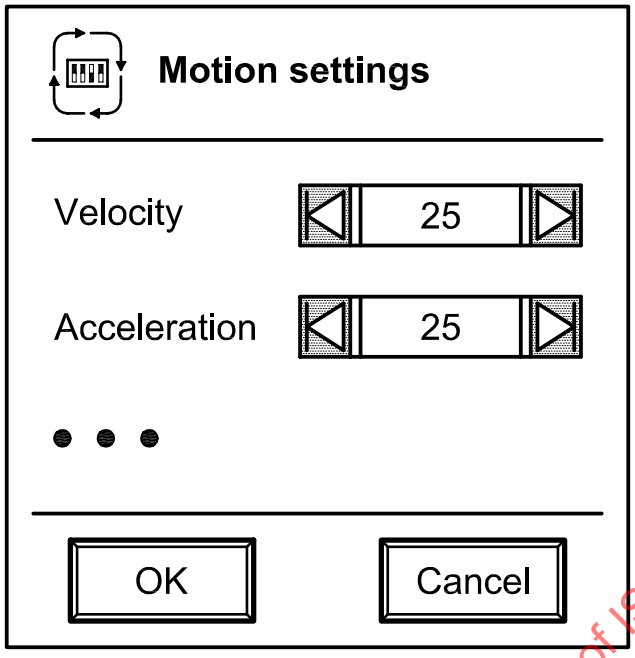


Figure 19 — Input of motion parameters using data fields

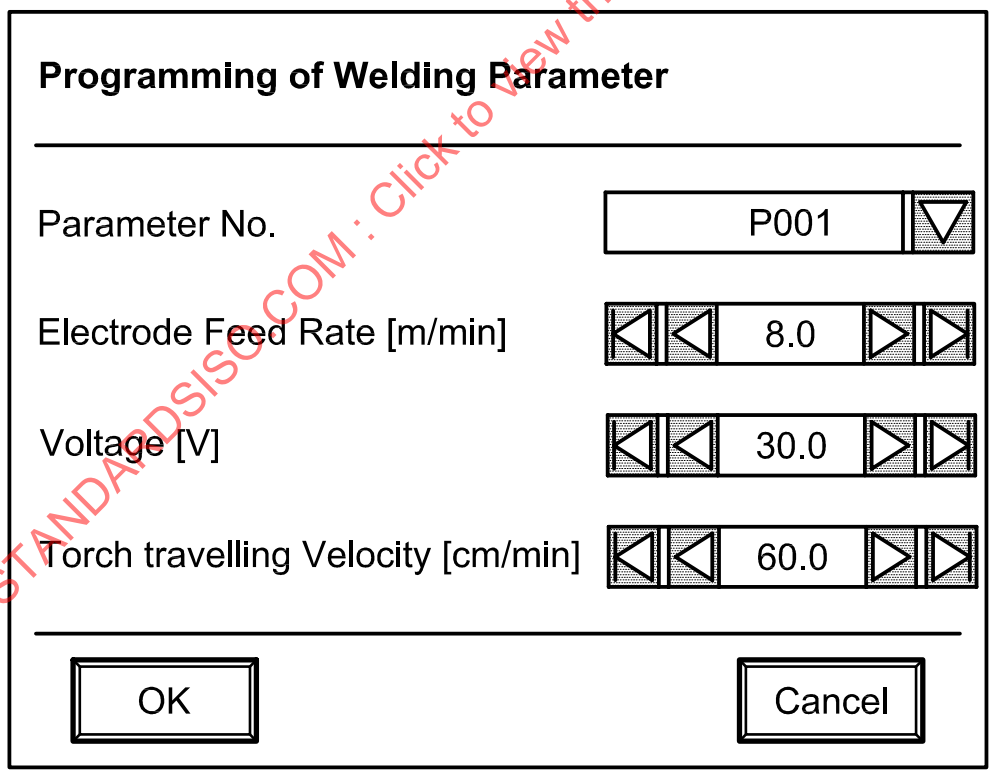


Figure 20 — Use of combo box and data fields for technology programming

5.4 Screen layout

The screen might be divided into the following areas:

— title bar;

— command bar;

— tool bar;

— programming window (subwindows, parameters, dialogue boxes, etc.);

— status bar.

A recommended initial layout of these principal areas is presented in Figure 21.

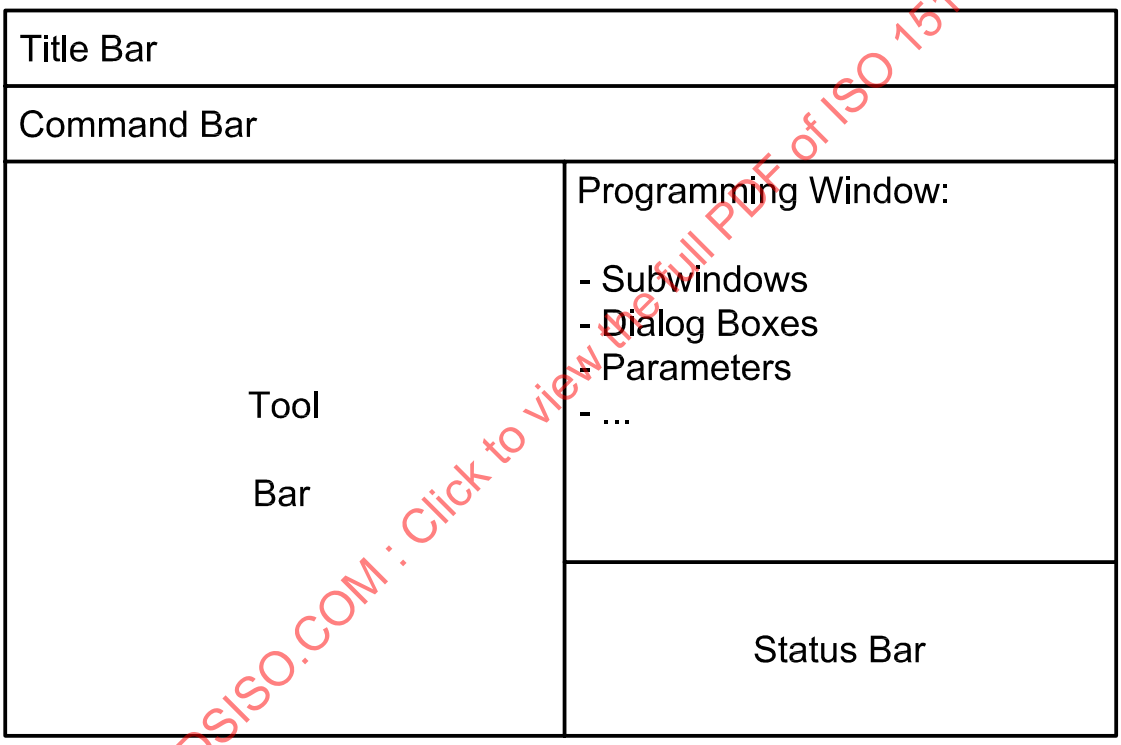


Figure 21 — Main elements and screen layout

The implementation can be based on different operating systems and graphical environments.

6 Functionality and icons

6.1 Functionality concept

A graphical user interface for robots might be divided into a number of different areas of use, each with its own requirements:

— programming;

— system operation;

— production monitoring;

— installing and configuring;

— simulation;

— service and support;

— training.

For each of these, separate standardization work might be necessary. The aim of this work item is the first area, programming. Table 1 shows a possible grouping of different programming elements.

Table 1 — Areas of use “Programming”

Program Control	Motion	Input / Output	Application	Others
— Program flow • If • For • While • Repeat • Switch • Return • Stop — Wait • Digital input • Specified time — Multitasking • Parallel tasks • Synchronization • Messages between tasks — Interrupts • Interrupt service routine • Enable / disable interrupt — Error Handling • Recover from errors	— Motion commands • Activating / deactivating of mechanical units • Motion type •• PTP •• linear •• circular • Moving the robot conditionally • Start / stop — Motion settings • Acceleration / deceleration • Robot configuration • Maximum speed • Payload • Programmed displacement • Soft servo • Tune servo — Motion related sensor functions • Definition and use of sensors • Search movements • Tracking — Other related motion	— Output • Digital output •• Set / reset •• Inverting •• Pulsing • Analog output •• Set / reset — Input • Digital input • Analog input — Operator communication — Computer communication • Read / write • Link request / link answer — Sensor functions • Definition and use of sensors • Binary sensors • Analog sensors • Identifying	— Spot welding — Arc welding • Arcon • Arcoff — Assembly / disassembly — Handling — Glueing — Painting — Finishing • Grinding • Polishing • Brushing	— Data management • Declaration •• Data types •• Procedures •• Functions • Instructions — Mathematics • Arithmetic and boolean operations • Mathematical functions •• Trigonometric •• Exponential •• Co-ordinate transformation • Expressions in logical instructions — User defined instructions — System and Time • Reading data / time • Handling timer

6.2 Icons

6.2.1 Construction rules for icons

6.2.1.1 General design considerations for icons

The designer of the icon shall respect the rules specified in IEC 60416.

General design considerations for icons are as follows.

— Begin by defining the purpose of the icons and its use.

— Brainstorm about possible ideas, considering real world metaphors.

— Draw your ideas using an icon-editor utility or pixel drawing package (to get immediate feedback on their appearance).

— Begin design in black and white; consider colour as an enhancing property.

— Test your images on different backgrounds.

— Design images assuming a light source from the upper left (consistency with other interface elements).

— Avoid using anti-aliasing on the outside edge of an icon.

— Consider the potential cultural impact of a graphic.

6.2.1.2 Rules for icon design

Rules for designing icons are as follows.

— Design icons as a set, considering their relationship to each other and to the user's tasks.

— Related icons should form icon families. An icon family consists of a common basic icon combined with additional elements. The additional elements distinguish the members of the icon family.

— Do several sketches or designs and test them for usability.

— Supply icons for your application in all standard sizes (32×32, 48×48 pixel).

— Icons for documents and data files should be distinct from the application's icon.

— Use a common style across all icons.

— Support user recognition by using effective metaphors. Use real-world objects to represent abstract ideas.

— Design your icons to be simple and distinct.

— Design the small icons to be as similar as possible to their larger counterparts.

— Icons used in different hierarchical levels may be visibly distinguished, e.g. by their colour, size, design of frame of icons.

6.2.2 GUI-R icons

Implementers should adopt the following icons for their own robot system. They are requested to keep the construction rules in 6.2.1, and the results should be similar to the design of the icons shown below.

Table 2 — Representation of icons for area of use "Programming"

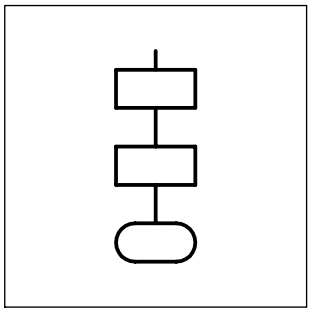
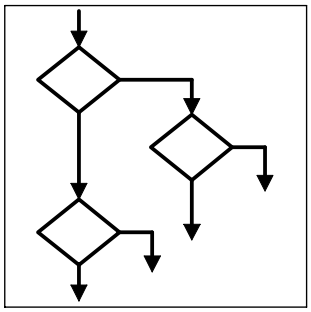
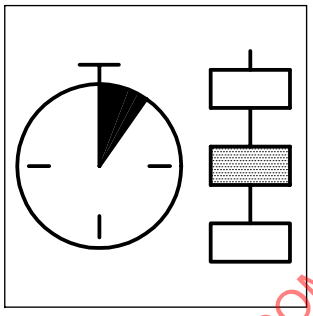
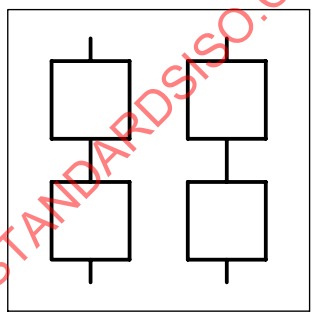
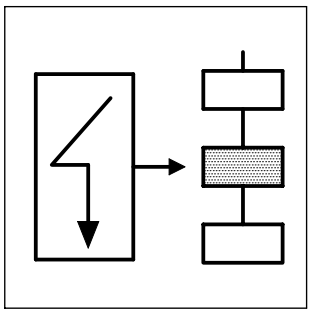
No.	Icon	Meaning	Remarks
1		Program control	
1.1		Program flow	
1.2		Wait	
1.3		Multitasking	
1.4		Interrupts	

Table 2 — *(continued)*

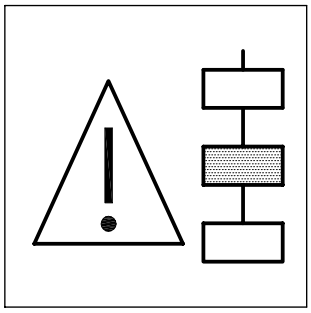
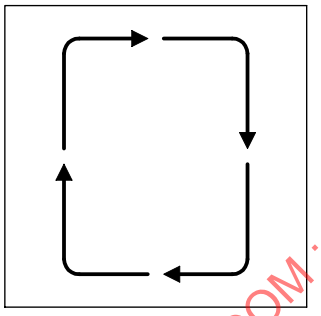
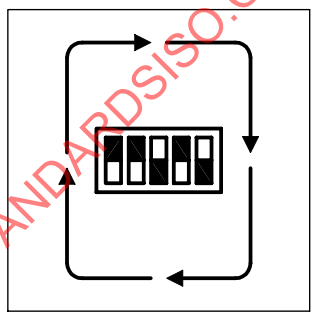
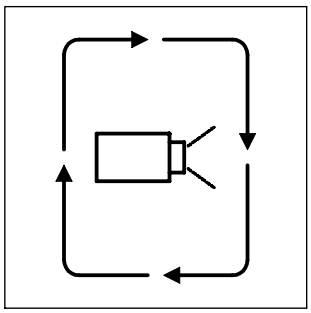
No.	Icon	Meaning	Remarks
1.5		Error handling	
2		Motion	as an alternative: free choice of a manufacturer's specific robot icon
2.1		Motion commands	
2.2		Motion settings	
2.3		Motion related sensor functions	

Table 2 — *(continued)*

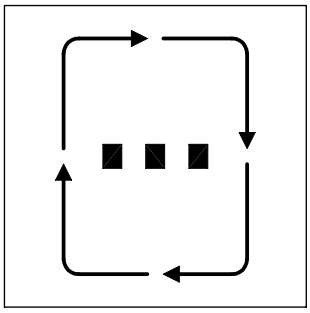
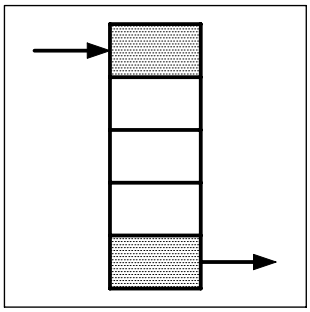
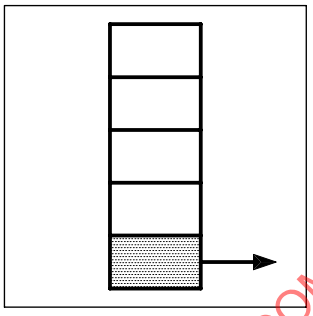
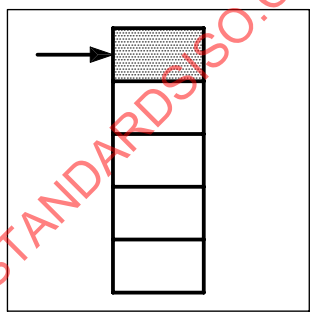
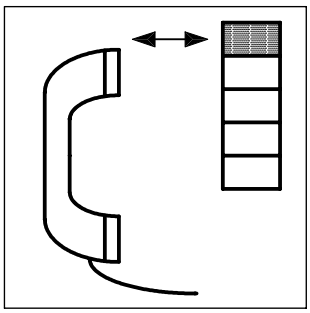
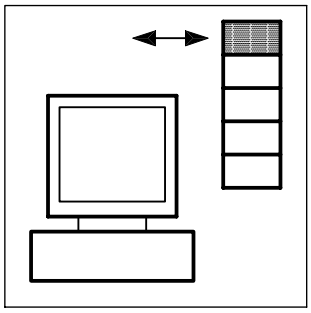
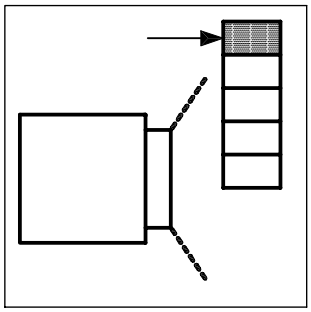
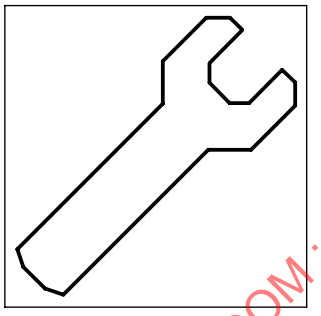
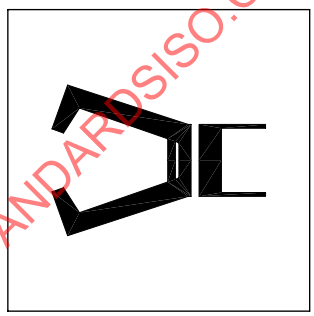
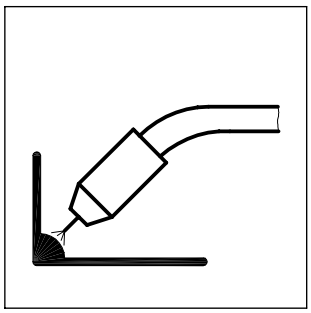
No.	Icon	Meaning	Remarks
2.4		Other related motion	
3		Input / Output	
3.1		Output	
3.2		Input	
3.3		Operator communication	

Table 2 — *(continued)*

No.	Icon	Meaning	Remarks
3.4		Computer communication	
3.5		Sensor functions	
4		Application	
4.1		Spot welding	
4.2		Arc welding	